



Amazon Aurora 첫걸음

엑셈 DB 기술 연구소

엑셈아카데미 <http://exem-academy.com>

Onsite 세미나 및 컨설팅 문의 edu@ex-em.com



엑셈 DB 기술연구소 소개



- 권건우 상무
 - MIS Ph.D.
 - 前 삼성 SDS DBA, TA
 - 現 엑셈 글로벌마케팅본부 상무
 - 現 DB 기술연구소 소장
 - DB Deep Internals 강의 (Oracle, PostgreSQL, MySQL, Aurora)
 - 엑셈아카데미 블록체인 인터널 강의
 - < Oracle, PostgreSQL, MySQL Core Architecture 1 > 책 집필
 - < Oracle, PostgreSQL, MySQL Core Architecture 2 > 책 집필



- 이근오 팀장
 - 삼성카드/삼성생명/ING생명 차세대 PJT 참여
 - 現 DB 기술연구소 팀장
 - 엑셈아카데미 DB Deep Internals 강의 (Oracle, PostgreSQL, MySQL, Aurora)
 - 엑셈아카데미 블록체인 인터널 강의
 - < Oracle, PostgreSQL, MySQL Core Architecture 1 > 책 집필
 - < Oracle, PostgreSQL, MySQL Core Architecture 2 > 책 집필



- 김숙진 연구원
 - 現 DB 기술연구소 선임 연구원
 - 엑셈아카데미 DB Deep Internals 강의 (Oracle, PostgreSQL, MySQL, Aurora)
 - 엑셈아카데미 오로라 첫걸음 강의
 - < Oracle, PostgreSQL, MySQL Core Architecture 1 > 책 집필
 - < Oracle, PostgreSQL, MySQL Core Architecture 2 > 책 집필



- 이대덕 연구원
 - 現 DB 기술연구소 선임 연구원
 - 엑셈아카데미 DB Deep Internals 강의 (Oracle, PostgreSQL, MySQL, Aurora)
 - 엑셈아카데미 블록체인 인터널 강의
 - < Oracle, PostgreSQL, MySQL Core Architecture 1 > 책 집필
 - < Oracle, PostgreSQL, MySQL Core Architecture 2 > 책 집필



Amazon Aurora 첫 걸음

1

AWS란? + 계정 생성
Aurora란? + Aurora 설치
EC2 설치 / AMI 전달

2

Aurora vs. Oracle

3

InnoDB

- 1) 역사
- 2) 아키텍처
- 3) 메모리
- 4) 디스크
- 5) 성능

4

모니터링

- 1) Amazon Cloudwatch
- 2) Enhanced Monitoring
- 3) HammerDB
- 4) MFAA

5

Replication

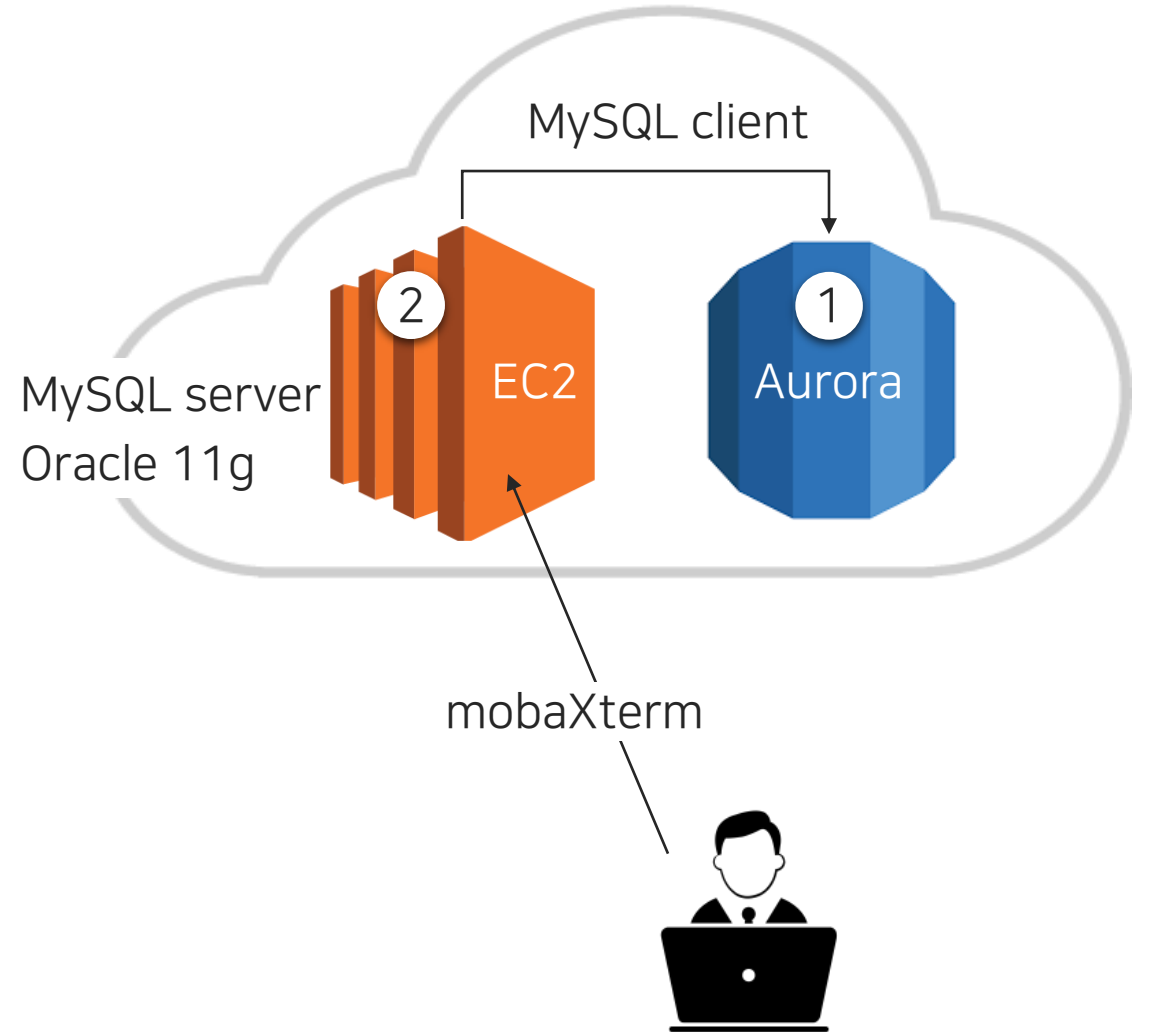
6

Migration



준비사항

1. mobaXterm 설치
 2. 메일 계정 확인
 3. Amazon Aurora 설치
 4. AMI (EC2 CentOS) 설치
- * EC2 CentOS - MySQL 5.7, Oracle 11g





Amazon Web Service 란?



아마존 웹 서비스(Amazon Web Services; AWS)

- 아마존닷컴이 제공하는 각종 원격 컴퓨팅 서비스(웹 서비스)
- 다른 웹 사이트나 클라이언트측 응용 프로그램에 대해 온라인 서비스를 제공
- 서비스의 상당수는 최종 사용자에게 직접 공개되는 것이 아니고, 다른 개발자가 사용 가능한 기능을 제공하는 플랫폼 제공 (PaaS)
- 각종 서비스는 REST 프로토콜 및 SOAP 프로토콜을 통해 접근, 이용 및 관리가 가능
- 비용은 실제 사용량에 따라 결정되며, 일부 서비스의 경우 미리 고정된 금액을 지불함



Compute

Amazon Elastic Compute Cloud (Amazon EC2)



Amazon Elastic MapReduce



Storage

Amazon Simple Storage Service (Amazon S3)



Amazon Elastic Block Storage (Amazon EBS)



AWS Import/Export



AWS Storage Gateway Service

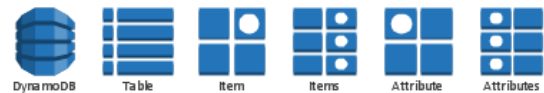


AWS Glacier



Database

Amazon DynamoDB



Amazon Relational Database Service (Amazon RDS)



Amazon ElastiCache



Networking

Amazon Route 53



Amazon Elastic Load Balancing



AWS Direct Connect

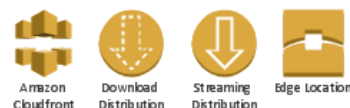


Amazon Virtual Private Cloud (VPC)



Content Delivery

Amazon Cloudfront



Elastic Network Instance



Application Services

Amazon Simple Queue Service (SQS)



Amazon Cloudsearch



Amazon Simple Email Service (SES)



Amazon Simple Workflow (SWF)



Amazon Simple Notification Service (SNS)



Deployment and Management

Amazon Elastic Beanstalk



AWS Identity and Access Management (IAM)



AWS CloudFormation



Monitoring

Amazon CloudWatch



Non-Service Specific



Groups





AWS 계정 생성





AWS 계정 생성

1) AWS 계정 새로 만들기

The screenshot shows the AWS login interface. On the left, there is a '로그인' (Login) section with the AWS logo at the top. Below the logo, the text reads '로그인 ⓘ' and 'AWS 계정의 이메일 주소'. A sub-note says 'IAM 사용자로 로그인하려면 계정 ID 또는 계정 별칭 을 입력하십시오.' Below this is an empty text input field. A blue button labeled '다음' (Next) is positioned below the input field. At the bottom of the login section, there is a link 'AWS를 처음 사용하십니까?' and a button labeled 'AWS 계정 새로 만들기' (Create new AWS account). A red arrow points to this button. On the right side of the page, there is a promotional graphic featuring a laptop with a checkmark icon and the text 'AWS Accounts Include 12 Months of Free Tier Access'. Below this, it lists 'Including use of Amazon EC2, Amazon S3, and Amazon DynamoDB' and provides the URL 'Visit aws.amazon.com/free for full offer terms'.



AWS 계정 생성

2) Credentials

aws

새로운 AWS 계정 만들기

AWS 계정 이름

이메일 주소

암호

암호 확인

계속

[기존의 AWS 계정으로 로그인](#)



AWS 계정 생성

3) 연락처 정보

회사 계정 개인 계정

* 필수 필드

전체 이름*

국가*

주소*

구/군/시*

시/도*

우편 번호*

전화 번호*

AWS 고객 동의

여기를 클릭하면 AWS 고객 동의의 조건을 읽고 동의하는 것을 의미합니다.

계정을 만들고 계속 진행



AWS 계정 생성

4) 결제 정보

본인 확인을 위해 \$1 결제되고, 추후 취소됨

아래에 결제 정보를 입력하십시오. 프리 티어를 통해 무료로 다양한 AWS 제품을 사용해볼 수 있습니다. 프리 티어의 적용을 받지 않는 서비스를 사용할 경우에만 신용카드나 직불카드로 결제하시면 됩니다.

▶ [FAQ](#)

신용/직불 카드 번호

만료 날짜

10 ▼

2017 ▼

카드 소유자 이름

내 연락처 주소 사용

(583, Yangcheon-ro Gangseo-gu Seoul 07547 KR)

새 주소 사용



AWS 계정 생성

5. ID 확인

① 자동화 시스템을 통해 즉시 전화를 받게 되며 제공된 PIN 번호를 입력하게 됩니다.

1. 전화 번호 제공
아래 정보를 입력하고 "지금 전화하기" 버튼을 클릭하십시오.

보안 확인 ⓘ

🔄 🔊



위에 보이는 문자를 입력하십시오.

국가 코드 전화 번호 나가기

대한민국 (+82)

지금 전화하기

2. 전화 거는 중

3. ID 확인 완료

② 자동화 시스템을 통해 즉시 전화를 받게 되며 제공된 PIN 번호를 입력하게 됩니다.

1. 전화 번호 제공 ✓

2. 전화 거는 중
전화의 지시를 따르고 메시지가 들리면 다음 PIN(개인 식별 번호)을 입력하십시오.

PIN: 6191

위에 표시된 번호에서 전화를 받지 못한 경우 잠시 기다려 주십시오. 이 페이지는 다음으로 필요한 사항을 자동으로 업데이트합니다.

3. ID 확인 완료

③ 자동화 시스템을 통해 즉시 전화를 받게 되며 제공된 PIN 번호를 입력하게 됩니다.

1. 전화 번호 제공 ✓

2. 전화 거는 중 ✓

3. ID 확인 완료
ID가 성공적으로 확인됨

지원 계획 선택 계속



AWS 계정 생성

6. 계획 지원

AWS Support는 귀하의 요구를 충족할 수 있는 선별된 계획을 제공합니다. 모든 계획은 고객 서비스, AWS 문서, 백서 및 지원 포럼에 대해 연중무휴 24시간 상시 액세스를 제공합니다. AWS 환경을 계획, 배포, 최적화하는 데 도움이 되는 기술 지원 및 추가 리소스에 액세스하려면 귀하의 AWS 사용에 가장 적합한 지원 계획을 선택하실 것을 권장합니다.

하나를 선택하십시오.

● 기본

설명: 계정 및 과금 관련 질문 및 AWS 커뮤니티 포럼 액세스에 대한 고객 서비스.

요금: 포함됨

● 개발자

사용 사례: AWS 체험

설명: 1차 접촉자는 지원 센터를 통해 기술 관련 질문을 하고 12~24시간 내(해당 지역 업무 시간 중)에 그에 대한 응답을 받을 수 있습니다.

요금: 29 USD/월부터 시작(사용량에 따른 등급)

● 비즈니스

사용 사례: 생산 용도로 AWS 사용하기

설명: 전화 및 채팅을 통한 연중무휴 24시간 지원, 긴급 지원 상황에 대한 1시간 내 응답 및 일반 타사 소프트웨어 지원. AWS 인프라 최적화 용도의 AWS Trusted Advisor에 대한 완전한 액세스 권한, 그리고 지원 사례 자동화 및 Trusted Advisor 결과 조회 용도의 AWS Support API에 대한 액세스 권한.

요금: 100 USD/월부터 시작(사용량에 따른 등급)

● 기업

사용 사례: 미션 크리티컬 용도로 AWS 사용하기

설명: 비즈니스 지원 플랜의 모든 기능뿐만 아니라 다음과 같은 기능이 있음 - 1) 사전 안내 및 모범 사례를 제공하여 AWS 솔루션의 플랜, 개발, 실행을 지원하는 할당된 기술 계정 관리자(TAM), 2) 청구, 계정 분석 및 지원을 제공하는 지원 컨시어지, 3) 인프라 이벤트 관리에 액세스하여 제품 출시, 계절별 프로모션/이벤트 및 마이그레이션 지원, 4) 우선 순위에 따른 사례 처리 방식으로 중요 지원 사례에 15분 내 응답

요금: 15,000 USD/월부터 시작(사용량에 따른 등급)

이 옵션을 선택하면 고객 지원 부서에서 48시간 내에 귀하에게 연락해 귀하의 요구 사항을 논의하고 가용 절차를 완료할 것입니다. 지원 리소스는 가용 완료 시 사용할 수 있고 그때까지는 요금이 발생하지 않습니다.

계획 비교 및 요금 예시를 비롯한 AWS Support의 모든 기능과 혜택을 알아보시려면 [여기를 클릭하십시오.](#)

계속



AWS 계정 생성

7. 완료



Amazon Web Services 소개

AWS(Amazon 웹 서비스) 계정을 만들어 주셔서 감사합니다. 귀하의 계정을 활성화하고 있으므로 AWS를 사용할 수 있습니다. 대부분 고객의 경우 활성화에는 몇 분 정도 소요됩니다(추가 계정 확인이 필요한 경우 몇 시간이 필요할 수 있습니다). 확인이 완료되고 계정이 활성화되면 이메일로 알려드립니다.

[관리 콘솔 시작](#)

[영업부서 연락](#)



실습 전 확인사항 - 계정

본인이 받은 계정과 비밀번호로 접속하기

예) 계정: exem_edu01@ex-em.com

비밀번호: 1234



Amazon Aurora 특징





Amazon Aurora란?



컴퓨팅



스토리지



데이터베이스



마이그레이션



네트워킹 및 콘텐츠 전송

Amazon Aurora

고성능 관리형 관계형 데이터베이스

Amazon RDS

MySQL, PostgreSQL, Oracle, SQL Server 및 MariaDB를 위한 관리형 관계형 데이터베이스 서비스

Amazon DynamoDB

관리형 NoSQL 데이터베이스

Amazon DynamoDB Accelerator(DAX)

DynamoDB용 완전관리형 인 메모리 캐시

Amazon ElastiCache

인 메모리 캐시

Amazon Redshift

빠르고 간단하며 비용 효과적인 데이터 웨어하우징

AWS Database Migration Service

최소한의 다운타임으로 데이터베이스를 이전하세요



Amazon Aurora 주요 특징



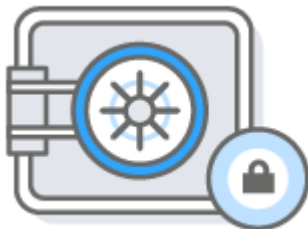
뛰어난 성능 및 확장성

- MySQL 보다 5배, PostgreSQL 보다 3배 많은 처리량 제공
- 비용 1/10
- 3개의 AZ에 걸쳐 읽기전용 복제본 최대 15개 추가
- 스토리지를 인스턴스당 최대 64TB 자동 확장



높은 가용성 및 내구성

- 3개 AZ 걸쳐 6개의 데이터 복사본 복제, 클라우드용으로 구축
- 내결함성 갖춘 자가 복구 스토리지
- Amazon S3로 백업
- 물리적 스토리지 장애로부터 투명하게 복구
- 보통 30초 이내 인스턴스 장애 조치 수행

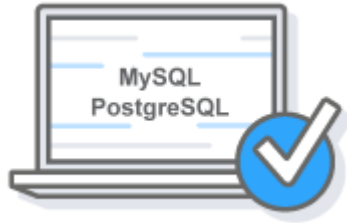


뛰어난 보안

- VPC를 사용한 네트워크 격리
- 사용자가 생성한 키를 사용한 저장 데이터 암호화
- AWS Key Management Service (KMS)를 통한 제어
- SSL을 사용한 전송 데이터 암호화



Amazon Aurora 주요 특징



MySQL 및 PostgreSQL 호환성

- 새로운 릴리스에 대한 호환성이 정기적으로 추가
- 가져오기/내보내기 도구나 스냅샷으로 마이그레이션 가능
- 코드, 애플리케이션, 드라이버 및 도구를 변경하지 않고 사용 가능



완전 관리형

- Amazon RDS에서 Amazon Aurora의 모든 것을 관리
- 하드웨어 프로비저닝, 소프트웨어 패치, 설정, 구성, 백업
- 자동적, 지속적으로 S3로 백업하므로 특정 시점으로 복구 가능
- CloudWatch, Enhanced Monitoring, Performance Insights 기능을 사용하여 성능 모니터링

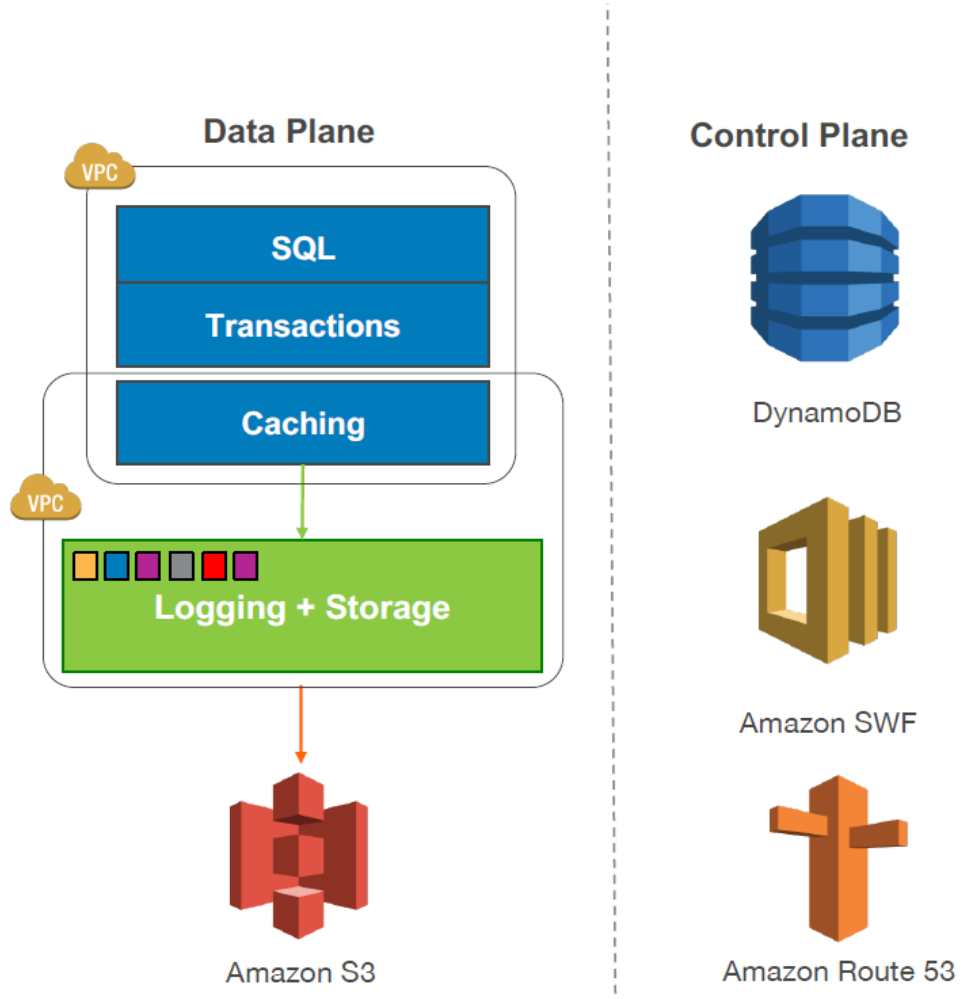


마이그레이션 지원

- AWS Database Migration Service



Amazon Aurora Architecture

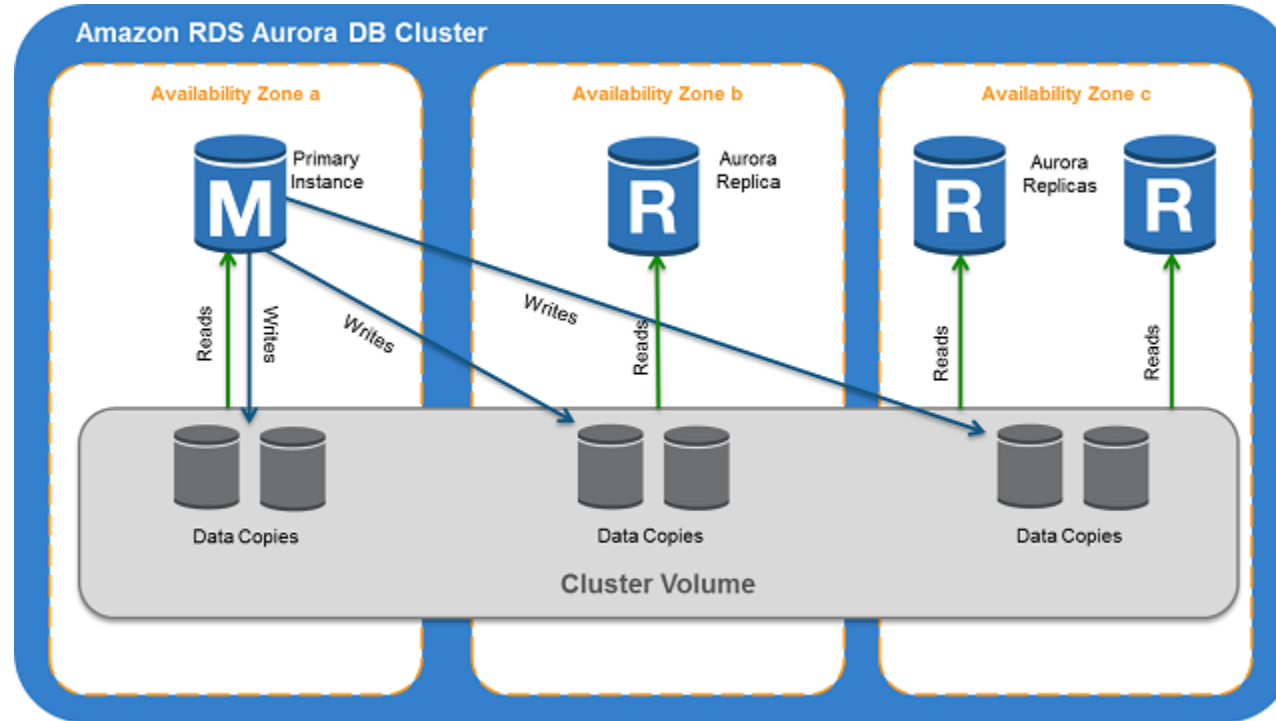


서비스 중심 아키텍처 적용

- 1) 로깅 및 스토리지 레이어를 멀티-테넌시 스케일-아웃 기반 데이터베이스 최적화된 스토리지 서비스로 전환
- 2) 서비스 내부 운영을 위하여 Amazon EC2, Amazon VPC, Amazon DynamoDB, Amazon SWF, and Amazon Route 53 등 다른 AWS 서비스들과 통합
- 3) 연속적인 백업을 위한 Amazon S3와 통합으로 99.999999999% 내구성 제공



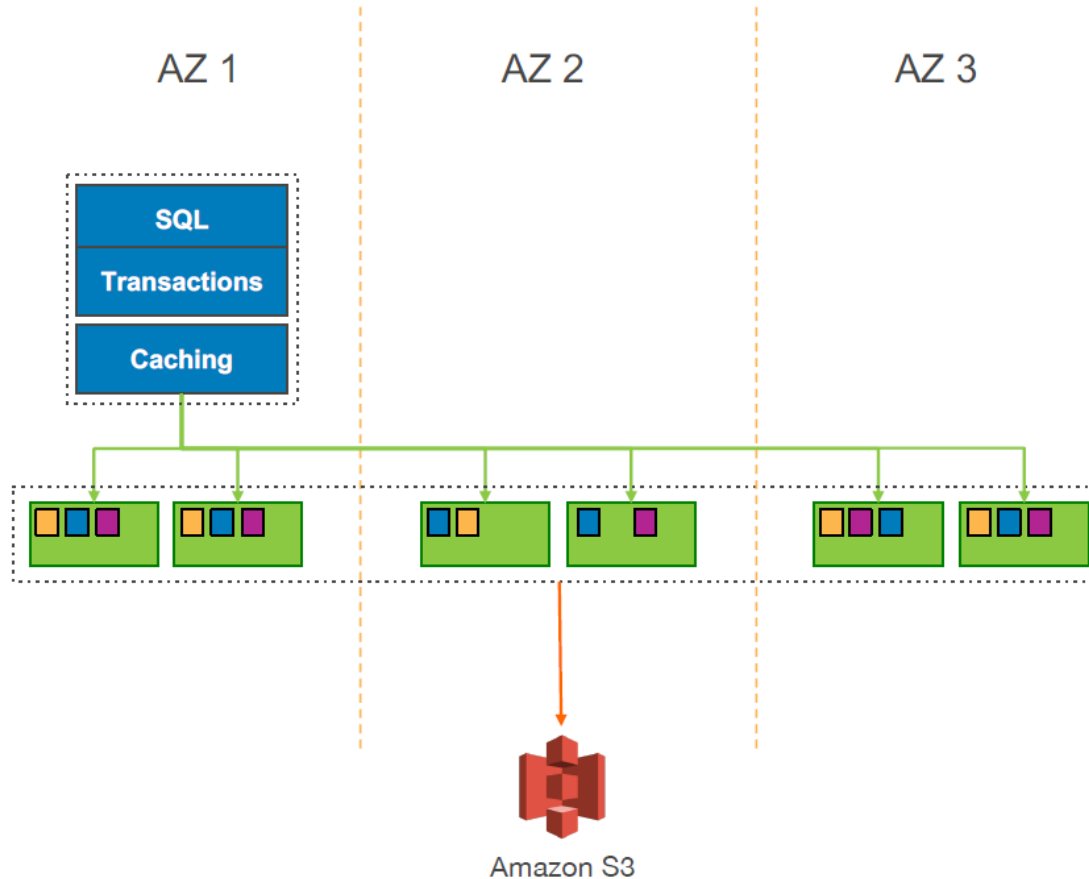
Amazon Aurora Architecture



- Amazon Aurora를 생성하면 DB 클러스터가 생성됨
- DB 클러스터는 **하나 이상의 인스턴스**와 이 인스턴스의 데이터를 관리하는 **클러스터 볼륨**으로 구성
- Aurora 클러스터 볼륨은 다중 가용 영역을 모두 아우르는 가상 데이터베이스 스토리지 볼륨으로서, 각 가용 영역에는 클러스터 데이터의 사본이 복사됨



Amazon Aurora Architecture



기본 고가용성

- 3 가용영역(AZ)에 6-way 복제
- 4/6 쓰기, 3/6 읽기 쿼럼
- S3 저장소에 연속 백업

SSD, Scale-out, Multi-tenant storage

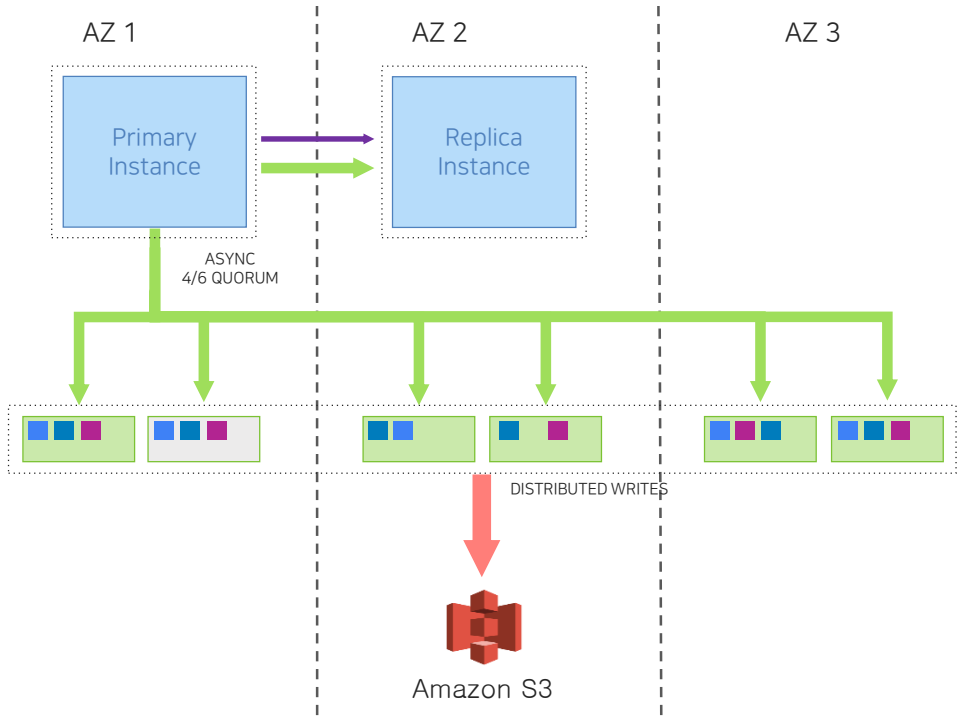
- 연속적 스토리지 확장
- 최대 64TB 크기
- 사용한 만큼만 지불

로그-구조 기반 스토리지

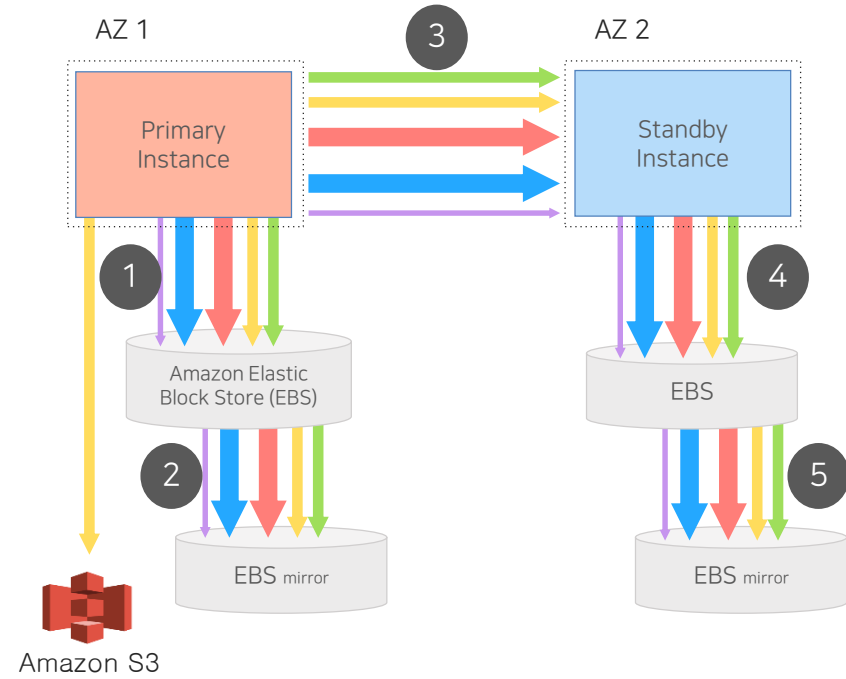


Amazon Aurora Architecture – I/O Traffic

AMAZON AURORA



MYSQL WITH STANDBY



TYPE OF WRITES





NEW!

Aurora Multi-Masters

다중 데이터센터에 읽기/쓰기에 대해 스케일 아웃(Scale Out) 가능한 최초의 관계형 데이터베이스 서비스



3개의 가용 영역에 15개의 읽기 복제본 확장 가능

읽기 복제본 자동 스케일링

빠른 읽기 복제본 페일 오버

어떤 노드에도 애플리케이션 다운타임 없음

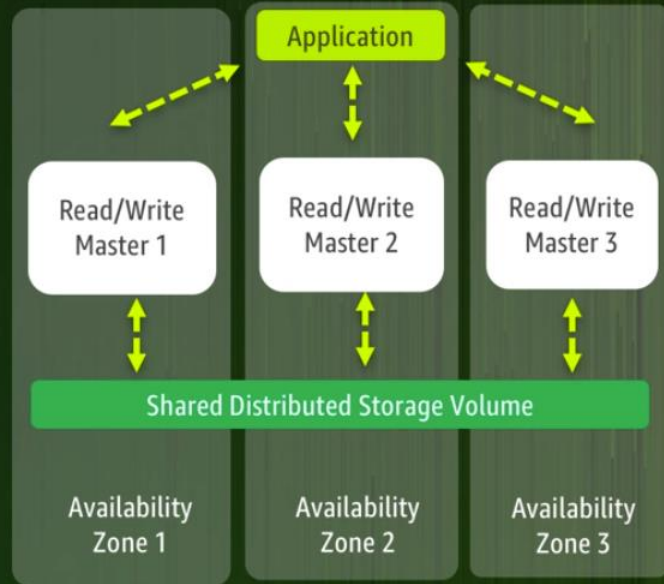
어떤 AZ에도 애플리케이션 다운타임 없음

빠른 읽기 성능 보장

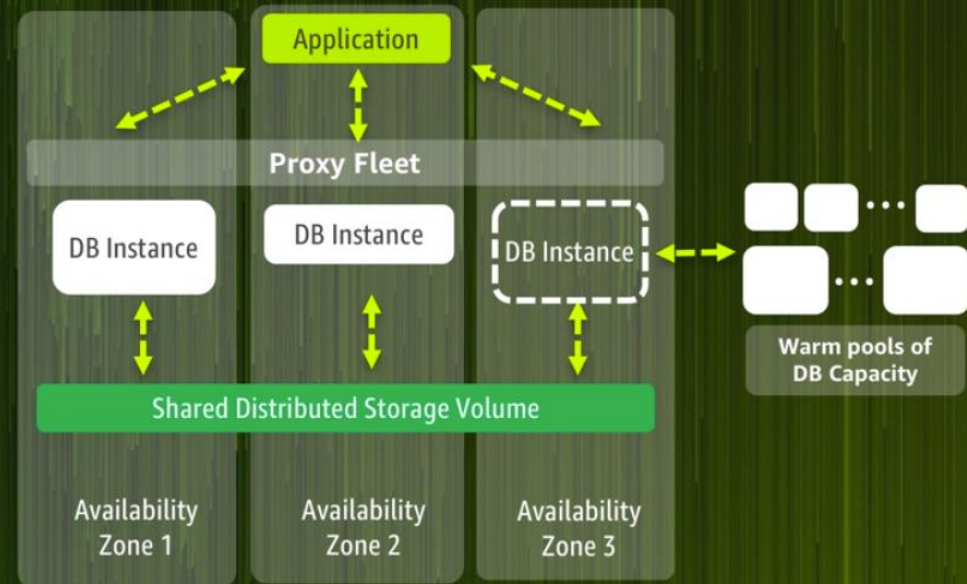
멀티 리전 기능 제공 (2018)



데이터베이스 서비스 혁신 방향



Amazon Aurora
Multi-masters



Amazon Aurora
Serverless



Aurora Multi-Master

애플리케이션의 다운타임을 없앨 수 있는 서비스다. 이 서비스는 애플리케이션이 한 클러스터 안에서 복수의 DB 인스턴스에 데이터를 읽고 쓸 수 있다. 지금까지는 읽기만 됐다. 이를 통해고가용성을 향상시킬 수 있다. 마스터 DB 서버에 장애가 나도 클러스터 안의 다른 서버가 즉시 이를 대체한다. 제시 CEO는 “특정 가용성 존에서 고객이 쓰기에 실패하더라도 응용 프로그램에 실질적인 영향을 미치지 않는다”고 설명했다.

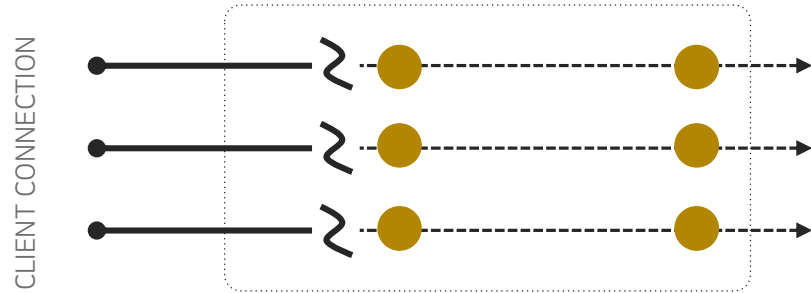
(출처: <https://byline.network/2017/12/4-7/>)

- 복수의 가용 영역에서 작동하는 복수의 읽기/쓰기 마스터 인스턴스 생성 가능
- 고가용성 향상
- 마스터 인스턴스 중 하나에 장애가 발생하면, 클러스터의 다른 인스턴스가 즉시 이어받아서 중단되지 않음 (인스턴스 장애 뿐만 아니라 AZ 전체에 장애 발생도 포함)

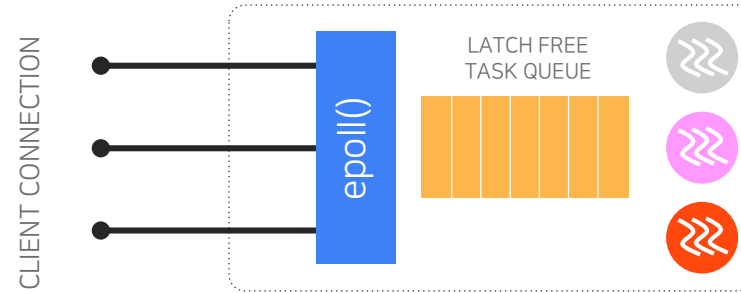


Adaptive thread pool

MYSQL THREAD MODEL



AURORA THREAD MODEL



- 병렬적으로 많은 쿼리 처리
- 동적으로 확장 / 축소



thread_handling

Aurora

```
E1> show global variables like 'thread_handling';
+-----+-----+
| Variable_name | Value |
+-----+-----+
| thread_handling | multiple-connections-per-thread |
+-----+-----+
1 row in set (0.01 sec)
```

RDS MySQL (5.7)

```
E2> show global variables like 'thread_handling';
+-----+-----+
| Variable_name | Value |
+-----+-----+
| thread_handling | one-thread-per-connection |
+-----+-----+
1 row in set (0.01 sec)
```

multiple-connections-per-thread

- Aurora에만 존재하는 것으로 dynamic connection pooling을 사용하는, clients에 의해 실행되는 쿼리 작업과 handling connections의 보장을 위해 스레드에 미리 풀이 할당됨 (pre-allocate pool)
- workload fluctuation(작업량 변동)에 따라 dynamic, auto-size
- 스레드의 미리 할당된 풀의 설계 때문에 각 스레드는 다수의 커넥션을 핸들하게 됨
- 동시 접속자 수에 대하여 16,000 max connections을 지원

one-thread-per-connection



Amazon Aurora 설치 시연





Aurora DB 생성

1) 홈페이지 접속 aws.amazon.com/ko/





Aurora DB 생성

2) 로그인



루트 사용자 로그인

이메일
internals@ex-em.com

비밀번호
[password field]

[로그인](#)

[다른 계정으로 로그인](#)
비밀번호가 생각나지 않는 경우



Amazon Aurora | 1/10의 비용으로 사용할 수 있는 엔터프라이즈급 데이터베이스

"MySQL 환경보다 10배 빠른 속도 정말 효과가 있습니다!"
-Alfresco

[자세히 알아보기](#)

Amazon.com 로그인 정보

Amazon Web Services는 귀하의 Amazon.com 계정의 정보를 사용하여 자격 증명을 확인하고 Amazon Web Services에 대한 액세스를 허용합니다. 이 사이트의 사용은 아래 링크된 이용 약관과 개인 정보 보호 정책을 따릅니다. 해당 제품 및 서비스를 AWS 부가 가치 대리점에서 구매한 경우를 제외하고 Amazon Web Services 제품 및 서비스의 사용은 아래 링크된 AWS 고객 계약에 따릅니다. AWS 고객 계약은 2017년 6월 28일에 업데이트되었습니다. 해당 업데이트에 대한 자세한 내용은 다음을 참조하십시오. [최근 변경 사항](#).

이용 약관 개인 정보 보호 정책 AWS 고객 계약 © 1996-2017, Amazon.com, Inc. 또는 계열사

An  company



Aurora DB 생성

3) RDS

region 선택



The screenshot shows the AWS Management Console interface. At the top, the navigation bar includes 'Services', 'Resource Groups', 'EC2', 'Database Migration Service', and 'RDS'. The region is set to 'Seoul'. The main content area displays a grid of AWS services. Under the 'Database' category, 'RDS' is highlighted with a red box. A red arrow points from the text 'RDS' to this box. Another red arrow points from the text 'region 선택' to the 'Seoul' dropdown menu in the top right corner.



Aurora DB 생성

4) Launch a DB Instance

The screenshot shows the AWS Management Console interface for the Amazon RDS service. The left-hand navigation pane lists various RDS components, with 'Instances' selected. The main content area features a top banner for Amazon Aurora with a 'Launch Aurora MySQL' button. Below this is a 'Resources' section listing available RDS resources in the Asia Pacific (Seoul) region. The 'Create Instance' section is highlighted with a red box and an arrow, containing a 'Launch a DB Instance' button. Below this is a 'Service Health' section with a table showing the status of Amazon RDS in the Seoul region as 'operating normally'.

RDS Dashboard

- Instances
- Clusters
- Reserved Instances
- Snapshots
- Parameter Groups
- External Licenses
- Option Groups
- Subnet Groups
- Events
- Event Subscriptions
- Notifications

Amazon Aurora
Amazon Aurora is a MySQL- and PostgreSQL-compatible enterprise-class database, starting at <\$1/day. Aurora supports up to 64TB of auto-scaling storage capacity, 6-way replication across three availability zones, and 15 low-latency read replicas.
[Learn more.](#)
[Launch Aurora MySQL](#)
Or, [Restore Aurora DB Cluster from S3](#)

Resources

You are using the following Amazon RDS resources in the Asia Pacific (Seoul) region (used/quota):

DB Instances (1/40)	Parameter Groups (2)
Allocated Storage (0 bytes/100.00 TB)	Default (2)
Click here to increase DB instances limit.	Custom (0/100)
Reserved DB Purchases (0/40)	Option Groups (1)
Snapshots (13)	default (1)
Manual (0/100)	Custom (0/20)
Automated (2)	Subnet Groups (1/50)
Recent Events (0)	Supported Platforms VPC
Event Subscriptions (0/20)	Default Network vpc-d38a7bbb
	External Licenses (0)

Create Instance

Amazon Relational Database Service (RDS) makes it easy to set up, operate, and scale a relational database in the cloud.

[Launch a DB Instance](#)

Note: Your DB Instances will launch in the Asia Pacific (Seoul) region:

Service Health

Current Status	Details
Amazon Relational Database Service (Seoul)	Service is operating normally

[View complete service health details](#)

Additional Information

- Starting with RDS
- Overview and Features
- Documentation
- Articles and Tutorials
- Data import guide for MySQL
- Data import guide for Oracle
- Data import guide for SQL Server
- Pricing
- Forums

What's New

Amazon Aurora – Lambda Integration

Now you can invoke Lambda functions directly from within an Aurora Database.
[Find out how](#)

Related Services

Amazon QuickSight

Analyze and visualize your RDS data using Amazon QuickSight. Get started for free.
[Get started for free](#)

Amazon ElastiCache

Add a managed Redis or Memcached-compatible in-memory cache to speed up your database access.
[Click here to learn more and launch your Cache Cluster](#)

Feedback English (US) © 2008 - 2017, Amazon Web Services, Inc. or its affiliates. All rights reserved. [Privacy Policy](#) [Terms of Use](#)



Aurora DB 생성

5) Select Engine - Amazon Aurora 선택

The screenshot shows the AWS Management Console interface for creating a database instance. The top navigation bar includes 'Services', 'Resource Groups', 'EC2', 'Database Migration Service', and 'RDS'. The main content area is titled 'Step 1: Select Engine' and includes a 'Free tier eligible only' checkbox. A list of database engines is shown on the left: Amazon Aurora, MySQL, MariaDB, PostgreSQL, ORACLE, and Microsoft SQL Server. The 'Amazon Aurora' option is selected and highlighted. To the right of the selected engine, there is a description: 'Amazon Aurora is a MySQL- and PostgreSQL-compatible enterprise-class database, starting at <\$1/day.' Below this, a list of features is provided: 'Up to 5 times the throughput of MySQL and 3 times the throughput of PostgreSQL.', 'Up to 64TB of auto-scaling SSD storage.', '6-way replication across three Availability Zones.', 'Up to 15 Read Replicas with sub-10ms replica lag.', and 'Automatic monitoring and failover in less than 30 seconds.' At the bottom of the selected engine's details, it says 'MySQL-compatible edition' and a blue 'Select' button is visible. A 'Cancel' button is located at the bottom right of the console window.



Aurora DB 생성

6) Specify DB Details

Step 1: Select Engine
Step 2: Specify DB Details
Step 3: Configure Advanced Settings

The following selections disqualify the instance from being eligible for the free tier:

- DB Instance Class
- Engine

You can receive a significant savings over on-demand instance costs with [Reserved Instances](#).

Estimate your monthly costs for the DB Instance using the [AWS Simple Monthly Calculator](#).

Specify DB Details

Instance Specifications

DB Engine: Aurora - compatible with MySQL 5.6.10a
DB Instance Class: db.r3.large — 2 vCPU, 15.25 GiB RAM
Multi-AZ Deployment: Create Replica in Different Zone

Settings

DB Instance Identifier*: mydbinstance
Master Username*: awsuser
Master Password*:
Confirm Password*:

Retype the value you specified for Master Password.

* Required

Cancel Previous Next Step

Instance Specifications

- DB Engine
- DB Instance Class
- Multi-AZ Deployment

Settings

- DB Instance Identifier
- Master Username
- Master Password
- Confirm Password



Instance Specifications: 인스턴스 세부정보

항목	설명
DB Engine	Aurora - compatible with MySQL 5.6.10a
DB Instance Class	<p>DB 클러스터의 각 인스턴스 처리 및 메모리 요건을 정의한 DB 인스턴스 클래스를 선택할 수 있음. Aurora는 db.t2.small, db.t2.medium, db.r3.large, db.r3.xlarge, db.r3.2xlarge, db.r3.4xlarge, db.r3.8xlarge DB 인스턴스 클래스를 지원함.</p> <p>db.t2.small — 1 vCPU, 2 GiB RAM db.t2.medium — 2 vCPU, 4 GiB RAM db.r3.large — 2 vCPU, 15.25 GiB RAM db.r3.xlarge — 4 vCPU, 30.5 GiB RAM db.r3.2xlarge — 8 vCPU, 61 GiB RAM db.r3.4xlarge — 16 vCPU, 122 GiB RAM db.r3.8xlarge — 32 vCPU, 244 GiB RAM db.r4.large — 2 vCPU, 15.25 GiB RAM db.r4.xlarge — 4 vCPU, 30.5 GiB RAM db.r4.2xlarge — 8 vCPU, 61 GiB RAM db.r4.4xlarge — 16 vCPU, 122 GiB RAM db.r4.8xlarge — 32 vCPU, 244 GiB RAM db.r4.16xlarge — 64 vCPU, 488 GiB RAM</p>
Multi-AZ Deployment	장애 조치 지원을 위해 다른 가용 영역(Availability Zone)에 Aurora 복제본(standby)을 생성할지 여부를 결정



Settings

항목	설명
DB Instance Identifier	<p>DB 클러스터의 기본 인스턴스 이름을 입력. DB 클러스터의 기본 인스턴스의 엔드포인트 주소로 사용됨.</p> <ul style="list-style-type: none">• 1~63자의 영숫자 문자 또는 하이픈으로 구성• 첫 번째 문자는 글자• 하이픈으로 끝나거나 하이픈 2개 연속은 사용 불가• 각 리전별로 AWS 계정 1개의 모든 DB 인스턴스는 고유해야 함
Master Username	<p>영숫자 문자를 사용하여 DB 클러스터의 DB 인스턴스에 로그인할 때 사용할 마스터 사용자 이름을 입력. 마스터 사용자 이름 계정에 기본적으로 부여되는 권한으로는 create, drop, references, event, alter, delete, index, insert, select, update, create temporary tables, lock tables, trigger, create view, show view, alter routine, create routine, execute, create user, process, show databases, grant option이 있음.</p>
Master Password	<p>8~41자의 인쇄 가능한 ASCII 문자(/, ", 및 @ 제외)로 마스터 사용자 암호 설정</p>
Confirm Password	<p>다시 한번 암호 입력</p>



Aurora DB 생성

7) Step3: Configure Advanced Settings

Network & Security

- VPC
- Subnet Group
- Publicly Accessible
- Availability Zone
- VPC Security Group(s)

Database Options

- DB Cluster Identifier
- Database Name
- Database Port
- DB Parameter Group
- DB Cluster Parameter Group
- Option Group
- Enable IAM DB Authentication
- Enable Encryption
- Master Key
- Description
- Account
- KMS Key ID

Failover

- Priority



Aurora DB 생성

7) Configure Advanced Settings

Backup

- Backup Retention Period

Monitoring Database Options

- Enable Enhanced Monitoring
- Monitoring Role
- Granularity

Maintenance

- Auto Minor Version Upgrade
- Maintenance Window

Backup

Backup Retention Period: 1 days

Monitoring

Enable Enhanced Monitoring: Yes

Monitoring Role: Default

Granularity: 60 second(s)

I authorize RDS to create the IAM role rds-monitoring-role.

Maintenance

Auto Minor Version Upgrade: Yes

Maintenance Window: No Preference

* Required

Cancel Previous **Launch DB Instance**

Feedback English (US)

- The backup retention period determines the period for which you can perform a point-in-time recovery
- Aurora offers 1-day backup retention for free!



Configure Advanced Settings – Network & Security

항목	설명
VPC	DB 인스턴스를 위한 가상 네트워크 환경을 선택
Subnet Group	VPC에서 DB 인스턴스가 사용할 수 있는 서브넷과 IP 범위를 선택
Publicly Accessible	퍼블릭 IP 주소를 할당하려면 Yes, 그렇지 않으면 No
Availability Zone	DB 인스턴스가 생성될 가용 영역(Availability Zone) 선택 * 가용 영역: 지리적으로 세분화된 영역에 존재하는 복수의 격리된 영역
VPC Security Group(s)	DB 인스턴스에 접속할 수 있는 모든 EC2 인스턴스와 장치의 네트워크 액세스에 대한 보안 그룹을 한 개 이상 선택



Configure Advanced Settings - Database Options

항목	설명
DB Cluster Identifier	선택한 리전에 속한 계정 고유의 DB 클러스터 식별자로 클러스터 엔드포인트 주소로 사용됨 <ul style="list-style-type: none">1~63자의 영숫자 문자 또는 하이픈으로 구성첫 번째 문자는 글자하이픈으로 끝나거나 하이픈 2개 연속은 사용 불가리전별로 모든 DB 클러스터에 대해 AWS 계정당 고유해야 함
Database Name	기본 데이터베이스 이름. 최대 64자 영숫자 문자로 입력
Database Port	DB 인스턴스가 어플리케이션 커넥션에 사용할 TCP/IP 포트. 기본적으로 MySQL 포트 3306으로 지정됨.
DB Parameter Group	DB 인스턴스에 적용되는 파라미터 그룹
DB Cluster Parameter Group	DB 클러스터와 연관된 파라미터 그룹
Option Group	옵션 그룹
Enable IAM DB Authentication	yes: IAM 데이터베이스 인증을 활성화
Enable Encryption	yes: DB 클러스터에 대해 암호화 활성화
Master Key	해당 데이터베이스를 암호화 하는데 쓰이는 키를 보호함



Configure Advanced Settings

	항목	설명
Failover	Priority	인스턴스의 장애 조치 우선순위를 선택함. (기본 값은 tier-1) 기본 인스턴스 장애로부터 복원할 때, 이 우선순위에 따라 승격할 Aurora Replicas 순서가 결정됨
	Backup Retention Period	데이터베이스 백업 사본을 보존하는 기간을 1~35일로 선택. 백업 사본은 데이터베이스를 마지막 특정 시점으로 복구(PITR) 하는데 사용됨
Monitoring	Enable Enhanced Monitoring	Yes: DB 클러스터가 실행되는 운영체제에 대한 실시간 수집 측정치를 활성화
	Monitoring Role	[Default] 옵션을 선택하면, RDS에서 필요한 IAM 역할을 생성하고, RDS에서 rds-monitoring-role 역할을 자동으로 생성함
	Granularity	Yes: DB 클러스터에 대해 측정치를 수집하는 간격(초)을 설정함.
Maintenance	Auto Minor Version Upgrade	Yes: MySQL DB 엔진의 부(minor) 버전 업그레이드가 있을 때 Aurora DB 클러스터가 자동으로 업그레이드를 실행
	Maintenance Window	시스템 유지 관리를 실행하는 기간을 주 단위로 선택함.



Aurora DB 생성

8) Aurora DB 생성 완료 (Monitoring)

Endpoint

The screenshot shows the AWS RDS console interface for monitoring an Aurora MySQL instance. The instance is named 'e1' and is in an 'available' state. The 'Cluster Endpoint' is 'e1-cluster.rds.amazonaws.com:3306 (authorized)'. The monitoring section displays various metrics:

Metric	Current Value	Threshold	Last Hour
CPU	6.5%		
Memory	454 MB		
Select Throughput	1.77/sec		
DML Throughput	0.5/sec		
Select Latency	0.22 Milliseconds		
DML Latency	0.199 Milliseconds		



Aurora DB 생성

8) Aurora DB 생성 완료 (Details)

The screenshot displays the AWS RDS console interface for an Aurora MySQL DB instance. The instance is named 'e1' and is in an 'available' state. The console provides a comprehensive overview of the instance's configuration, including its engine, instance class, availability zone, and various security and network settings.

Engine	DB Instance	Status	CPU	Current Activity	Maintenance	Class	VPC	Multi-AZ	Replication Role	Encrypted
Aurora MySQL	e1	available	6.67%	2 Selects/sec	None	db.t2.small	vpc-d38a7bbb	No	writer	Yes

Cluster Endpoint: e1-cluster.rds.amazonaws.com:3306 (authorized)

Configuration Details	Security and Network	Instance and IOPS	Monitoring Details
ARN : arn:aws:rds:ap-northeast-2:638505169563:db:e1 Engine : Aurora MySQL 5.6.10a Created Time : September 18, 2017 at 10:01:35 AM UTC+9 DB Name : E1 Username : E1 Parameter Group : default.aurora5.6 (in-sync) DB Cluster Parameter Group : default.aurora5.6 (in-sync) Copy Tags To Snapshots : No Resource ID : PGEKDBANY7A6R4U4QWITQ0EC7U IAM DB Authentication Enabled : No	Availability Zone : ap-northeast-2c VPC : vpc-d38a7bbb Subnet Group : default (Complete) Subnets : subnet-819e51e9, subnet-933800de Security Groups : default (sg-133e397b) (active) Publicly Accessible : Yes Endpoint : e1. Port : 3306 Certificate Authority : rds-ca-2015 (Mar 5, 2020)	Instance Class : db.t2.small Storage Type : DB Cluster	Enhanced Monitoring Enabled : No

Encryption Details	Availability and Durability	Maintenance Details
Encryption Enabled : Yes KMS Key ID : [redacted]	DB Instance Status : available Multi AZ : No	Auto Minor Version Upgrade : Yes Maintenance Window : fri:15:52-fri:16:22 Pending Maintenance : None

Aliases: aws/rds
Description: Default master key that protects my RDS database volumes when no other key is defined

Instance Actions | **Tags** | **Logs**



Aurora DB 생성

8) Aurora DB 생성 완료 (Replication)

The screenshot displays the AWS RDS console interface. The main content area shows the details for an Aurora MySQL DB cluster. The cluster is in the 'available' state. The 'DB Cluster Details' section lists various configuration parameters, and the 'DB Cluster Members' table shows the cluster's composition.

DB INSTANCE	ROLE	ZONE	REPLICATION SOURCE	REPLICA LAG
e1	writer	ap-northeast-2c	e1-cluster	-

DB Cluster Details:

- ARN: arn:aws:rds:ap-northeast-2:638505169563:cluster:e1-cluster
- DB Cluster: e1-cluster (available)
- DB Cluster Role: Master
- Cluster Endpoint: e1-cluster.cl-...rds.amazonaws.com
- Reader Endpoint: e1-cluster.cl-...rds.amazonaws.com
- Port: 3306
- Automated Backups: Enabled (1 Day)
- Earliest Restorable Time: September 20, 2017 at 5:06:11 AM UTC+9
- Latest Restore Time: September 21, 2017 at 9:51:15 AM UTC+9
- Backup Window: 19:55-20:25
- Maintenance Window: thu:14:32-thu:15:02
- DB Cluster Parameter Group: default_aurora5.6
- Resource ID: cluster-
- A: A
- IAM DB Authentication Enabled: No



Aurora DB 클러스터 연결

1) MySQL Client

```
mysql -u[userid] -- port[port] --host[host] -p[password]
```

```
ex) mysql -uroot -- port 3306 -- host mycluster-cluster.cluster-0123456789012.ap-northeast-2.rds.amazonaws.com -paurora00
```

```
[mysql@localhost ~]$ mysql -uroot -- port 3306 -- host e1-cluster.cluster-0123456789012.ap-northeast-2.rds.amazonaws.com
-p
Enter password:

Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 485
Server version: 5.6.10 MySQL Community Server (GPL)

Copyright (c) 2000, 2016, Oracle and/or its affiliates. All rights reserved.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

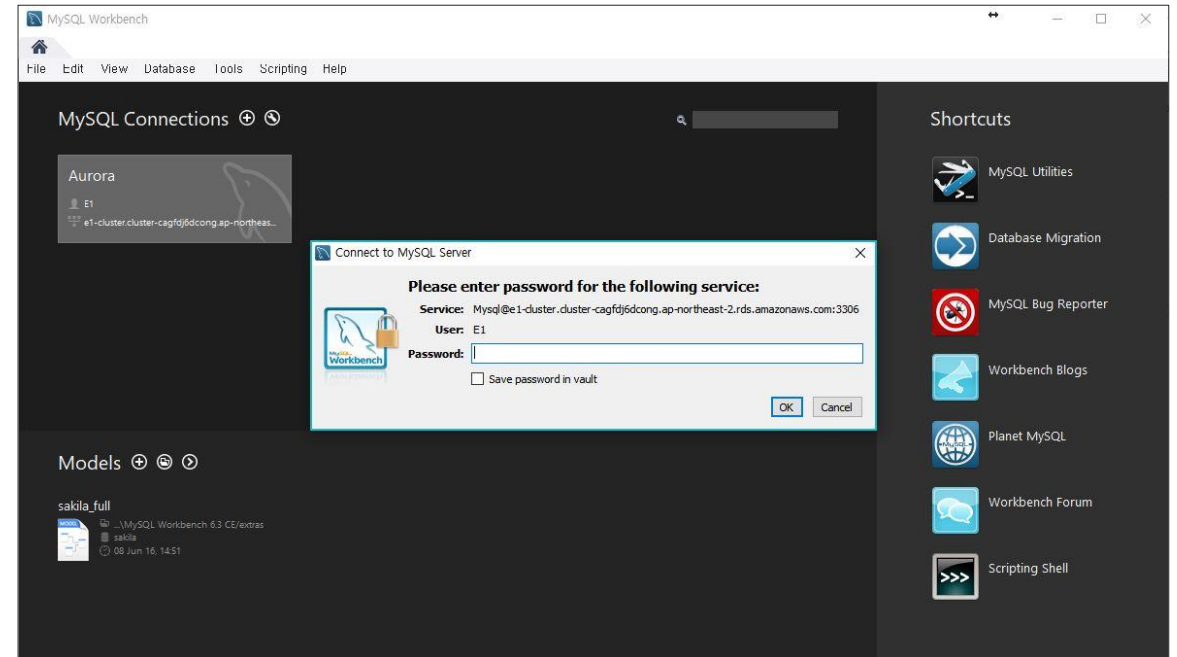
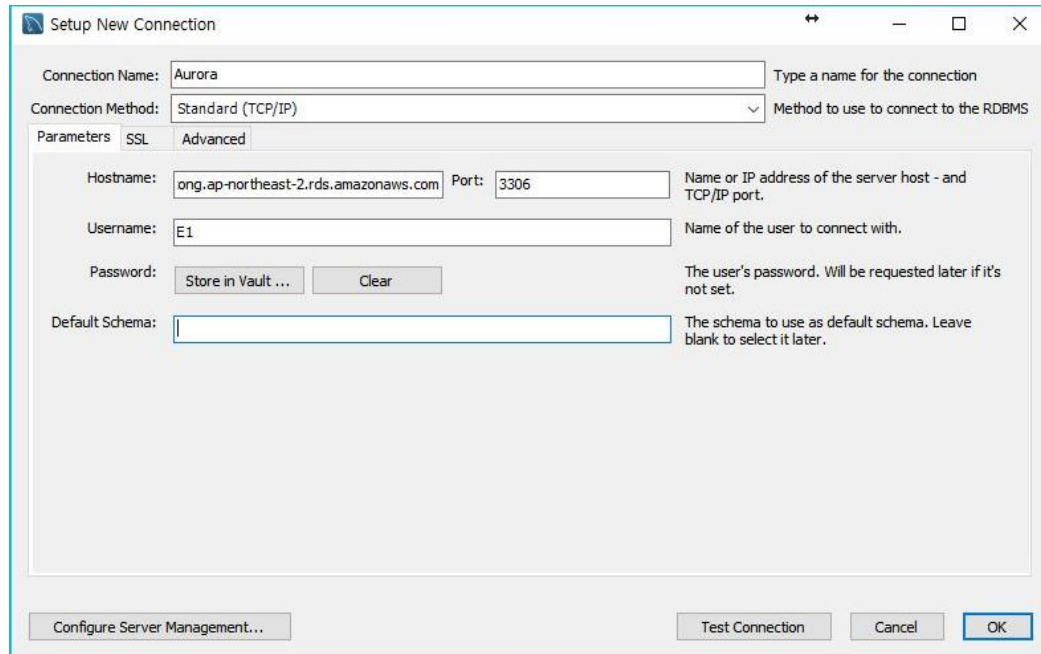
Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> show databases; -> "edu 확인"
```



Aurora DB 클러스터 연결

2) MySQL Workbench



ex)

Hostname: mycluster-cluster.cluster-0123456789012.ap-northeast-2.rds.amazonaws.com

Port: 3306

Username: edu

Password: aurora00



EC2 생성





EC2란?



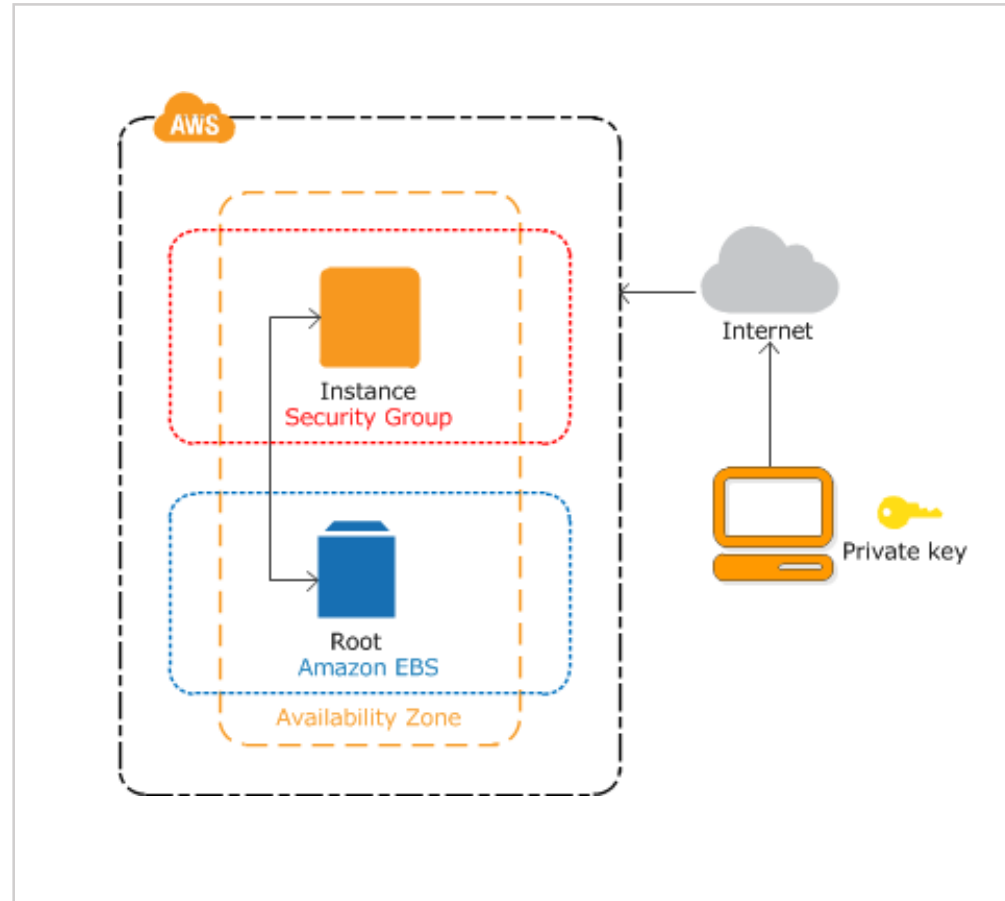
Amazon EC2

Amazon Elastic Compute Cloud

- AWS 클라우드에서 확장식 컴퓨팅을 제공
- 하드웨어에 선투자 필요 없음
- 빠르게 애플리케이션을 개발, 배포
- 원하는 만큼 가상 서버를 구축, 보안 및 네트워크 구성, 스토리지 관리 가능
- 트래픽 예측 필요성 줄어듦



EC2란?





EC2란?

Instances	가상 컴퓨팅 환경
AMI(Amazon Machine Image)	필요한 소프트웨어가 이미 구성되어 있는 템플릿
Instance Types	인스턴스를 위한 CPU, 메모리, 스토리지, 네트워킹 용량의 여러 가지 구성 제공
key pairs	인스턴스 로그인 정보 보호
Instance Store Volume	임시 데이터를 저장하는 스토리지 볼륨으로 인스턴스 종료시 삭제됨
Amazon EBS(Elastic Block Store)	영구 스토리지 볼륨에 데이터 저장
region, AZ(availability zone)	다른 물리적 장소에서 액세스
Security Groups	인스턴스에 연결할 수 있는 프로토콜, 포트, 소스 IP 범위를 지정하는 방화벽 기능
EIP (Elastic IP addresses)	동적 클라우드 컴퓨팅을 위한 고정 IPv4 주소
Tags	사용자가 생성하여 Amazon EC2 리소스에 할당할 수 있는 메타데이터



EC2 설치

Resources

You are using the following Amazon EC2 resources in the Asia Pacific (Seoul) region:

- 1 Running Instances
- 0 Elastic IPs
- 0 Dedicated Hosts
- 2 Snapshots
- 4 Volumes
- 0 Load Balancers
- 2 Key Pairs
- 7 Security Groups
- 0 Placement Groups

Just need a simple virtual private server? Get everything you need to jumpstart your project - compute, storage, and networking – for a low, predictable price. [Try Amazon Lightsail for free.](#)

Create Instance

To start using Amazon EC2 you will want to launch a virtual server, known as an Amazon EC2 instance.

[Launch Instance](#)

Note: Your instances will launch in the Asia Pacific (Seoul) region

Service Health

Service Status:

- Asia Pacific (Seoul): This service is operating normally

Availability Zone Status:

- ap-northeast-2a: Availability zone is operating normally
- ap-northeast-2c: Availability zone is operating normally

[Service Health Dashboard](#)

Scheduled Events

Asia Pacific (Seoul): No events



EC2 설치

The screenshot shows the AWS Management Console interface for selecting an Amazon Machine Image (AMI). The page title is "Step 1: Choose an Amazon Machine Image (AMI)". The search bar contains the text "centos". The "AWS Marketplace" filter is selected in the left-hand navigation pane. The search results display several AMI options:

- CentOS 7 (x86_64) - with Updates HVM**: 48 reviews, \$0.00/hr for software + AWS usage fees. Description: "This is the Official CentOS 7 x86_64 HVM image that has been built with a minimal profile, suitable for use in HVM instance types only. The image contains just enough packages to ..."
- CentOS 6 (x86_64) - with Updates HVM**: 33 reviews, \$0.00/hr for software + AWS usage fees. Description: "This is the Official CentOS 6 x86_64 HVM image that has been built with a minimal profile. The image contains just enough packages to run within AWS, bring up an SSH Server and ..."
- ProComputers Core powered by CentOS 6 (HVM)**: 0 reviews, \$0.00/hr for software + AWS usage fees. Description: "This is a minimal CentOS 6 (x86_64) installation, mainly used as a common base system on top of which all other appliances released by ProComputers.com are built. It contains just ..."
- CentOS 7.2 Minimal HVM**: 0 reviews, \$0.01/hr or \$79/yr (10% savings) for software + AWS usage fees. Description: "Unlike CentOS 7 AMI which comes up 6.X, this AMI comes up as a real CentOS 7.2 Minimal OS. It also runs as a Hardware Virtualized Machine (HVM), 64bit, SELinux Enabled, No product ..."
- CentOS 7.3 Minimal HVM**: (partially visible)



EC2 설치

aws Services Resource Groups EC2 Database Migration Service RDS CloudWatch IAM

Welcome to AWS Seoul Support

1. Choose AMI 2. Choose Instance Type 3. Configure Instance 4. Add Storage 5. Add Tags 6. Configure Security Group 7. Review

Step 1: Choose an Amazon Machine Image (AMI) Cancel and Exit

CentOS 7 (x86_64) - with Updates HVM

CentOS 7 (x86_64) - with Updates HVM

This is the Official CentOS 7 x86_64 HVM image that has been built with a minimal profile, suitable for use in HVM instance types only. The image contains just enough packages to run within AWS, bring up an SSH Server and allow users to login. Please note that this is the default CentOS-7 image that we recommend everyone uses. It contains ... [More info](#)

[View Additional Details in AWS Marketplace](#)

Free tier eligible

Product Details

- Sold by** Centos.org
- Customer Rating** ★★★★★ (48)
- Latest Version** 1708
- Base Operating System** Linux/Unix, CentOS 7
- Delivery Method** 64-bit Amazon Machine Image (AMI)
- License Agreement** [End User License Agreement](#)
- On Marketplace Since** 10/6/14

Highlights

- All official CentOS Linux images are built with SELINUX set to enforcing mode. However, we test the images with both Selinux enabled as well as permissive.

Pricing Details

Hourly Fees

Instance Type	Software	EC2	Total
R3 Eight Extra Large	\$0.00	\$3.192	\$3.192/hr
T2 Nano	\$0.00	\$0.007	\$0.007/hr
T2 Micro	\$0.00	\$0.012	\$0.012/hr
T2 Small	\$0.00	\$0.024	\$0.024/hr
T2 Medium	\$0.00	\$0.048	\$0.048/hr
T2 Large	\$0.00	\$0.096	\$0.096/hr
T2 Xlarge	\$0.00	\$0.192	\$0.192/hr
T2 2xlarge	\$0.00	\$0.384	\$0.384/hr
R4 16 Extra Large	\$0.00	\$5.12	\$5.12/hr
M4 Extra Large	\$0.00	\$0.246	\$0.246/hr
M4 Large	\$0.00	\$0.492	\$0.492/hr
M4 Medium	\$0.00	\$0.738	\$0.738/hr
M4 Small	\$0.00	\$0.984	\$0.984/hr
High I/O Quadruple Extra Large	\$0.00	\$1.464	\$1.464/hr
T2 Large	\$0.00	\$0.115	\$0.115/hr
C4 Double Extra Large	\$0.00	\$0.454	\$0.454/hr
C4 Extra Large	\$0.00	\$0.908	\$0.908/hr
C4 Large	\$0.00	\$1.362	\$1.362/hr
C4 Medium	\$0.00	\$1.816	\$1.816/hr
C4 Small	\$0.00	\$2.270	\$2.270/hr
R3 Double Extra Large	\$0.00	\$0.798	\$0.798/hr
X1 32 Extra Large	\$0.00	\$19.341	\$19.341/hr
T2 Double Extra Large	\$0.00	\$0.461	\$0.461/hr
T2 Extra Large	\$0.00	\$0.23	\$0.23/hr
High I/O Extra Large	\$0.00	\$1.001	\$1.001/hr
C4 Eight Extra Large	\$0.00	\$1.815	\$1.815/hr
M4 Quadruple Extra Large	\$0.00	\$0.984	\$0.984/hr

[Cancel](#) [Continue](#)

Feedback English (US) © 2008 - 2017, Amazon Web Services, Inc. or its affiliates. All rights reserved. Privacy Policy Terms of Use



EC2 설치

aws Services Resource Groups EC2 Database Migration Service RDS CloudWatch IAM

Welcome to AWS Seoul Support

1. Choose AMI 2. Choose Instance Type 3. Configure Instance 4. Add Storage 5. Add Tags 6. Configure Security Group 7. Review

Step 2: Choose an Instance Type

Amazon EC2 provides a wide selection of instance types optimized to fit different use cases. Instances are virtual servers that can run applications. They have varying combinations of CPU, memory, storage, and networking capacity, and give you the flexibility to choose the appropriate mix of resources for your applications. [Learn more](#) about instance types and how they can meet your computing needs.

Filter by: All instance types Current generation Show/Hide Columns

Currently selected: t2.micro (Variable ECUs, 1 vCPUs, 2.5 GHz, Intel Xeon Family, 1 GiB memory, EBS only)
 Note: The vendor recommends using a t2.micro instance (or larger) for the best experience with this product.

	Family	Type	vCPUs	Memory (GiB)	Instance Storage (GB)	EBS-Optimized Available	Network Performance	IPv6 Support
<input type="checkbox"/>	General purpose	t2.nano	1	0.5	EBS only	-	Low to Moderate	Yes
<input checked="" type="checkbox"/>	General purpose	t2.micro <small>Free tier eligible</small>	1	1	EBS only	-	Low to Moderate	Yes
<input type="checkbox"/>	General purpose	t2.small	1	2	EBS only	-	Low to Moderate	Yes
<input type="checkbox"/>	General purpose	t2.medium	2	4	EBS only	-	Low to Moderate	Yes
<input type="checkbox"/>	General purpose	t2.large	2	8	EBS only	-	Low to Moderate	Yes
<input type="checkbox"/>	General purpose	t2.xlarge	4	16	EBS only	-	Moderate	Yes
<input type="checkbox"/>	General purpose	t2.2xlarge	8	32	EBS only	-	Moderate	Yes
<input type="checkbox"/>	General purpose	m4.large	2	8	EBS only	Yes	Moderate	Yes
<input type="checkbox"/>	General purpose	m4.xlarge	4	16	EBS only	Yes	High	Yes
<input type="checkbox"/>	General purpose	m4.2xlarge	8	32	EBS only	Yes	High	Yes
<input type="checkbox"/>	General purpose	m4.4xlarge	16	64	EBS only	Yes	High	Yes
<input type="checkbox"/>	General purpose	m4.10xlarge	40	160	EBS only	Yes	10 Gigabit	Yes
<input type="checkbox"/>	General purpose	m4.16xlarge	64	256	EBS only	Yes	25 Gigabit	Yes
<input type="checkbox"/>	Compute optimized	c4.large	2	3.75	EBS only	Yes	Moderate	Yes
<input type="checkbox"/>	Compute optimized	c4.xlarge	4	7.5	EBS only	Yes	High	Yes

Cancel Previous Review and Launch Next: Configure Instance Details

Feedback English (US) © 2008 - 2017, Amazon Web Services, Inc. or its affiliates. All rights reserved. Privacy Policy Terms of Use



EC2 설치

aws Services Resource Groups EC2 Database Migration Service RDS CloudWatch IAM

Welcome to AWS Seoul Support

1. Choose AMI 2. Choose Instance Type 3. Configure Instance 4. Add Storage 5. Add Tags 6. Configure Security Group 7. Review

Step 3: Configure Instance Details

Configure the instance to suit your requirements. You can launch multiple instances from the same AMI, request Spot instances to take advantage of the lower pricing, assign an access management role to the instance, and more.

Number of instances [Launch into Auto Scaling Group](#)

Purchasing option Request Spot instances

Network [Create new VPC](#)

Subnet [Create new subnet](#)

Auto-assign Public IP

IAM role [Create new IAM role](#)

Shutdown behavior

Enable termination protection Protect against accidental termination

Monitoring Enable CloudWatch detailed monitoring
[Additional charges apply.](#)

Tenancy
[Additional charges will apply for dedicated tenancy.](#)

▼ **Advanced Details**

User data As text As file Input is already base64 encoded

(Optional)

[Cancel](#) [Previous](#) [Review and Launch](#) [Next: Add Storage](#)



EC2 설치

aws Services Resource Groups EC2 Database Migration Service RDS CloudWatch IAM

Welcome to AWS Seoul Support

1. Choose AMI 2. Choose Instance Type 3. Configure Instance 4. Add Storage 5. Add Tags 6. Configure Security Group 7. Review

Step 4: Add Storage

Your instance will be launched with the following storage device settings. You can attach additional EBS volumes and instance store volumes to your instance, or edit the settings of the root volume. You can also attach additional EBS volumes after launching an instance, but not instance store volumes. [Learn more](#) about storage options in Amazon EC2.

Volume Type	Device	Snapshot	Size (GiB)	Volume Type	IOPS	Throughput (MB/s)	Delete on Termination	Encrypted
Root	/dev/sda1	snap-0603349e290384e0c	8	General Purpose SSD (GP2)	100 / 3000	N/A	<input type="checkbox"/>	Not Encrypted

[Add New Volume](#)

Free tier eligible customers can get up to 30 GB of EBS General Purpose (SSD) or Magnetic storage. [Learn more](#) about free usage tier eligibility and usage restrictions.

[Cancel](#) [Previous](#) [Review and Launch](#) [Next: Add Tags](#)

Feedback English (US) © 2008 - 2017, Amazon Web Services, Inc. or its affiliates. All rights reserved. Privacy Policy Terms of Use



EC2 설치

The screenshot shows the AWS Management Console interface for configuring an EC2 instance. The top navigation bar includes the AWS logo and various service links: Services, Resource Groups, EC2, Database Migration Service, RDS, CloudWatch, and IAM. The user is logged in as 'Welcome to AWS' from 'Seoul'.

The main content area is titled 'Step 5: Add Tags' and includes a progress bar with seven steps: 1. Choose AMI, 2. Choose Instance Type, 3. Configure Instance, 4. Add Storage, 5. Add Tags (current step), 6. Configure Security Group, and 7. Review. Below the title, a text block explains that a tag is a case-sensitive key-value pair and provides a link to learn more about tagging EC2 resources.

A table for adding tags is shown with two columns: 'Key' (127 characters maximum) and 'Value' (255 characters maximum). Below the table, a message states 'This resource currently has no tags' and provides instructions to use the 'Add tag button' or 'click to add a Name tag', along with a note to ensure the IAM policy includes permissions to create tags.

An 'Add Tag' button is located on the left side of the table, with a note '(Up to 50 tags maximum)'. At the bottom of the page, there are navigation buttons: 'Cancel', 'Previous', 'Review and Launch', and 'Next: Configure Security Group' (which is highlighted with a red border).

The footer contains a 'Feedback' link, the language 'English (US)', and copyright information: '© 2008 - 2017, Amazon Web Services, Inc. or its affiliates. All rights reserved. Privacy Policy Terms of Use'.



EC2 설치

aws Services Resource Groups EC2 Database Migration Service RDS CloudWatch IAM

Welcome to AWS Seoul Support

1. Choose AMI 2. Choose Instance Type 3. Configure Instance 4. Add Storage 5. Add Tags 6. Configure Security Group 7. Review

Step 6: Configure Security Group

A security group is a set of firewall rules that control the traffic for your instance. On this page, you can add rules to allow specific traffic to reach your instance. For example, if you want to set up a web server and allow Internet traffic to reach your instance, add rules that allow unrestricted access to the HTTP and HTTPS ports. You can create a new security group or select from an existing one below. [Learn more](#) about Amazon EC2 security groups.

Assign a security group: Create a new security group Select an existing security group

Security group name: CentOS 7 -x86_64- - with Updates HVM-1708-AutogenByAWSMP-

Description: This security group was generated by AWS Marketplace and is based on recomr

Type	Protocol	Port Range	Source	Description
SSH	TCP	22	Custom 0.0.0.0/0	e.g. SSH for Admin Desktop

Warning
Rules with source of 0.0.0.0/0 allow all IP addresses to access your instance. We recommend setting security group rules to allow access from known IP addresses only.

Feedback English (US) © 2008 - 2017, Amazon Web Services, Inc. or its affiliates. All rights reserved. Privacy Policy Terms of Use



EC2 설치

Step 7: Review Instance Launch
Please review your instance launch details. You can go back to edit changes for each section. Click **Launch** to assign a key pair to your instance and complete the launch process.

AMI Details [Edit AMI](#)

CentOS 7 (x86_64) - with Updates HVM
Free tier eligible
CentOS Linux 7 x86_64 HVM EBS 1708_01
Root Device Type: ebs Virtualization type: hvm

Hourly Software Fees: \$0.00 per hour on t2.micro instance (Additional taxes may apply.)
Software charges will begin once you launch this AMI and continue until you terminate the instance.

By launching this product, you will be subscribed to this software and agree that your use of this software is subject to the pricing terms and the seller's [End User License Agreement](#)

Instance Type [Edit instance type](#)

Instance Type	ECUs	vCPUs	Memory (GiB)	Instance Storage (GB)	EBS-Optimized Available	Network Performance
t2.micro	Variable	1	1	EBS only	-	Low to Moderate

Security Groups [Edit security groups](#)

Security group name: CentOS 7 -x86_64- - with Updates HVM-1708-AutogenByAWSMP-
Description: This security group was generated by AWS Marketplace and is based on recommended settings for CentOS 7 (x86_64) - with Updates HVM version 1708 provided by Centos.org

Type	Protocol	Port Range	Source	Description
This security group has no rules				

Instance Details [Edit instance details](#)

Number of instances: 1 **Purchasing option:** On demand

- Network:** vpc-d38a7bbb
- Subnet:** No preference (default subnet in any Availability Zone)
- EBS-optimized:** No
- Monitoring:** No
- Termination protection:** No
- Shutdown behavior:** Stop
- IAM role:** None

[Cancel](#) [Previous](#) [Launch](#)



EC2 설치

Step 7: Review Instance Launch

Please review your instance launch details. You can go back to edit changes for each section. Click **Launch** to assign a key pair to your instance and complete the launch process.

AMI Details

CentOS 7 (x86_64) - with Updates HVM
CentOS Linux 7 x86_64 HVM EBS 1708_01
Root Device Type: ebs Virtualization type: hvm

Hourly Software Fees: \$0.00 per hour on t2.micro instance (Additional taxes may apply)
Software charges will begin once you launch this AMI and continue until you terminate the instance.

By launching this product, you will be subscribed to this software and agree that you accept the [End User License Agreement](#).

Instance Type

Instance Type	ECUs	vCPUs	Memory (GiB)	Instance Profile
t2.micro	Variable	1	1	EBS

Security Groups

Security group name: CentOS 7 -x86_64- - with Updates HVM-1708-AutogenBy...
Description: This security group was generated by AWS Marketplace an...

Type	Protocol	Port Range

Instance Details

- Number of instances: 1
- Network: vpc-d38a7bbb
- Subnet: No preference (default subnet in any Availability Zone)
- EBS-optimized: No
- Monitoring: No
- Termination protection: No
- Shutdown behavior: Stop
- IAM role: None

Select an existing key pair or create a new key pair

A key pair consists of a **public key** that AWS stores, and a **private key file** that you store. Together, they allow you to connect to your instance securely. For Windows AMIs, the private key file is required to obtain the password used to log into your instance. For Linux AMIs, the private key file allows you to securely SSH into your instance.

Note: The selected key pair will be added to the set of keys authorized for this instance. Learn more about [removing existing key pairs from a public AMI](#).

Create a new key pair
Key pair name
ec2
Download Key Pair

You have to download the **private key file** (*.pem file) before you can continue. **Store it in a secure and accessible location.** You will not be able to download the file again after it's created.

Cancel Launch Instances

Cancel Previous Launch

Create a new Key pair
Key pair name 입력
Download Key Pair



EC2 설치

Launch Status

Your instances are now launching
The following instance launches have been initiated: i-0ad2581b6ef3bbe1e [View launch log](#)

Get notified of estimated charges
Create billing alerts to get an email notification when estimated charges on your AWS bill exceed an amount you define (for example, if you exceed the free usage tier).

How to connect to your instances

Your instances are launching, and it may take a few minutes until they are in the **running** state, when they will be ready for you to use. Usage hours on your new instances will start immediately and continue to accrue until you stop or terminate your instances. Click **View Instances** to monitor your instances' status. Once your instances are in the **running** state, you can **connect** to them from the Instances screen. [Find out](#) how to connect to your instances.

▼ **Getting started with your software**

To get started with CentOS 7 (x86_64) - with Updates HVM [View Usage Instructions](#)

To manage your software subscription [Open Your Software on AWS Marketplace](#)

▼ **Here are some helpful resources to get you started**

- [How to connect to your Linux instance](#)
- [Learn about AWS Free Usage Tier](#)
- [Amazon EC2: User Guide](#)
- [Amazon EC2: Discussion Forum](#)

While your instances are launching you can also

- [Create status check alarms](#) to be notified when these instances fail status checks. (Additional charges may apply)
- [Create and attach additional EBS volumes](#) (Additional charges may apply)
- [Manage security groups](#)

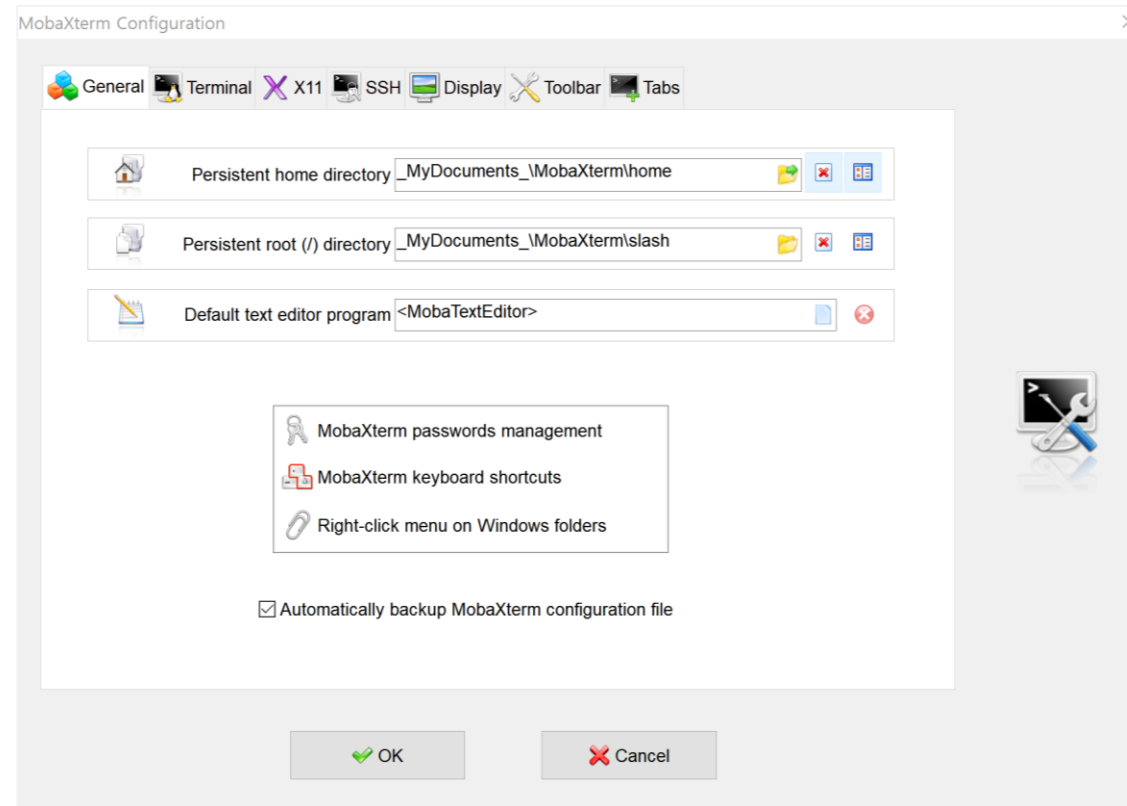
[View Instances](#)

Feedback English (US) © 2008 - 2017, Amazon Web Services, Inc. or its affiliates. All rights reserved. Privacy Policy Terms of Use



MobaXterm을 이용하여 접속

Key를 ctrl+c 해서 mobaXterm home directory로 이동





ssh를 이용한 ec2 접속

Usage Instructions: SSH to the instance and login as 'centos' using the key specified at launch.
Additional information may be found at : <http://wiki.centos.org/Cloud/AWS>.

```
[KSJ.DESKTOP-ADNO4PJ] ➤ ssh centos@ec2-13-124-193-136.ap-northeast-2.compute.amazonaws.com -i ~/ec2.pem
```

```
X11 forwarding request failed on channel 0
```

```
Last login: Thu Sep 28 10:55:15 2017 from 115.178.73.2
```

```
  ____|  |____|_ )  
  _| (_____/   Amazon Linux AMI  
  ___|\___|___|
```

```
https://aws.amazon.com/amazon-linux-ami/2016.03-release-notes/
```

```
22 package(s) needed for security, out of 94 available
```

```
Run "sudo yum update" to apply all updates.
```

```
Amazon Linux version 2017.03 is available.
```

```
[ec2-user@ip-172-31-16-116 ~]$ sudo su -
```

```
Last login: Fri Sep 29 00:26:19 UTC 2017 on pts/1
```



EC2 CentOS7 - MySQL 5.7 설치

1. ssh로 접속

```
ssh centos@ec2-52-78-240-107.ap-northeast-2.compute.amazonaws.com -i ~/ec2.pem
```

2. root 계정으로 이동

```
sudo su -
```

3. yum 업데이트 실행

```
sudo yum update -y
```

4. centos 디렉토리로 이동

```
cd /home/centos
```



EC2 CentOS7 – MySQL 5.7 설치

5. wget 설치

* 이때 주의할 것! sudo yum install mysql 를 입력하면 mariadb가 설치됨.

```
sudo yum install wget
```

```
Resolving Dependencies
```

```
--> Running transaction check
```

```
---> Package wget.x86_64 0:1.14-15.el7_4.1 will be installed
```

```
--> Finished Dependency Resolution
```

```
Dependencies Resolved
```

```
=====
```

Package	Arch	Version	Repository	Size
Installing:				
wget	x86_64	1.14-15.el7_4.1	updates	547 k

```
=====
```

```
Transaction Summary
```

```
=====
```

Install 1 Package
Total download size: 547 k
Installed size: 2.0 M
Is this ok [y/d/N]:y

```
Downloading packages:
```

```
wget-1.14-15.el7_4.1.x86_64.rpm | 547 kB 00:00:00
```

```
Running transaction check
```

```
Running transaction test
```

```
Transaction test succeeded
```

```
Running transaction
```

```
Installing : wget-1.14-15.el7_4.1.x86_64 1/1
```

```
Verifying : wget-1.14-15.el7_4.1.x86_64 1/1
```

```
Installed:
```

```
wget.x86_64 0:1.14-15.el7_4.1
```

```
Complete!
```



EC2 CentOS7 - MySQL 5.7 설치

6. mysql rpm wget으로 가져오기 - 리스트에 추가할 패키지를 가져옴

(* 이때 버전 확인은 <https://dev.mysql.com/downloads/repo/yum/>)

```
[root@ip centos]# wget http://dev.mysql.com/get/mysql57-community-release-el7-8.noarch.rpm
```

```
--2017-12-01 05:41:39-- http://dev.mysql.com/get/mysql57-community-release-el7-8.noarch.rpm
Resolving dev.mysql.com (dev.mysql.com)... 137.254.60.11
Connecting to dev.mysql.com (dev.mysql.com)|137.254.60.11|:80... connected.
HTTP request sent, awaiting response... 301 Moved Permanently
Location: https://dev.mysql.com/get/mysql57-community-release-el7-8.noarch.rpm [following]
--2017-12-01 05:41:40-- https://dev.mysql.com/get/mysql57-community-release-el7-8.noarch.rpm
Connecting to dev.mysql.com (dev.mysql.com)|137.254.60.11|:443... connected.
HTTP request sent, awaiting response... 302 Found
Location: https://repo.mysql.com//mysql57-community-release-el7-8.noarch.rpm [following]
--2017-12-01 05:41:40-- https://repo.mysql.com//mysql57-community-release-el7-8.noarch.rpm
Resolving repo.mysql.com (repo.mysql.com)... 184.31.44.242
Connecting to repo.mysql.com (repo.mysql.com)|184.31.44.242|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 9116 (8.9K) [application/x-redhat-package-manager]
Saving to: 'mysql57-community-release-el7-8.noarch.rpm'
100%[=====>] 9,116  --.-K/s  in 0s
2017-12-01 05:41:41 (259 MB/s) - 'mysql57-community-release-el7-8.noarch.rpm' saved [9116/9116]
```



EC2 CentOS7 - MySQL 5.7 설치

7. yum list에 MySQL 5.7 버전 추가

```
[root@ip-172-31-15-153 centos]# sudo yum localinstall mysql57-community-release-el7-8.noarch.rpm
```

```
Loaded plugins: fastestmirror
Examining mysql57-community-release-el7-8.noarch.rpm: mysql57-community-release-el7-8.noarch
Marking mysql57-community-release-el7-8.noarch.rpm to be installed
Resolving Dependencies
--> Running transaction check
---> Package mysql57-community-release.noarch 0:el7-8 will be installed
--> Finished Dependency Resolution
Dependencies Resolved
```

Package	Arch	Version	Repository	Size
Installing:				
mysql57-community-release	noarch	el7-8	/mysql57-community-release-el7-8.noarch	8.2 k

```
Transaction Summary
=====
Install 1 Package
Total size: 8.2 k
Installed size: 8.2 k
Is this ok [y/d/N]:y
Downloading packages:
Running transaction check
Running transaction test
Transaction test succeeded
Running transaction
Installing : mysql57-community-release-el7-8.noarch      1/1
Verifying  : mysql57-community-release-el7-8.noarch      1/1
Installed:
  mysql57-community-release.noarch 0:el7-8
Complete
```




EC2 CentOS7 - MySQL 5.7 설치

8. MySQL 5.7 설치

```
[root@ip-172-31-15-153 centos]# sudo yum install mysql-community-server
```

```

Loaded plugins: fastestmirror
mysql-connectors-community           | 2.5 kB  00:00
mysql-tools-community                | 2.5 kB  00:00
mysql57-community                    | 2.5 kB  00:00
(1/3): mysql-connectors-community/x86_64/primary_db | 16 kB  00:00
(2/3): mysql-tools-community/x86_64/primary_db     | 37 kB  00:00
(3/3): mysql57-community/x86_64/primary_db         | 124 kB 00:00
Loading mirror speeds from cached hostfile
* base: data.nicehosting.co.kr
* extras: data.nicehosting.co.kr
* updates: data.nicehosting.co.kr
Resolving Dependencies
--> Running transaction check
--> Package mysql-community-server.x86_64:5.7.20-1.el7 will be installed
--> Processing Dependency: mysql-community-common(x86-64) = 5.7.20-1.el7 for package: mysql-community-server-5.7.20-1.el7.x86_64
--> Processing Dependency: mysql-community-client(x86-64) >= 5.7.9 for package: mysql-community-server-5.7.20-1.el7.x86_64
--> Processing Dependency: libaio.so.1(LIBAIO_0.4)(64bit) for package: mysql-community-server-5.7.20-1.el7.x86_64
--> Processing Dependency: libaio.so.1(LIBAIO_0.1)(64bit) for package: mysql-community-server-5.7.20-1.el7.x86_64
--> Processing Dependency: libaio.so.1(64bit) for package: mysql-community-server-5.7.20-1.el7.x86_64
--> Running transaction check
--> Package libaio.x86_64:0.3.109-13.el7 will be installed
--> Package mariadb.x86_64:1:5.5.56-2.el7 will be obsoleted
--> Package mysql-community-client.x86_64:5.7.20-1.el7 will be obsoleting
--> Processing Dependency: mysql-community-libs(x86-64) >= 5.7.9 for package: mysql-community-client-5.7.20-1.el7.x86_64
--> Package mysql-community-common.x86_64:5.7.20-1.el7 will be installed
--> Running transaction check
--> Package mariadb-libs.x86_64:1:5.5.56-2.el7 will be obsoleted
--> Processing Dependency: libmysqlclient.so.18(64bit) for package: 2:postfix-2.10.1-6.el7.x86_64
--> Processing Dependency: libmysqlclient.so.18(libmysqlclient_18)(64bit) for package: 2:postfix-2.10.1-6.el7.x86_64
--> Package mysql-community-libs.x86_64:5.7.20-1.el7 will be obsoleting
--> Running transaction check
--> Package mysql-community-libs-compat.x86_64:5.7.20-1.el7 will be obsoleting
--> Finished Dependency Resolution
Dependencies Resolved

=====
Package      Arch Version Repository Size
=====
Installing:
mysql-community-client x86_64 5.7.20-1.el7 mysql57-community 24 M
  replacing mariadb.x86_64:1:5.5.56-2.el7
mysql-community-libs x86_64 5.7.20-1.el7 mysql57-community 2.1 M
  replacing mariadb-libs.x86_64:1:5.5.56-2.el7
mysql-community-libs-compat x86_64 5.7.20-1.el7 mysql57-community 2.0 M
  replacing mariadb-libs.x86_64:1:5.5.56-2.el7
mysql-community-server x86_64 5.7.20-1.el7 mysql57-community 164 M
Installing for dependencies:
libaio x86_64 0.3.109-13.el7 base 24 k
mysql-community-common x86_64 5.7.20-1.el7 mysql57-community 272 k
Transaction Summary
=====

```

```

[Install 4 Packages (+2 Dependent packages)
Total download size: 192 M
Is this ok [y/d/N]: y
Downloading packages:
(1/6): libaio-0.3.109-13.el7.x86_64.rpm | 24 kB 00:00
warning: /var/cache/yum/x86_64/7/mysql57-community/packages/mysql-community-common-5.7.20-1.el7.x86_64.rpm: Header V3 DSA/SHA1 Signature, key ID 5072e1f5: NOKEY
Public key for mysql-community-common-5.7.20-1.el7.x86_64.rpm is not installed
(2/6): mysql-community-common-5.7.20-1.el7.x86_64.rpm | 272 kB 00:00
(3/6): mysql-community-libs-5.7.20-1.el7.x86_64.rpm | 2.1 MB 00:00
(4/6): mysql-community-libs-compat-5.7.20-1.el7.x86_64.rpm | 2.0 MB 00:00
(5/6): mysql-community-client-5.7.20-1.el7.x86_64.rpm | 24 MB 00:09
(6/6): mysql-community-server-5.7.20-1.el7.x86_64.rpm | 164 MB 00:42
-----
Total                4.4 MB/s | 192 MB 00:43
Retrieving key from file:///etc/pki/rpm-gpg/RPM-GPG-KEY-mysql
Importing GPG key 0x5072E1F5:
  Userid   : "MySQL Release Engineering <mysql-build@oss.oracle.com>"
  Fingerprint: a4a9 4068 76fc bd3c 4567 70c8 8c71 8d3b 5072 e1f5
  Package   : mysql57-community-release-el7-8.noarch (installed)
  From      : /etc/pki/rpm-gpg/RPM-GPG-KEY-mysql
Is this ok [y/N]: y
Running transaction check
Running transaction test
Transaction test succeeded
Running transaction
  Installing : mysql-community-common-5.7.20-1.el7.x86_64           1/8
  Installing : mysql-community-libs-5.7.20-1.el7.x86_64           2/8
  Installing : mysql-community-client-5.7.20-1.el7.x86_64         3/8
  Installing : libaio-0.3.109-13.el7.x86_64                       4/8
  Installing : mysql-community-server-5.7.20-1.el7.x86_64         5/8
  Installing : mysql-community-libs-compat-5.7.20-1.el7.x86_64    6/8
  Erasing    : 1:mariadb-5.5.56-2.el7.x86_64                       7/8
  Erasing    : 1:mariadb-libs-5.5.56-2.el7.x86_64                 8/8
  Verifying  : mysql-community-libs-5.7.20-1.el7.x86_64           1/8
  Verifying  : mysql-community-server-5.7.20-1.el7.x86_64        2/8
  Verifying  : mysql-community-common-5.7.20-1.el7.x86_64        3/8
  Verifying  : mysql-community-client-5.7.20-1.el7.x86_64        4/8
  Verifying  : mysql-community-libs-compat-5.7.20-1.el7.x86_64    5/8
  Verifying  : libaio-0.3.109-13.el7.x86_64                       6/8
  Verifying  : 1:mariadb-libs-5.5.56-2.el7.x86_64                 7/8
  Verifying  : 1:mariadb-5.5.56-2.el7.x86_64                     8/8
Installed:
mysql-community-client.x86_64:5.7.20-1.el7
mysql-community-libs.x86_64:5.7.20-1.el7
mysql-community-libs-compat.x86_64:5.7.20-1.el7
mysql-community-server.x86_64:5.7.20-1.el7
Dependency Installed:
libaio.x86_64:0.3.109-13.el7 mysql-community-common.x86_64:5.7.20-1.el7
Replaced:
mariadb.x86_64:1:5.5.56-2.el7 mariadb-libs.x86_64:1:5.5.56-2.el7
Complete!

```



EC2 CentOS7 - MySQL 5.7 설치

9. mysqld 상태 확인

```
[root@ip-172-31-15-153 centos]# sudo service mysql status
Redirecting to /bin/systemctl status mysqld.service
● mysqld.service - MySQL Server
  Loaded: loaded (/usr/lib/systemd/system/mysqld.service; enabled; vendor preset: disabled)
  Active: inactive (dead)
  Docs: man:mysqld(8)
        http://dev.mysql.com/doc/refman/en/using-systemd.html
```

10. mysqld 시작

```
[root@ip-172-31-15-153 centos]# sudo service mysql start

Redirecting to /bin/systemctl start mysqld.service
```

11. root@localhost 임시 비밀번호 조회

```
[root@ip-172-31-15-153 centos]# sudo grep 'temporary password' /var/log/mysqld.log

2017-12-01T05:55:17.840596Z 1 [Note] A temporary password is generated for root@localhost: *=f+zh2Ur_7+
```



EC2 CentOS7 – MySQL 5.7 설치

12. mysql 접속

```
[root@ip-172-31-15-153 centos]# mysql -uroot -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 5
Server version: 5.7.20
```

Copyright (c) 2000, 2017, Oracle and/or its affiliates. All rights reserved.

Oracle is a registered trademark of Oracle Corporation and/or its affiliates. Other names may be trademarks of their respective owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

```
mysql> select now();
ERROR 1820 (HY000): You must reset your password using ALTER USER statement before executing this statement.
```

13. root 비밀번호 변경

(validate_password_policy 파라미터가 Medium으로 설정되어 있어서 비밀번호 복잡도가 높음)

```
mysql> alter user 'root'@'localhost' identified by '*Oracle00*';
Query OK, 0 rows affected (0.00 sec)
```



EC2 CentOS7 – MySQL 5.7 설치

14. MySQL 접속

```
[root@ip-172-31-15-153 ~]# mysql -hlocalhost -uroot -p*Oracle00*
mysql: [Warning] Using a password on the command line interface can be insecure.
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 10
Server version: 5.7.20 MySQL Community Server (GPL)
```

Copyright (c) 2000, 2017, Oracle and/or its affiliates. All rights reserved.

Oracle is a registered trademark of Oracle Corporation and/or its affiliates. Other names may be trademarks of their respective owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

15. validate_password plugin 해제

```
mysql> uninstall plugin validate_password;
Query OK, 0 rows affected (0.00 sec)
```

cf) PLUGIN 재설정: `INSTALL PLUGIN plugin_name SONAME 'shared_library_name'`

(참조: <https://stackoverflow.com/questions/36301100/how-do-i-turn-off-the-mysql-password-validation>)

16. 비밀번호 변경

```
mysql> set password for 'root'@'localhost'=PASSWORD('oracle');
Query OK, 0 rows affected, 1 warning (0.01 sec)
```



EC2 - Oracle Client 설치

1. Download instantclient, rlwrap

```
[centos@ip-172-31-15-153 ~]$ ls -l /tmp
total 60200
-rwx-----. 1 centos centos 60704657 Dec 26 04:20 instantclient-basic-linux.x64-11.2.0.4.0.zip
-rwx-----. 1 centos centos 834491 Dec 26 02:21 instantclient-sqlplus-linux.x64-11.2.0.4.0.zip
-rw-r--r--. 1 root root 98608 Nov 29 2014 rlwrap-0.42-1.el7.x86_64.rpm
```

2. install rlwrap (as root)

```
[root@ip-172-31-15-153 ~]# yum localinstall /tmp/rlwrap-0.42-1.el7.x86_64.rpm
```

```
Loaded plugins: fastestmirror
Examining /tmp/rlwrap-0.42-1.el7.x86_64.rpm: rlwrap-0.42-1.el7.x86_64
Marking /tmp/rlwrap-0.42-1.el7.x86_64.rpm to be installed
Resolving Dependencies
--> Running transaction check
---> Package rlwrap.x86_64 0:0.42-1.el7 will be installed
--> Processing Dependency: perl(Data::Dumper) for package: rlwrap-0.42-1.el7.x86_64
Loading mirror speeds from cached hostfile
* base: ftp.daumkakao.com
* extras: data.nicehosting.co.kr
* updates: data.nicehosting.co.kr
--> Running transaction check
---> Package perl-Data-Dumper.x86_64 0:2.145-3.el7 will be installed
--> Finished Dependency Resolution
```

Dependencies Resolved

Package	Arch	Version	Repository	Size
Installing: rlwrap	x86_64	0.42-1.el7	/rlwrap-0.42-1.el7.x86_64	209 k

```
Installing for dependencies:
perl-Data-Dumper x86_64 2.145-3.el7 base 47 k
Transaction Summary
=====
Install 1 Package (+1 Dependent package)

Total size: 256 k
Total download size: 47 k
Installed size: 306 k
Is this ok [y/d/N]: y
Downloading packages:
perl-Data-Dumper-2.145-3.el7.x86_64.rpm | 47 kB 00:00:00
Running transaction check
Running transaction test
Transaction test succeeded
Running transaction
  Installing : perl-Data-Dumper-2.145-3.el7.x86_64 1/2
  Installing : rlwrap-0.42-1.el7.x86_64 2/2
  Verifying : perl-Data-Dumper-2.145-3.el7.x86_64 1/2
  Verifying : rlwrap-0.42-1.el7.x86_64 2/2

Installed:
rlwrap.x86_64 0:0.42-1.el7

Dependency Installed:
perl-Data-Dumper.x86_64 0:2.145-3.el7

Complete!
```



EC2 - Oracle Client 설치

3. install instantclient (as centos)

```
[centos@ip-172-31-15-153 ~]$ unzip /tmp/instantclient-basic-linux.x64-11.2.0.4.0.zip
```

```
Archive: /tmp/instantclient-basic-linux.x64-11.2.0.4.0.zip
```

```
inflating: instantclient_11_2/BASIC_README
```

```
inflating: instantclient_11_2/adrci
```

```
inflating: instantclient_11_2/genezi
```

```
inflating: instantclient_11_2/libclntsh.so.11.1
```

```
inflating: instantclient_11_2/libnnz11.so
```

```
inflating: instantclient_11_2/libocci.so.11.1
```

```
inflating: instantclient_11_2/libociei.so
```

```
inflating: instantclient_11_2/libocijdbc11.so
```

```
inflating: instantclient_11_2/ojdbc5.jar
```

```
inflating: instantclient_11_2/ojdbc6.jar
```

```
inflating: instantclient_11_2/uidrvci
```

```
inflating: instantclient_11_2/xstreams.jar
```

```
[centos@ip-172-31-15-153 ~]$ unzip /tmp/instantclient-sqlplus-linux.x64-11.2.0.4.0.zip
```

```
Archive: /tmp/instantclient-sqlplus-linux.x64-11.2.0.4.0.zip
```

```
inflating: instantclient_11_2/SQLPLUS_README
```

```
inflating: instantclient_11_2/glogin.sql
```

```
inflating: instantclient_11_2/libsqlplus.so
```

```
inflating: instantclient_11_2/libsqlplusic.so
```

```
inflating: instantclient_11_2/sqlplus
```



EC2 - Oracle Client 설치

4. editing .bash_profile

```
[centos@ip-172-31-15-153 ~]$ cat ~/.bash_profile
# .bash_profile

# Get the aliases and functions
if [ -f ~/.bashrc ]; then
    . ~/.bashrc
fi

# User specific environment and startup programs

PATH=$PATH:$HOME/.local/bin:$HOME/bin

export PATH

set -o vi

export ORACLE_HOME=/home/centos/instantclient_11_2
export LD_LIBRARY_PATH=$ORACLE_HOME
export PATH=$PATH:$ORACLE_HOME
```

5. run sqlplus

```
[centos@ip-172-31-15-153 tmp]$ rlwrap sqlplus system/oracle@//ec2-13-125-14-185.ap-northeast-2.compute.amazonaws.com:1521/B203
```

```
SQL*Plus: Release 11.2.0.4.0 Production on Tue Dec 26 07:05:39 2017
```

```
Copyright (c) 1982, 2013, Oracle. All rights reserved.
```

```
Connected to:
Oracle Database 11g Enterprise Edition Release 11.2.0.4.0 - 64bit Production
With the Partitioning, OLAP, Data Mining and Real Application Testing options
```

```
SQL>
```



Aurora 명령어 실습





Aurora 명령어 실습

항목	명령어
인스턴스 접속	mysql -u[유저ID] --port[포트번호] --host[호스트/엔드포인트] -p 비밀번호
비밀번호 변경	콘솔에서 변경. [Amazon RDS] - [Instances] - [edu1] - [Instance actions] - [Modify] - [New master password]
현재 사용자 확인	select user();
현재 데이터베이스 확인	select database();
데이터베이스 현재 상태	status
데이터베이스 조회	show databases; (* 설치 시 생성했던 데이터베이스 확인)
데이터베이스 변경	use [데이터베이스명];
데이터베이스 생성	create database [데이터베이스명];
데이터베이스 삭제	drop database [데이터베이스명];
데이터베이스 생성 쿼리 보기	show create database [데이터베이스명];



Aurora 명령어 실습

항목	명령어
사용자 조회	<code>select * from user;</code>
사용자 추가	<code>insert into user (host,user,password,ssl_cipher,x509_issuer,x509_subject) values ('localhost','user1',password('aurora00'),'', '');</code>
사용자 추가 및 권한 부여	<p>1) localhost <code>grant all privileges on [데이터베이스명].* to [사용자명]@'localhost' identified by '[비밀번호]';</code> ex) <code>grant all privileges on mysql.* to 'user2'@'localhost' identified by 'aurora00';</code></p> <p>2) localhost를 제외한 모든 호스트 <code>grant all privileges on [데이터베이스명].* to [사용자명]@'%' identified by 'password';</code></p>
사용자 삭제	<code>drop user [사용자명] from user;</code>
권한 뺏기	<code>revoke all privileges *.* from '[사용자명]'@'localhost';</code>
테이블 생성	<code>create table [테이블명] ([컬럼명] [타입명]);</code>



Aurora 명령어 실습

항목	명령어
테이블 생성 쿼리 보기	show create table [테이블명];
테이블 조회	show tables;
특정 테이블 조회	show table status from [데이터베이스명] like '테이블명';
테이블 구조 보기	1) desc [테이블명]; 2) explain [테이블명];
테이블 정보 삭제	delete from [테이블명];
테이블 삭제	drop table [테이블명];
Insert	insert into [테이블명] values(값1, 값2, ...); insert into [테이블명] (컬럼명1, 컬럼명2, ...) values (값1, 값2, ...);
Delete	delete from [테이블명] where 조건;
Update	update [테이블명] set 컬럼명1=새로운 값 where 조건;



Aurora 명령어 실습

항목	명령어
컬럼 추가	alter table [테이블명] add [컬럼명] [타입];
특정 컬럼 뒤 컬럼 추가	alter table [테이블명] add [컬럼명] [타입] after [특정 컬럼명];
컬럼 삭제	alter table [테이블명] drop [컬럼명];
변수 확인	show variables; ex) show variables like '%log%'; show global variables;
변수 변경	set global [variable]
버전 확인	select @@version; select version();
베이스 디렉토리 확인	select @@basedir;
데이터 디렉토리 확인	select @@datadir;
오류 보기	show errors;
경고 보기	show warnings;
프로세스 리스트 확인	show processlist;
프로세스 죽이기	kill [프로세스 ID];



Aurora vs. Oracle





Aurora VS Oracle



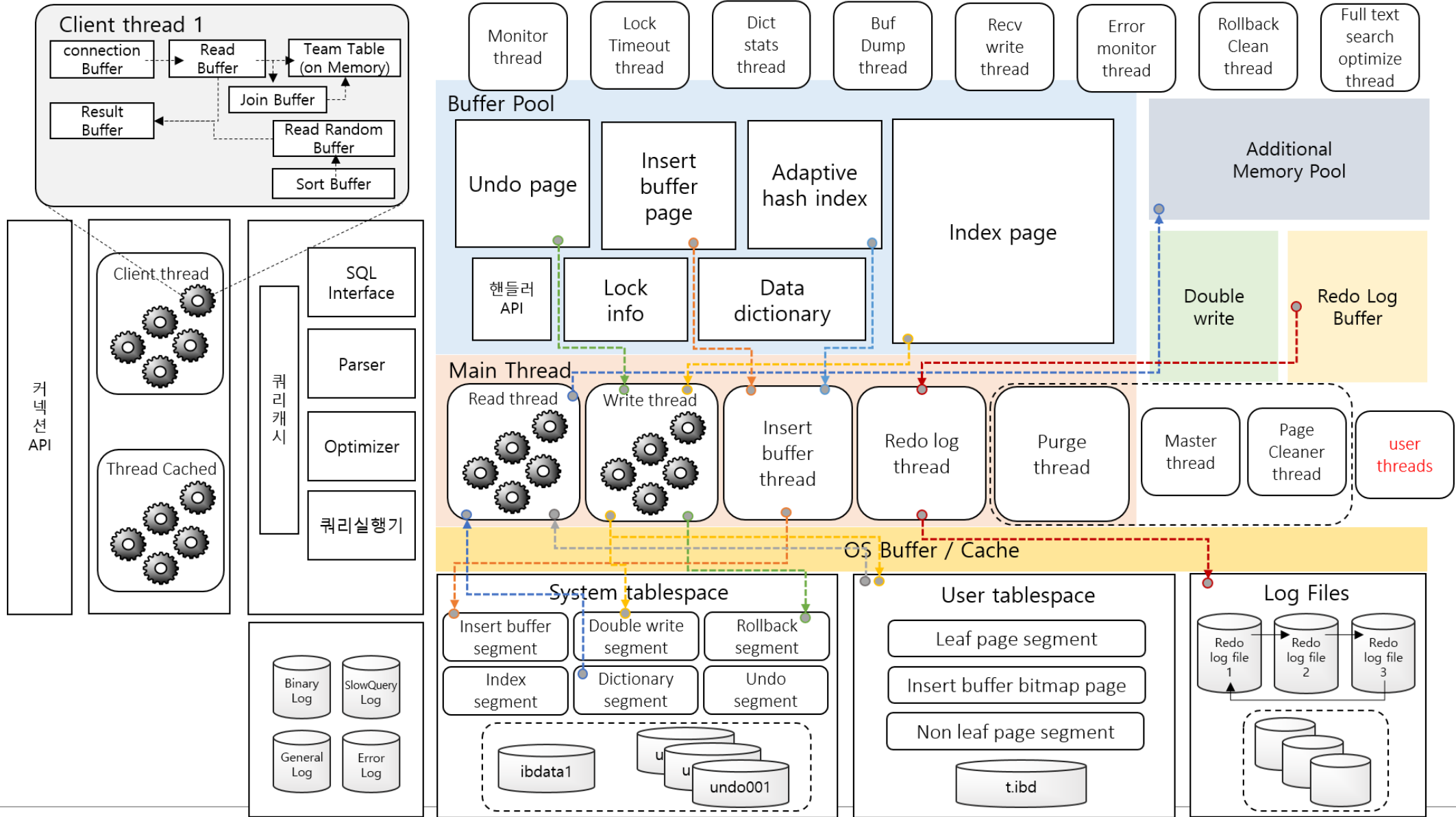
Microsoft Excel
워크시트

- 1) Architecture
- 2) Memory Cache, Files
- 3) Undo, Redo, Security Items
- 4) SQL
- 5) 집합함수
- 6) Data Types
- 7) Parameter Group
- 8) PL/SQL Partitioning Methods
- 9) Hint
- 10) Waitevent Interface
- 11) Backup & Recovery
- 12) Isolation level
- 13) sqlload, exp & imp
- 14) 지원 언어
- 15) Online Rebuild 가능 여부
- 16) Aurora 통계정보 관리
- 17) Storage engine



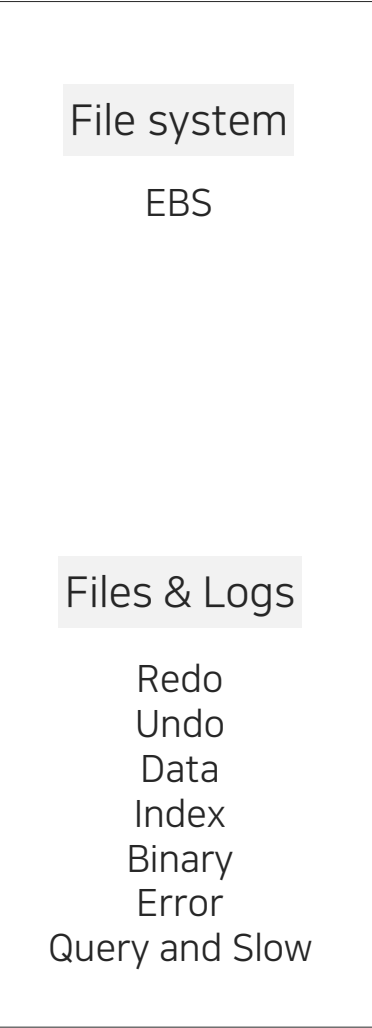
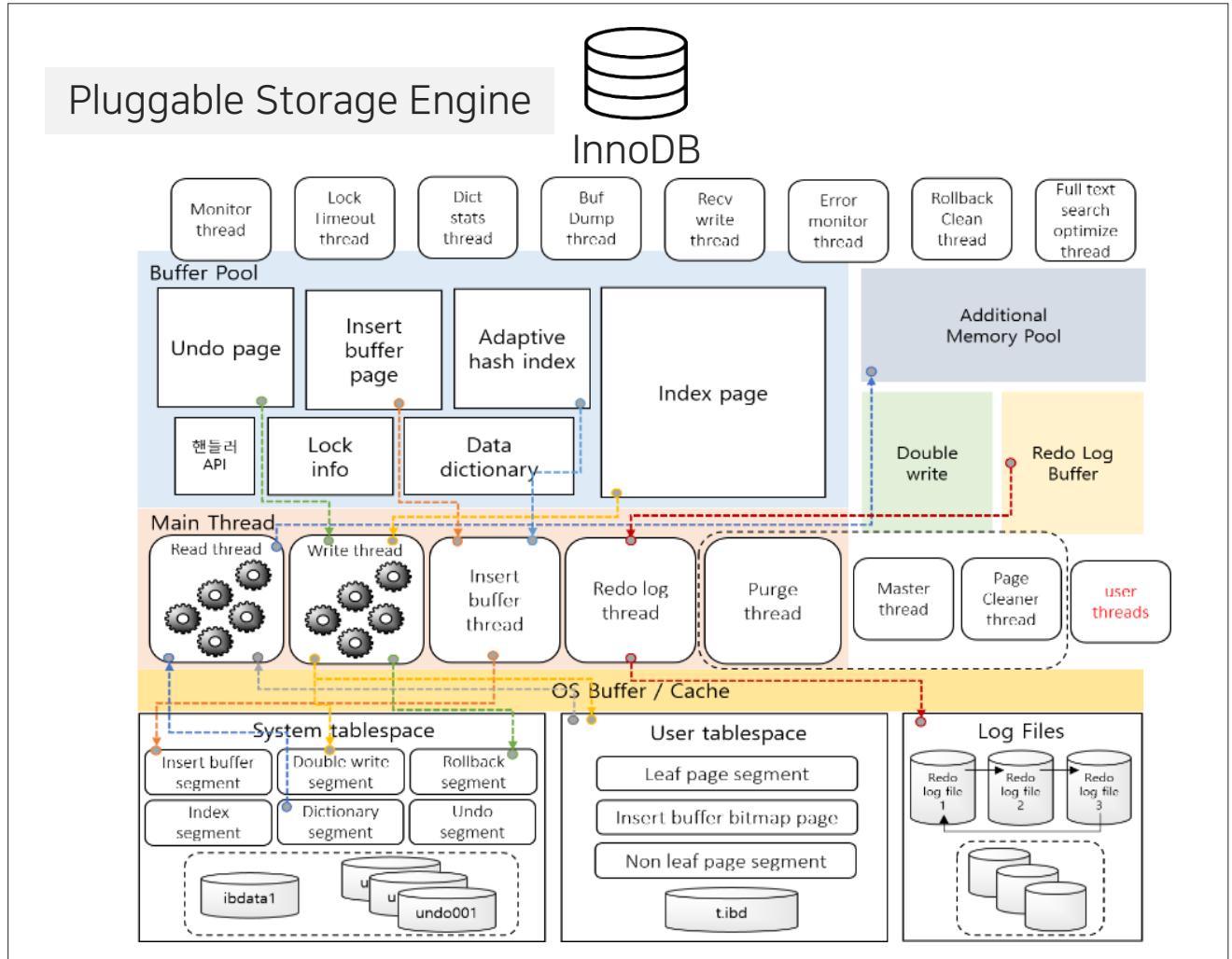
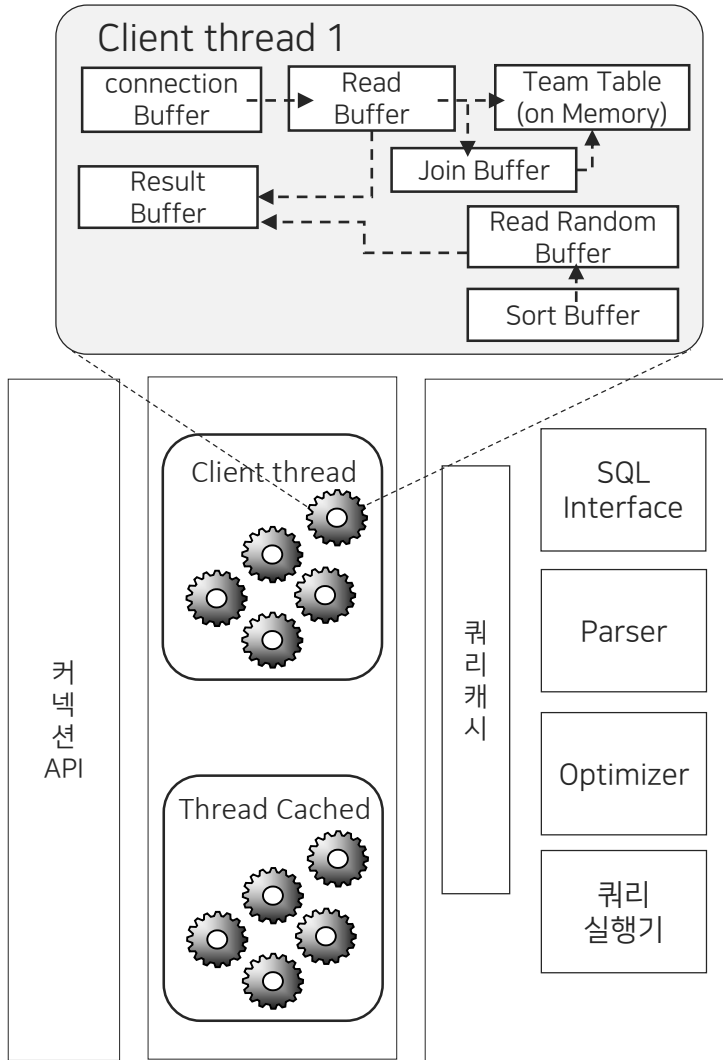
1. InnoDB Architecture

InnoDB 엔진을 Aurora Cluster Volume 위에 customizing하여 구성
Aurora의 File System은 EBS를 사용하지 않으며, Quorum system을 차용한 Cluster Shared Volume



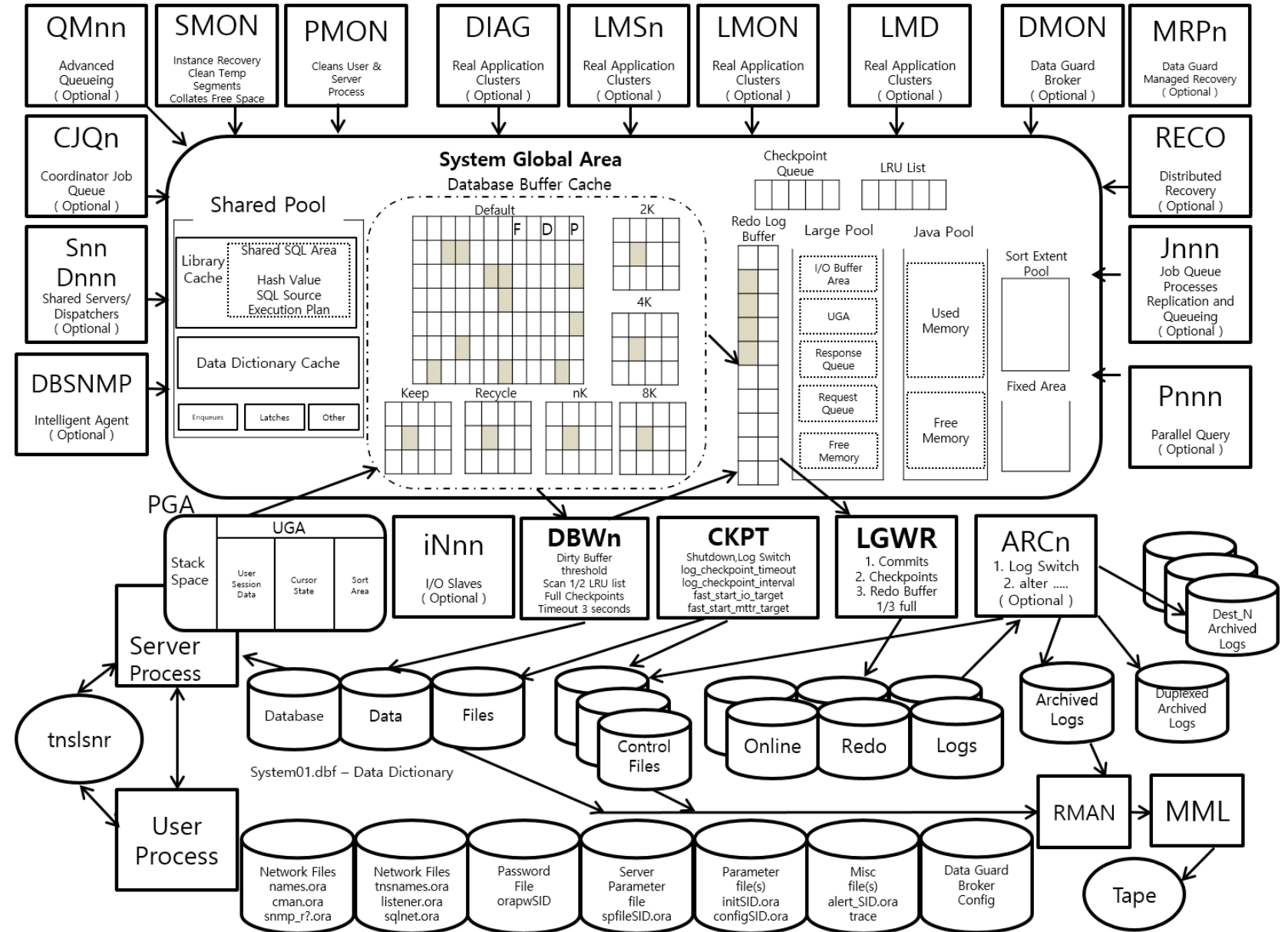


1. Architecture – Aurora





1. Architecture - Oracle





2. Memory Cache, Files

		Oracle	Aurora
Memory Cache	PGA (per server process)	Hash area	Join Buffer (per client session)
		Sort area	Sort Buffer (per client session)
		Stack area	stack (per client session)
	SGA	Shared pool	(Query Cache)
		DB Buffer Cache	InnoDB Buffer Cache
		Redo Log Buffer	InnoDB Log Buffer
Files	init.ora	Parameter Group	
	System tablespace file	ibdata1	
	User tablespace files	table.ibd	



3. Redo/Undo, Security Items

	Oracle	Aurora
Redo / Undo Logs	Redo Logs	InnoDB Redo logs
	Undo tablespace files	ibdata1 - page
Built-in Accounts	SYS, SYSTEM, others <code>select username from dba_users;</code>	rdsadmin (로그인 X) <code>select host,user from mysql.user;</code>
Object Privileges Handled	GRANT, REVOKE	GRANT, REVOKE, Host consideration
Privilege Granularity	Column, Row	Column (row via view)
Group Management	ROLES	On Roadmap
SSL Supported	Yes	Yes
Encryption	Transparent, RC4, DES, AES, etc.	AES, DES, MD5 SHA1; 128/256
Auditing	Audit subsystem	Handled via logging



4. SQL

	Oracle	Aurora
1. 타입	NUMBER create table t1 (c1 number) ;	<ul style="list-style-type: none"> • INTEGER, INT, SMALLINT, TINYINT, MEDIUMINT, BIGINT • DECIMAL, NUMERIC • FLOAT, DOUBLE create table t1 (c1 integer, c2 int, c3 smallint, c4 tinyint, c5 mediumint, c6 bigint) ;
	VARCHAR2 create table t2 (c1 varchar2(10)); insert into t2 values ('가나다라'); ERROR at line 1: ORA-12899: value too large for column "SYSTEM"."T2"."C1" (actual: 12, maximum: 10)	VARCHAR() (괄호안에는 characters(자릿수), max: 65,532 bytes) create table t2 (c1 varchar(10)); insert into t2 values ('가나다라');
	CHAR create table t3 (c1 char(10)); insert into t3 values ('가나다라');	CHAR () (max: 255 characters) create table t3 (c1 char(10)); insert into t3 values ('가나다라');
	DATE create table t4 (c1 date); insert into t4 values (sysdate);	DATETIME create table t4(c1 datetime); insert into t4 values ('가나다라');

source: <http://blog.naver.com/steelcatx/60186190415>
<http://sarc.io/index.php/mariadb/605-dbms-migration-oracle-to-mysql-mariadb>



4. SQL

	Oracle	Aurora
2. 함수	NVL select nvl('컬럼명', '') from dual; select nvl(null,'o') as c1 from dual;	IFNULL select ifnull('컬럼명', '') from dual; select ifnull(null,'o');
	SYSDATE select sysdate as c1 from dual;	NOW() select now();
	TRUNC select trunc(sysdate) as c1 from dual;	CURDATE() (시간 없이 일자만 나옴) select curdate();
	DECODE select decode('Y','Y',1,0) as c1 from dual ;	CASE select case 'Y' when 'Y' then 1 else 0 end as c1 ;
	TO_CHAR, TO_NUMBER select to_char(1234) from dual;	CAST select cast(1234 as char) from dual;

source: <http://blog.naver.com/steelcatx/60186190415>
<http://sarc.io/index.php/mariadb/605-dbms-migration-oracle-to-mysql-mariadb>



4. SQL

	Oracle	Aurora
3. 시퀀스	<p>사용자 함수 사용 (시퀀스명. nextval)</p> <pre>create table emp1 (empno number(4) primary key, ename varchar(10), hiredate date); insert into emp1 values (emp_seq.nextval, 'Julia', sysdate); select * from emp1;</pre>	<ul style="list-style-type: none"> • auto-increment 기능을 사용하여 고유한 식별자를 자동으로 채움 • 고유한 식별자(함수 또는 유사한 것을 사용)를 생성하는 대체 방법을 만들고, 시퀀스에 대한 참조를 새 함수로 변경 <pre>create table emp1(empno int(11) not null auto_increment primary key, ename varchar(20) not null hiredate datetime); insert into emp1(ename, hiredate) values ('Alice', now());</pre>
4. 힌트 지원	많음	적음
5. ROWNUM	<p>rownum은 고정 값이 아닌 동적으로 부여되는 번호 rownum을 가져오는 쿼리를 미리 해야 함</p> <pre>select * from EMP where rownum <= 10; select * from (select rownum numrow, EMP.* from EMP) where numrow > 3 and numrow <=5;</pre>	<p>limit 으로 대체</p> <pre>SELECT * FROM \${TABLE_NAME} LIMIT 5,10; LIMIT { [offset,] rowcount} -> limit 5, 10 LIMIT rowcount OFFSET offset -> limit 10 offset 5</pre>



4. SQL

	Oracle	Aurora
6. 날짜	TO_CHAR (SYSDATE, 'YYYYMMDD') select to_char(sysdate, 'YYYYMMDD') from dual;	DATE_FORMAT(NOW(), '%Y%m%d'); select date_format(now(), '%Y %c/%e');
	TO_DATE (날짜, 'YYYYMMDD')	STR_TO_DATE (날짜, '%Y-%m-%d');
	TO_CHAR(SYSDATE, 'D') -1 1(일요일)~7 (토요일) 자바스크립트가 0~6으로 인식하기에 Oracle 쿼리에서 -1을 해서 맞추는 경우가 많음 select to_char(sysdate, 'D')-1 from dual;	DATE_FORMAT(NOW(), '%w') 0(일요일)~6(토요일) select date_format(now(), '%w');
7. 문자열 가르기	1. 3번째 자리부터 끝까지 select substr('exem1234',3) from dual; 2. 3번째 자리부터 3글자 select substr('exem1234',3,3) from dual; 3. 뒤에서 3글자 select substr('exem1234',-3) from dual;	1. substring('문자열',1,3) = substr('문자열',1,3) 오라클 예제 1,2,3 과 동일 2. left('문자열',3) select left ('exem1234',3); 3. right('문자열',3) select right ('exem1234',3);



4. SQL

	Oracle	Aurora
8. 문자열 합치기	<p>문자열(또는 컬럼) ' - '</p> <pre>select 'exem' '1234' from dual; select concat('exem','1234') from dual;</pre>	<p>CONCAT(문자열(또는 컬럼), ' - ') SQL_MODE에 PIPES_AS_CONCAT 지정하면 오라클 처럼 로 문자열 합치기 가능</p> <pre>set sql_mode=PIPES_AS_CONCAT; select 'exem' '1234'; select concat('exem','1234');</pre>
9. START WITH CONNECT BY	지원됨	WITH문을 이용한 RECURSIVE 쿼리로 변경
10. 서브쿼리	<p>지원됨</p> <pre>update EMP set sal = sal + 100 where sal < (select sal from (select * from EMP where empno=7839));</pre>	<p>서브 쿼리에 alias가 없다면 붙여야 함</p> <pre>update EMP set sal = sal + 100 where sal < (select sal from (select * from EMP where empno=7839));</pre> <p>ERROR 1248 (42000): Every derived table must have its own alias</p> <pre>update EMP set sal = sal + 100 where sal < (select sal from (select * from EMP where empno=7839)s);</pre>



4. SQL

	Oracle	Aurora
11. 예약어가 컬럼명일 때	컬럼명을 double quotation(" ")으로 감싸기 <pre>create table t5("FROM" number(8) null, "TO" number(8) null); insert into t5 values (1,1); insert into t5 values (2,2); select "FROM" from t5 where "TO"=2;</pre>	컬럼명을 backquote(` `)로 감싸기 <pre>create table t5(`from` int(8), `to` int(8)); insert into t5(`from`,`to`) values (1,1),(2,2); select `from` from t5 where `to`=2;</pre>
12. DELETE	DELETE [테이블명] <pre>delete t5; delete from t5;</pre>	DELETE FROM [테이블명] 으로 수정 <pre>delete from t5;</pre>
13. 테이블명 대소문자 구분	구분 X <pre>create table t6 (c1 number(8), c2 number(8)); insert into t4 values (1,1); select * from T6;</pre>	기본적으로 구분하나 설정으로 변경 가능 <pre>create table t6 (c1 int(8), c2 int(8)); select * from T6;</pre> <p>ERROR 1146 (42S02): Table 'aurora.T6' doesn't exist</p>



4. SQL

		Oracle	Aurora
14. OUTER JOIN	1) LEFT OUTER JOIN	<pre>SELECT A.*, B.* FROM A, B WHERE A.ID = B.ID(+);</pre> <pre>select EMP.*, DEPT.* from EMP, DEPT where EMP.DEPTNO=DEPT.DEPTNO(+);</pre> <pre>select EMP.*, DEPT.* from EMP LEFT OUTER JOIN DEPT on EMP.DEPTNO=DEPT.DEPTNO;</pre>	<p>ANSI 스타일</p> <pre>SELECT A.*, B.* FROM A LEFT OUTER JOIN B ON A.ID = B.ID;</pre> <pre>select EMP.*, DEPT.* from EMP LEFT OUTER JOIN DEPT on EMP.DEPTNO=DEPT.DEPTNO;</pre>
	2) RIGHT OUTER JOIN	<pre>SELECT A.*, B.* FROM A, B WHERE A.ID(+) = B.ID;</pre> <pre>select EMP.*, DEPT.* from EMP, DEPT where EMP.DEPTNO(+)=DEPT.DEPTNO;</pre> <pre>select EMP.*, DEPT.* from EMP RIGHT OUTER JOIN DEPT on EMP.DEPTNO=DEPT.DEPTNO;</pre>	<pre>SELECT A.*, B.* FROM A RIGHT OUTER JOIN B ON A.ID = B.ID;</pre> <pre>select EMP.*, DEPT.* from EMP RIGHT OUTER JOIN DEPT on EMP.DEPTNO=DEPT.DEPTNO;</pre>



4. SQL

	Oracle	Aurora
15. 테이블 리스트 확인	<pre>select tablespace_name, table_name from all_tables;</pre>	<pre>select table_schema, table_name from information_schema.tables; select table_schema, table_name from information_schema.tables where table_schema='aurora';</pre>
16. 저장 프로시저 있는지 여부 파악해서 Create 하기	<pre>CREATE OR REPLACE PROCEDURE 프로시저명</pre>	<pre>DROP PROCEDURE IF EXISTS 프로시저명; 을 한 뒤에 CREATE PROCEDURE 프로시저명</pre>



5. 집합함수

항목	설명
AVG()	Return the average value of the argument
BIT_AND()	Return bitwise AND
BIT_OR()	Return bitwise OR
BIT_XOR()	Return bitwise XOR
COUNT()	Return a count of the number of rows returned
COUNT(DISTINCT)	Return the count of a number of different values
GROUP_CONCAT()	Return a concatenated string
MAX()	Return the maximum value
MIN()	Return the minimum value
STD()	Return the population standard deviation
STDDEV()	Return the population standard deviation
STDDEV_POP()	Return the population standard deviation
STDDEV_SAMP()	Return the sample standard deviation
SUM()	Return the sum
VAR_POP()	Return the population standard variance
VAR_SAMP()	Return the sample variance
VARIANCE()	Return the population standard variance



6. Data Types

1) Numeric Data Types

Name	Description	Range	Synonym	Oracle
TINYINT	A very small integer	-128 to 127 signed	BOOLEAN	NUMBER(3, 0)
SMALLINT	A small integer	-32768 to 32767 signed		NUMBER(5, 0)
MEDIUMINT	A medium-sized integer	-8388608 to 8388607 signed		NUMBER(7, 0)
INT	A normal-size integer	-2147483648 to 2147483647 signed	INTEGER	NUMBER(10, 0)
BIGINT	A large integer	-9223372036854775808 to 9223372036854775807 signed		NUMBER(19, 0)
DECIMAL	A packed "exact" fixed-point number		DEC, NUMERIC, FIXED	FLOAT (24)
FLOAT	A small (single-precision) floating-point number			FLOAT
DOUBLE	Normal-size (double-precision) floating-point number	<ul style="list-style-type: none">-1.7976931348623157E+308 to -2.2250738585072014E-30802.2250738585072014E-308 to 1.7976931348623157E+308	DOUBLE PRECISION, REAL	FLOAT (24)
BIT	A bit-field type	1 to 64		RAW



6. Data Types

2) String Data Types

Name	Description	Synonym	Oracle
CHAR	Fixed-length string		CHAR
VARCHAR	Variable-length string		VARCHAR2, CLOB
BINARY	Fixed-length binary byte string	CHAR BYTE	
VARBINARY	Variable-length binary byte string		
TINYBLOB	Tiny binary large object up to 255 bytes		RAW
BLOB	Binary large object up to 65,535 bytes		BLOB, RAW
MEDIUMBLOB	Medium binary large object upto 16,777,215 bytes		BLOB, RAW
LOB	Long BLOB holding up to 4GB		BLOB, RAW
TINYTEXT	A TEXT column with a maximum length of 255 characters		VARCHAR2
TEXT	A TEXT column with a maximum length of 65,535 characters		VARCHAR2, CLOB
MEDIUMTEXT	A TEXT column with a maximum length of 16,777,215 characters		CLOB, RAW
LONGTEXT	A TEXT column with a maximum length of 4,294,967,295 characters		CLOB, RAW
ENUM	Enumeration or string object that can have one value chosen from a list of values		VARCHAR2



6. Data Types

3) Date and Time Data Types

Name	Description	Format	Range	Oracle
DATE	A date	'YYYY-MM-DD'	'1000-01-01' to '9999-12-31'	DATE
TIME	A time	'HH:MM:SS. sssss'	'-838:59:59.999999' to '838:59:59.999999'	DATE
DATETIME	A date and time combination	'YYYY-MM-DD HH:MM:SS'	'1000-01-01 00:00:00.000000' to '9999-12-31 23:59:59.999999'	DATE
TIMESTAMP	A timestamp	'YYYY-MM-DD HH:MM:DD'	'1970-01-01 00:00:01' (UTC) to '2038- 01-19 05:14:07'	DATE
YEAR	A year in two-digit or four- digit format	'YY', 'YYYY'	<ul style="list-style-type: none">• 70 to 69• 1901 to 2155, and 0000	NUMBER



7. Parameter Group (1)

Variable	Is Modifiable	Description
autocommit	0	autocommit 모드 설정, 1인 경우 autocommit 설정, 0인 경우 disable
binlog_cache_size	0	트랜잭션 실행동안 바이너리 로그를 저장하기 위한 캐시 사이즈
binlog_format	0	바이너리 로그의 포맷을 설정
datadir	X	data directory 설정
innodb_undo_logs	0	롤백 세그먼트 개수
key_buffer_size	0	설정
log_bin	X	바이너리 로그를 생성할지의 여부를 설정
slow_query_log	0	long_query_time보다 오래 실행되는 쿼리를 로그로 남기는 작업을 설정
long_query_time	0	slow query 기준이 되는 값(초)
max_allowed_packet	0	가장 크게 생성될 수 있는 packet 사이즈를 설정함.
max_binlog_size	X	바이너리 로그 파일의 최대 크기 설정값



7. Parameter Group (2)

Variable	Is Modifiable	Description
max_connections	O	동시에 접근 가능한 커넥션 개수 설정
innodb_open_files	O	multiple tablespace를 사용하는 경우, .ibd 파일의 최대 숫자 지정
port	X	접속시 사용할 port 설정 (3306)
read_buffer_size	O	Sequential scan(full table scan)을 사용하는 경우, 사용되는 버퍼
read_only	X	데이터를 수정할 수 없게 설정
slow_query_log_file	X	slow_query_log 파일 이름을 설정
socket	X	사용할 소켓 파일 설정
sort_buffer_size	O	filesort 정렬 작업할 때 사용되는 버퍼
default_storage_engine	X	기본적으로 사용할 스토리지 엔진 값을 설정
tmpdir	X	temporary table 생성하는 디렉토리 설정
wait_timeout	O	활동하지 않는 커넥션을 끊을 때까지 서버가 대기하는 시간



7. Parameter Group (3)

Variable	Is Modifiable	Description
lower_case_table_names	0	대소문자 구분을(테이블명, 유저명)
innodb_lock_wait_timeout	0	취소 전 row lock을 위해 대기하는 시간 명시 오라클과 달리 NO WAIT가 없음
general_log	0	제네럴 로그 (connect/disconnect, each SQL statement from clients) 설정
slow_query_log	0	슬로우 쿼리 로그 설정
log_queries_not_using_indexes	0	인덱스를 사용하지 않는 모든 쿼리를 슬로우 쿼리 로그에 저장
log_output	0	로그 저장 형태
long_query_time	0	로그에 기록할 쿼리의 최단 실행시간
performance_schema	0	performance 정보 기록 설정



8. PL/SQL 지원 여부 Server-side scripts (Stored procedure)

Oracle	Aurora
PL/SQL	yes (proprietary syntax)

9. Partitioning Methods

Oracle	Aurora
horizontal partitioning <pre>create table partition_table partition by range(deptno)(partition p1 values less than (20), partition p2 values less than (30), partition p3 values less than (40)) as select * from EMP;</pre>	horizontal partitioning <pre>create table partition_table partition by range(deptno)(partition p1 values less than (20), partition p2 values less than (30), partition p3 values less than (40)) as select * from EMP;</pre>



10. 힌트

분류	힌트명	설명	용법
<ul style="list-style-type: none"> 예약어중 힌트의 용도로 사용 (/*! */) 	USE INDEX	사용할 인덱스를 선택	USE {INDEX KEY} [FOR {JOIN ORDER BY GROUP BY}] ([index_list])
	IGNORE INDEX	명시된 인덱스를 사용하지 않음	IGNORE {INDEX KEY} [FOR {JOIN ORDER BY GROUP BY}] (index_list)
	FORCE INDEX	사용할 인덱스를 강제로 선택	FORCE {INDEX KEY} [FOR {JOIN ORDER BY GROUP BY}] (index_list)
	STRAIGHT_JOIN	조인의 순서를 FROM절의 순서로 고정	SELECT STRAIGHT_JOIN ... FROM
	SQL_SMALL_RESULT	옵티마이저가 결과가 '작을' 것이라고 알려줌	SELECT SQL_SMALL_RESULT ... FROM
	SQL_BIG_RESULT	옵티마이저가 결과가 '클' 것이라고 알려줌	SELECT SQL_BIG_RESULT ... FROM
	SQL_BUFFER_RESULT	결과를 임시테이블에 두며, 서브쿼리에서 사용 불가	SELECT SQL_BUFFER_RESULT ... FROM
	SQL_CACHE	쿼리를 쿼리 캐시(query_cache_type=2)에 캐시함	SELECT SQL_CACHE ... FROM
	SQL_NO_CACHE	쿼리를 쿼리 캐시에 캐시하지 않음	SELECT SQL_NO_CACHE ... FROM
	SQL_CALC_FOUND_ROWS	LIMIT이 붙든 안붙든 모든 로우 수를 계산시킴, FOUND_ROWS()함수로 조회가능.	SELECT SQL_CALC_FOUND_ROWS ... FROM
	HIGH_PRIORITY	테이블 락을 지원하는(ex. Myisam)스토리지 엔진에서 사용, 해당 힌트가 명시된 문장을 락 대기 큐에서의 우선순위를 정함	SELECT HIGH_PRIORITY ... FROM
	LOW_PRIORITY		SELECT LOW_PRIORITY ... FROM
	DELAYED		INSERT DELAYED INTO ...

source: <http://dev.mysql.com/doc/refman/5.7/en/keywords.html>



10. 힌트 - straight_join 힌트 사용법

- 힌트 사용방법 : SQL 일부

```
explain select      straight_join  t1.c1, t1.c2, t2.c2 from exem_i.t1, exem_i.t2 where t1.c1 = t2.c1
```

```
-----
```

id	select_type	table	type	possible_keys	key	key_len	ref	rows	Extra
1	SIMPLE	t1	index	PRIMARY	t1_idx1	34	NULL	5	Using index
1	SIMPLE	t2	ALL	PRIMARY	NULL	NULL	NULL	2	Using where; Using join buffer (flat, BNL join)

```
-----
```

- 힌트 사용방법 : 주석표기 방식

```
explain select /*! straight_join */t1.c1, t1.c2, t2.c2 from exem_i.t1, exem_i.t2 where t1.c1 = t2.c1
```

```
-----
```

id	select_type	table	type	possible_keys	key	key_len	ref	rows	Extra
1	SIMPLE	t1	index	PRIMARY	t1_idx1	34	NULL	5	Using index
1	SIMPLE	t2	ALL	PRIMARY	NULL	NULL	NULL	2	Using where; Using join buffer (flat, BNL join)

```
-----
```



10. 힌트 - Use index 힌트 사용법

- use index 힌트는 해당 인덱스가 존재할 때만 사용한다.(존재하지 않는 인덱스를 사용할 경우, 오류 발생)

```
explain select t1.c1, t1.c2, t2.c2 from exem_i.t1 /*! use index(t1_idx1) */, exem_i.t2 where t1.c1 = t2.c1
```

```
-----
```

id	select_type	table	type	possible_keys	key	key_len	ref	rows	Extra
1	SIMPLE	t2	ALL	PRIMARY	NULL	NULL	NULL	2	
1	SIMPLE	t1	index	NULL	t1_idx1	34	NULL	5	Using where; Using index; Using join buffer (flat, BNL join)

```
-----
```

```
explain select t1.c1, t1.c2, t2.c2 from exem_i.t1 /*! use index(t1_idx2) */, exem_i.t2 where t1.c1 = t2.c1
```

```
-----
```

```
ERROR 1176 (42000) at line 37 in file: '5.hint.sql': Key 't1_idx2' doesn't exist in table 't1'
```



10. 힌트 - Hint 사용법

- ignore index(key | index) 사용 예제

```
mysql> explain select * from employees.employees where emp_no<30000
```

id	select_type	table	partitions	type	possible_keys	key	key_len	ref	rows	filtered	Extra
1	SIMPLE	employees	NULL	range	PRIMARY	PRIMARY	4	NULL	39392	100.00	Using where

```
mysql> explain select * from employees.employees /*! ignore index(primary) */ where emp_no<30000;
```

id	select_type	table	partitions	type	possible_keys	key	key_len	ref	rows	filtered	Extra
1	SIMPLE	employees	NULL	ALL	NULL	NULL	NULL	NULL	299113	33.33	Using where



10. 힌트 - Hint 사용법

- straight join 사용 예제

```
mysql> explain select * from employees,departments limit 1;
```

```
+-----+-----+-----+-----+-----+-----+-----+
| id | select_type | table          | partitions | type | possible_keys | key          |
+-----+-----+-----+-----+-----+-----+-----+
| 1 | SIMPLE      | departments    | NULL       | index | NULL          | dept_name   |
| 1 | SIMPLE      | employees      | NULL       | ALL  | NULL          | NULL        |
+-----+-----+-----+-----+-----+-----+-----+
key_len | ref | rows | filtered | Extra |
+-----+-----+-----+-----+-----+-----+
122     | NULL | 9 | 100.00 | Using index |
NULL    | NULL | 299113 | 100.00 | Using join buffer (Block Nested Loop) |
+-----+-----+-----+-----+-----+-----+
2 rows in set, 1 warning (0.00 sec)
```

```
mysql> explain select /*! STRAIGHT_JOIN */ * from employees,departments limit 1;
```

```
+-----+-----+-----+-----+-----+-----+-----+
| id | select_type | table          | partitions | type | possible_keys | key          |
+-----+-----+-----+-----+-----+-----+-----+
| 1 | SIMPLE      | employees      | NULL       | ALL  | NULL          | NULL        |
| 1 | SIMPLE      | departments    | NULL       | index | NULL          | dept_name   |
+-----+-----+-----+-----+-----+-----+-----+
key_len | ref | rows | filtered | Extra |
+-----+-----+-----+-----+-----+-----+
NULL    | NULL | 299113 | 100.00 | NULL |
122     | NULL | 9 | 100.00 | Using index;
                          Using join buffer (Block Nested Loop) |
+-----+-----+-----+-----+-----+-----+
2 rows in set, 1 warning (0.00 sec)
```




10. 힌트 - Hint 사용법

- SQL_CACHE 사용 예제

```
mysql> select * from employees.employees limit 100000;
mysql> show status like 'Qcache_inserts';
+-----+-----+
| Variable_name | Value |
+-----+-----+
| Qcache_inserts | 3     |
+-----+-----+

mysql> select /*! sql_cache */ * from employees.employees limit 100000;
mysql> show status like 'Qcache_inserts';
+-----+-----+
| Variable_name | Value |
+-----+-----+
| Qcache_inserts | 4     |
+-----+-----+
```



11. Waitevent Interface

Oracle	
EVENT_CLASS	COUNT(*)
Administrative	55
Application	17
Cluster	50
Commit	2
Concurrency	33
Configuration	24
Idle	96
Network	35
Other	958
Queueing	9
Scheduler	8
System I/O	32
User I/O	48

Aurora	
EVENT_CLASS	COUNT
idle	1
wait/io/file	49
wait/io/socket	3
wait/io/table	1
wait/lock/table	1
wait/synch/cond	44
wait/synch/mutex	160
wait/synch/rwlock	33

Oracle
`select wait_class from v$event_name;`

Aurora
`use performance_schema;`
`select * from setup_instruments;`



12. Backup & Recovery

Oracle	Aurora
RMAN	Dashboard RDS 백업 설정



13. Isolation level

	Oracle	Aurora
Level 0 Read Uncommitted	지원 안함	거의 사용하지 않음
Level 1 Read Committed	Oracle이 기본적으로 사용되는 격리 수준	
Level 2 Repeatable Read	지원 안함, for update로 구현 가능	InnoDB가 기본적으로 사용되는 격리 수준
Level 3 Serializable		

Oracle

```
select sid, serial#, flag,  
case when bitand(flag, 268435456) = 0 then 'serializable'  
else 'read committed'  
end as isolation level  
from v$transaction t, v$session s  
where t.addr=s.taddr  
and auid = USERENV('SESSIONID');
```

Aurora

```
select @@global.tx_isolation;
```



14. sqlload, exp·imp

Oracle	Aurora
sqlload	mysqlimport (mysql client tool)
exp·imp	X
pump	X



15. 지원 언어

Oracle		Aurora	
Ada	Java	C	Java
C	JavaScript (Node.js)	C#	JavaScript
C#	Objective-C	C++	Lisp
C++	OCaml	Clojure	Objective C
D	Perl	Cobol	OCaml
Delphi	PHP	Delphi	Perl
Eiffel	Python	Eiffel	PHP
Erlang	Ruby	Erlang	Python
Haskell	Scheme	Fortran	R
	Tcl	Groovy	Ruby
		Haskell	Scala
			Tcl
			Visual Basic

Oracle

```
select parameter, value from v$nls_valid_values where parameter like '%CHARACTERSET%' order by value;
```

Aurora

```
show character set;
```



16. 온라인 리빌드 가능 여부

Oracle	Aurora
<ul style="list-style-type: none">• rebuild index online• ALTER INDEX [schema_name].index_name REBUILD ONLINE [PARAMETERS ('DEFAULT');	<ul style="list-style-type: none">• MySQL 5.6부터 Online DDL을 지원 (in place 인덱스 생성, alter table add nullable column 등)• Oracle과 달리 MySQL은 동일 컬럼에 대한 인덱스를 추가로 생성 가능하므로, 리빌드에 대한 처리도 온라인상으로 가능• ALTER TABLE table_name ADD COLUMN column_name, ROW_FORMAT = COMPACT, ALGORITHM=INPLACE, LOCK=NONE;



17. Aurora 통계정보 관리

show global variables like '%stats%';

```

E1_aurora> show global variables like '%innodb_statS%';
+-----+-----+
| Variable_name | Value |
+-----+-----+
| innodb_stats_auto_recalc | ON |
| innodb_stats_method | nulls_equal |
| innodb_stats_on_metadata | OFF |
| innodb_stats_persistent | ON |
| innodb_stats_persistent_sample_pages | 20 |
| innodb_stats_sample_pages | 8 |
| innodb_stats_transient_sample_pages | 8 |
+-----+-----+
7 rows in set (0.00 sec)

```

항목	값	타입	설명
innodb_stats_auto_recalc	ON	boolean	<ul style="list-style-type: none"> 테이블의 로우가 10%이상 변경되면 recalculate함 innodb_stats_persistent 가 on 되어있어야 함 stats_auto_recalc로 테이블 단위 설정 가능
innodb_stats_method	nulls_equal	nulls_equal nulls_unequal nulls_ignored	<ul style="list-style-type: none"> InnoDB Table의 인덱스 값의 Null 값 처리방법 (동일처리, 불일치, 무시)
innodb_stats_on_metadata	OFF	boolean	<ul style="list-style-type: none"> SHOW TABLE STATUS 명령어 사용 INFORMATION_SCHEMA.TABLES 테이블과 INFORMATION_SCHEMA.STATISTICS 테이블에 액세스 시 통계정보 갱신
innodb_stats_persistent	ON	boolean	<ul style="list-style-type: none"> 통계정보를 영구적으로 디스크에 저장할지 여부, 테이블 생성시 stats_persistent 옵션으로 테이블 별 설정 가능
innodb_stats_persistent_sample_pages	20	integer	<ul style="list-style-type: none"> 통계정보를 디스크에 저장할 때 샘플링 페이지 수
innodb_stats_sample_pages	8	integer	<ul style="list-style-type: none"> 5.6 버전부터 사용되지 않음 (innodb_stats_transient_sample_page 사용)
innodb_stats_transient_sample_pages	8	integer	<ul style="list-style-type: none"> 통계정보를 자동으로 수집할 때 샘플링 페이지 수



17. Aurora 통계정보 관리

```
select * from mysql.innodb_table_stats;
```

```
mysql> select * from mysql.innodb_table_stats;
```

database_name	table_name	last_update	n_rows	clustered_index_size	sum_of_other_index_sizes
db1	ACT	2017-09-12 07:42:55	0	1	0
db1	CL_SCHED	2017-09-12 07:34:46	0	1	0
db1	DEPARTMENT	2017-09-12 07:36:18	0	1	0
db1	EMPLOYEE	2017-09-12 07:38:31	0	1	0
db1	EMPPROJACT	2017-09-12 07:42:48	0	1	0
db1	EMP_PHOTO	2017-09-12 07:39:50	0	1	0
db1	IN_TRAY	2017-09-12 07:43:11	0	1	0
db1	PROJACT	2017-09-12 07:42:38	0	1	0
db1	PROJECT	2017-09-12 07:42:03	0	1	0
db1	t1	2017-09-12 08:14:29	0	1	0
example_db	dept	2017-09-20 09:02:23	0	1	0
example_db	employee	2017-09-20 09:06:30	3	1	0
example_db	posts	2017-09-20 09:02:42	0	1	0
mysql	aurora_s3_load_history	2017-09-11 04:47:14	0	1	0
mysql	columns_priv	2014-08-01 21:43:55	0	1	0
mysql	db	2014-08-01 21:43:54	0	1	1
mysql	event	2014-08-01 21:43:56	2	1	0
mysql	func	2014-08-01 21:43:55	0	1	0
mysql	help_category	2014-08-01 21:43:56	40	1	1
mysql	help_keyword	2014-08-01 21:43:55	0	1	1
mysql	help_relation	2014-08-01 21:43:55	0	1	0
mysql	help_topic	2014-08-01 21:43:58	582	97	5
mysql	ndb_binlog_index	2014-08-01 21:43:55	0	1	0
mysql	plugin	2014-08-01 21:43:55	0	1	0



17. Aurora 통계정보 관리

```
select * from mysql.innodb_table_stats;
```

```

| mysql      | plugin      | 2014-08-01 21:43:55 | 0 | 1 | 0 |
| mysql      | proc        | 2017-09-11 04:47:14 | 21 | 7 | 0 |
| mysql      | procs_priv  | 2014-08-01 21:43:55 | 0 | 1 | 1 |
| mysql      | proxies_priv | 2015-02-11 21:06:31 | 0 | 1 | 1 |
| mysql      | rds_configuration | 2014-08-01 21:43:55 | 0 | 1 | 0 |
| mysql      | rds_global_status_history | 2014-08-01 21:43:55 | 0 | 1 | 0 |
| mysql      | rds_global_status_history_old | 2014-08-01 21:43:55 | 0 | 1 | 0 |
| mysql      | rds_heartbeat2 | 2014-08-01 21:43:55 | 0 | 1 | 0 |
| mysql      | rds_history | 2014-08-01 21:43:55 | 0 | 1 | 0 |
| mysql      | rds_replication_status | 2014-08-01 21:43:55 | 0 | 1 | 0 |
| mysql      | rds_sysinfo | 2014-08-01 21:43:55 | 0 | 1 | 0 |
| mysql      | servers     | 2014-08-01 21:43:55 | 0 | 1 | 0 |
| mysql      | tables_priv | 2014-08-01 21:43:55 | 0 | 1 | 1 |
| mysql      | user        | 2017-09-11 04:47:14 | 0 | 1 | 0 |
+-----+-----+-----+-----+-----+-----+

```

database_name	데이터베이스 이름	n_rows	로우 수
table_name	테이블이나 파티션 이름	clustered_index_size	프라이머리 인덱스 사이즈(페이지)
last_update	마지막으로 갱신된 시간	sum_of_other_index_sizes	PK인덱스를 제외한 모든 인덱스의 사이즈 합계



17. Aurora 통계정보 관리

```
select * from mysql.innodb_index_stats where table_name = 'employees';
```

```
mysql> select * from mysql.innodb_index_stats where table_name = 'employees';
```

```

+-----+-----+-----+-----+-----+-----+-----+-----+
|database_name|table_name|index_name|last_update          | stat_name  |stat_value |sample_size| stat_description      |
+-----+-----+-----+-----+-----+-----+-----+-----+
|employees   |employees|PRIMARY  |2016-06-22 14:46:23 | n_diff_pfx01| 299468 | 20| emp_no                |
|employees   |employees|PRIMARY  |2016-06-22 14:46:23 | n_leaf_pages| 886 | NULL| Number of leaf pages in the index |
|employees   |employees|PRIMARY  |2016-06-22 14:46:23 | size       | 929 | NULL| Number of pages in the index      |
+-----+-----+-----+-----+-----+-----+-----+-----+
3 rows in set (0.06 sec)

```

database_name	데이터베이스 이름	stat_name	통계정보의 이름
table_name	테이블이나 파티션 이름	stat_value	해당 통계정보의 수치
index_name	인덱스 이름	sample_size	stat_value를 측정할때의 샘플링한 사이즈
last_update	마지막으로 갱신된 시간	stat_description	해당 통계정보의 설명



18. Storage engine

- MyISAM The default non-transactional storage engine that provides full-text indexing and is highly portable
- Memory (HEAP) A non-transactional storage engine that stores data in memory
- BDB (Berkeley DB) The first transactional-safe storage engine
- InnoDB A transactional-safe storage engine designed for maximum performance when processing large volume of data and that provides row-level locking

Comparison for Common Storage Engines

Feature	MyISAM	Heap (Memory)	BDB	InnoDB
Transactional	No	No	Yes	Yes
Lock granularity	Table	Table	Page	Row
Storage	A data file (.MYD) and an index file (.MYI) for each table	In-memory	A single data and index file (.db) for each table	A set of data files for all the tables
Portable	Yes	N/A	No	Yes

source: http://docs.oracle.com/cd/E12151_01/doc.150/e12155/oracle_mysql_compared.htm#i1026408



18. Storage engine

Aurora의 storage engine은 **InnoDB**만 사용

```
mysql> show global variables like '%engine%';
+-----+-----+
| Variable_name | Value |
+-----+-----+
| default_storage_engine | InnoDB |
| default_tmp_storage_engine | InnoDB |
| ignore_default_storage_engine_errors | OFF |
| storage_engine | InnoDB |
+-----+-----+
4 rows in set (0.01 sec)
```



18. Storage engine

Aurora

create table .. engine = MyISAM으로 적고, show table status 하면 엔진이 InnoDB로 보임

```
E1> create table t6 ( c1 int, c2 varchar(120) ) engine=myisam ;  
Query OK, 0 rows affected, 1 warning (0.03 sec)
```

```
E1> show table status ;
```

Name	Engine	Version	Row_format	Rows	Avg_row_length	Data_length	Max_data_length	Index_length	Data_free
t1	InnoDB	10	Compact	0	0	16384	0	0	0
t2	InnoDB	10	Compact	0	0	16384	0	0	0
t3	InnoDB	10	Compact	3	5461	16384	0	0	0
t4	InnoDB	10	Compact	0	0	16384	0	0	0
t5	InnoDB	10	Compact	4	4096	16384	0	0	0
t6	InnoDB	10	Compact	0	0	16384	0	0	0

Auto_increment	Create_time	Update_time	Check_time	Collation	Checksum	Create_options	Comment
NULL	NULL	NULL	NULL	latin1_swedish_ci	NULL	row_format=COMPACT	
NULL	NULL	NULL	NULL	latin1_swedish_ci	NULL		
NULL	NULL	NULL	NULL	utf8_general_ci	NULL		
NULL	NULL	NULL	NULL	utf8_general_ci	NULL		
NULL	NULL	NULL	NULL	utf8_general_ci	NULL		
NULL	NULL	NULL	NULL	latin1_swedish_ci	NULL		

```
6 rows in set (0.00 sec)
```



19. Character set

```
mysql> show character set ;
```

Charset	Description	Default collation	Maxlen
big5	Big5 Traditional Chinese	big5_chinese_ci	2
dec8	DEC West European	dec8_swedish_ci	1
cp850	DOS West European	cp850_general_ci	1
hp8	HP West European	hp8_english_ci	1
koi8r	KOI8-R Relcom Russian	koi8r_general_ci	1
latin1	cp1252 West European	latin1_swedish_ci	1
latin2	ISO 8859-2 Central European	latin2_general_ci	1
swe7	7bit Swedish	swe7_swedish_ci	1
ascii	US ASCII	ascii_general_ci	1
ujis	EUC-JP Japanese	ujis_japanese_ci	3
sjis	Shift-JIS Japanese	sjis_japanese_ci	2
hebrew	ISO 8859-8 Hebrew	hebrew_general_ci	1
tis620	TIS620 Thai	tis620_thai_ci	1
euckr	EUC-KR Korean	euckr_korean_ci	2
koi8u	KOI8-U Ukrainian	koi8u_general_ci	1
gb2312	GB2312 Simplified Chinese	gb2312_chinese_ci	2
greek	ISO 8859-7 Greek	greek_general_ci	1
cp1250	Windows Central European	cp1250_general_ci	1
gbk	GBK Simplified Chinese	gbk_chinese_ci	2
latin5	ISO 8859-9 Turkish	latin5_turkish_ci	1
armscii8	ARMSCII-8 Armenian	armscii8_general_ci	1
utf8	UTF-8 Unicode	utf8_general_ci	3
ucs2	UCS-2 Unicode	ucs2_general_ci	2
cp866	DOS Russian	cp866_general_ci	1
keybcs2	DOS Kamenicky Czech-Slovak	keybcs2_general_ci	1
macce	Mac Central European	macce_general_ci	1
macroman	Mac West European	macroman_general_ci	1
cp852	DOS Central European	cp852_general_ci	1
latin7	ISO 8859-13 Baltic	latin7_general_ci	1
utf8mb4	UTF-8 Unicode	utf8mb4_general_ci	4
cp1251	Windows Cyrillic	cp1251_general_ci	1
utf16	UTF-16 Unicode	utf16_general_ci	4
utf16le	UTF-16LE Unicode	utf16le_general_ci	4
cp1256	Windows Arabic	cp1256_general_ci	1

- Aurora 1.14 (40개)
- MariaDB 10.1 (40개)
- MySQL 5.7 (41개)

Oracle

`select parameter, value from V$NLS_PARAMETERS;`

Aurora

`Ws`

`status;`

`show variables like 'char%';`



InnoDB 특징

1. 역사
2. 아키텍처
3. 메모리
4. 디스크
5. 성능





InnoDB 특징

1. 역사
2. 아키텍처
3. 메모리
4. 디스크
5. 성능



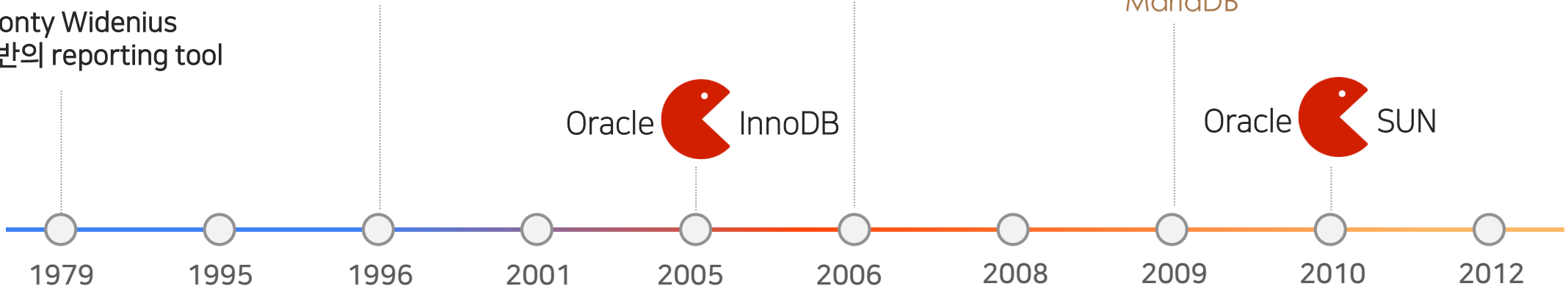


MySQL 및 InnoDB의 역사



SQL 필요
"MSQL(David Hughes) + ISAM(Unireg)"
-> MySQL Ver. 1

TcX의 Monty Widenius
ISAM 기반의 reporting tool
"Unireg"



Heikki Tuuri "InnoDB"



MySQL Ver. 4에
InnoDB 포함

Oracle InnoDB



Monty Program AB
"MariaDB"



SkySQL ->
Maria DB Foundation
MariaDB 10.0





InnoDB 특징

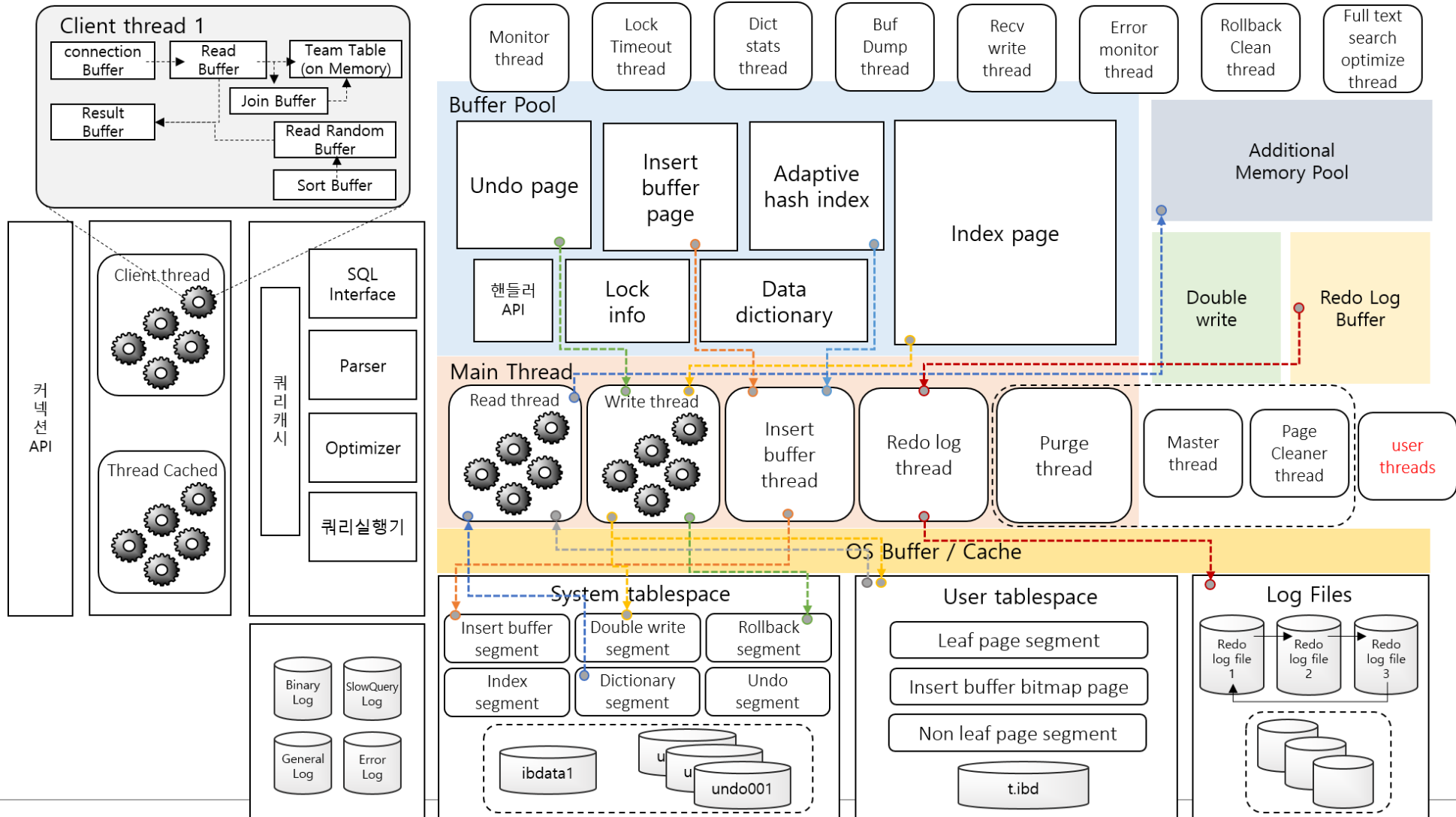
1. 역사
2. 아키텍처
3. 메모리
4. 디스크
5. 성능





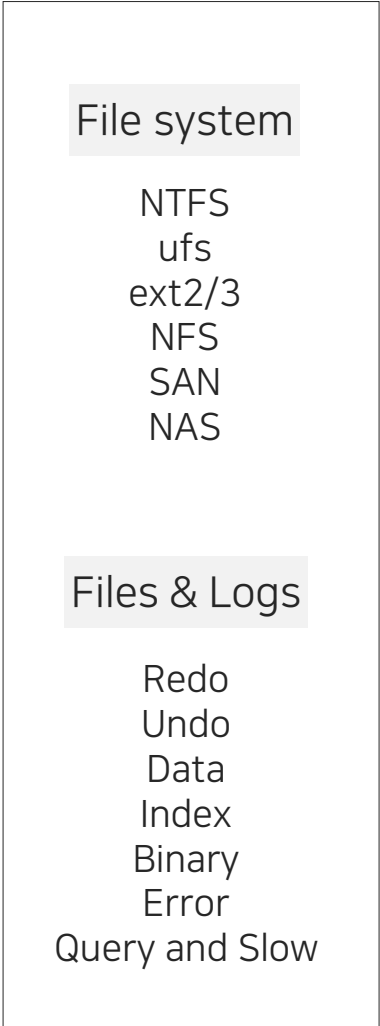
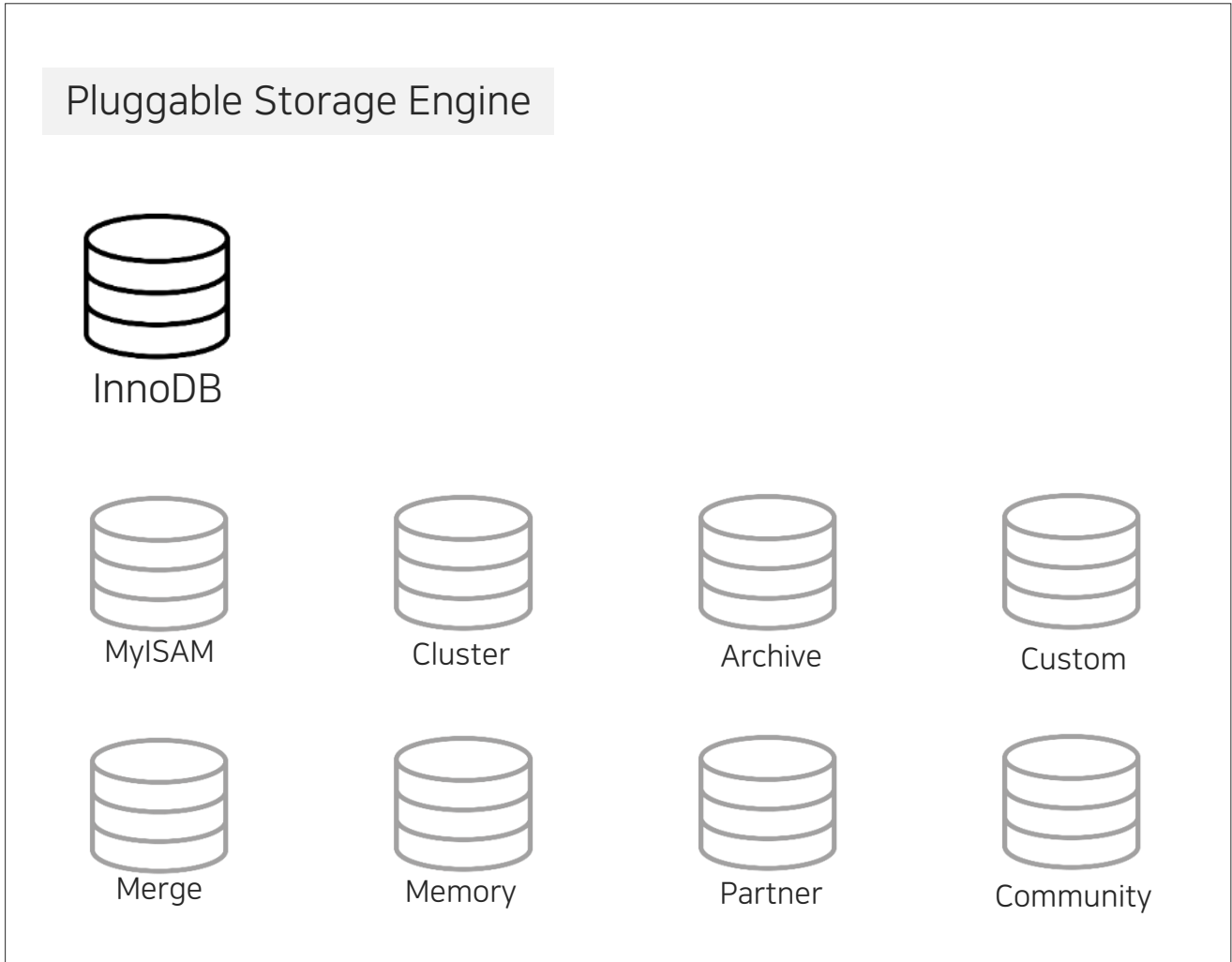
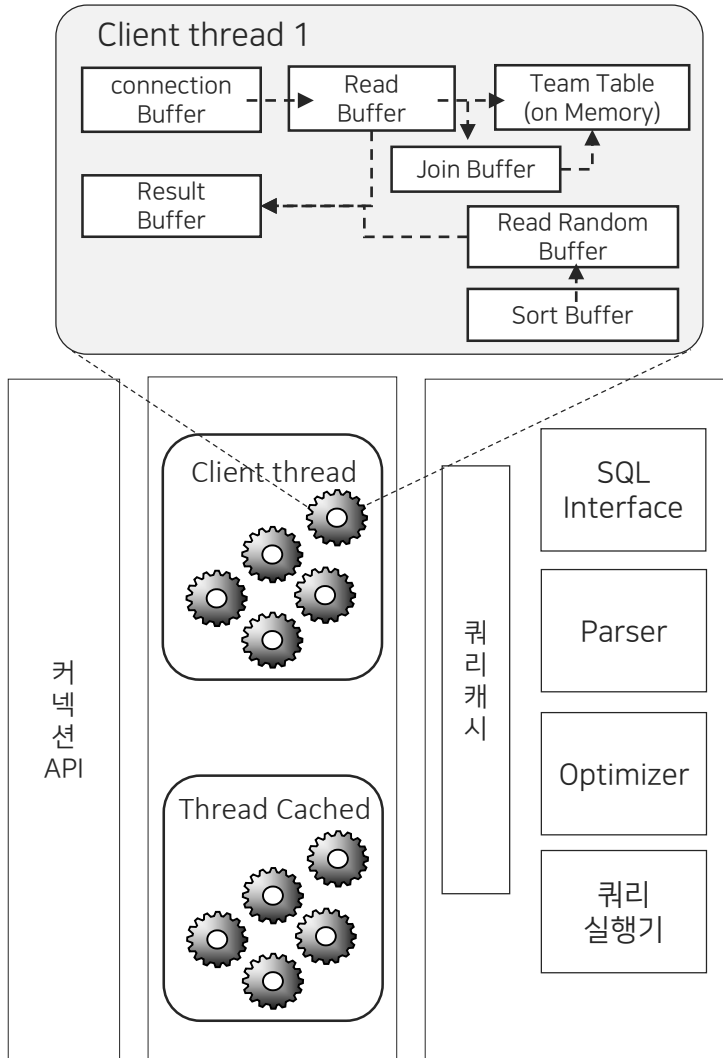
InnoDB Architecture

InnoDB 엔진을 Aurora Cluster Volume 위에 customizing하여 구성
Aurora의 File System은 EBS를 사용하지 않으며, Quorum system을 차용한 Cluster Shared Volume



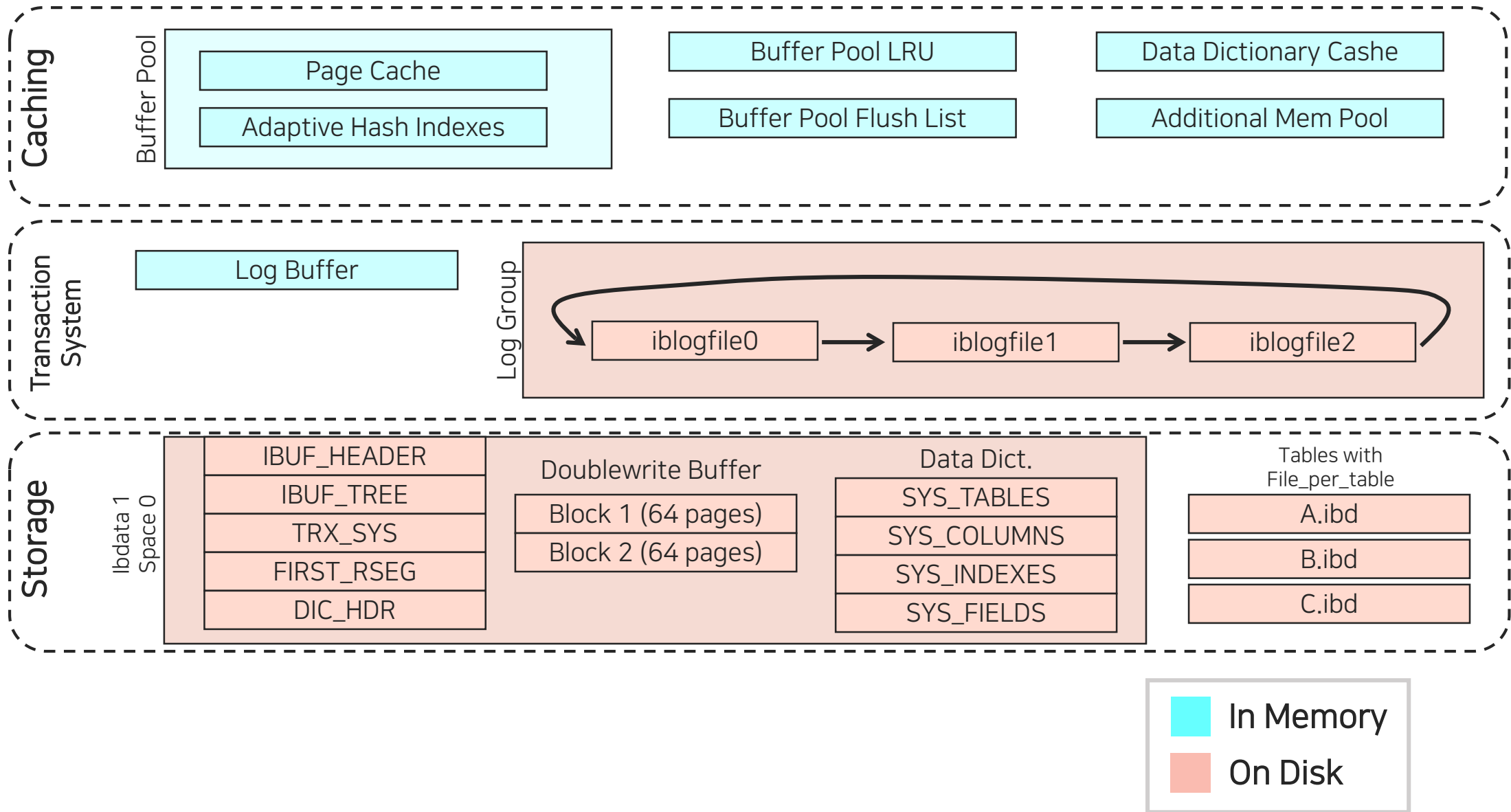


InnoDB Architecture





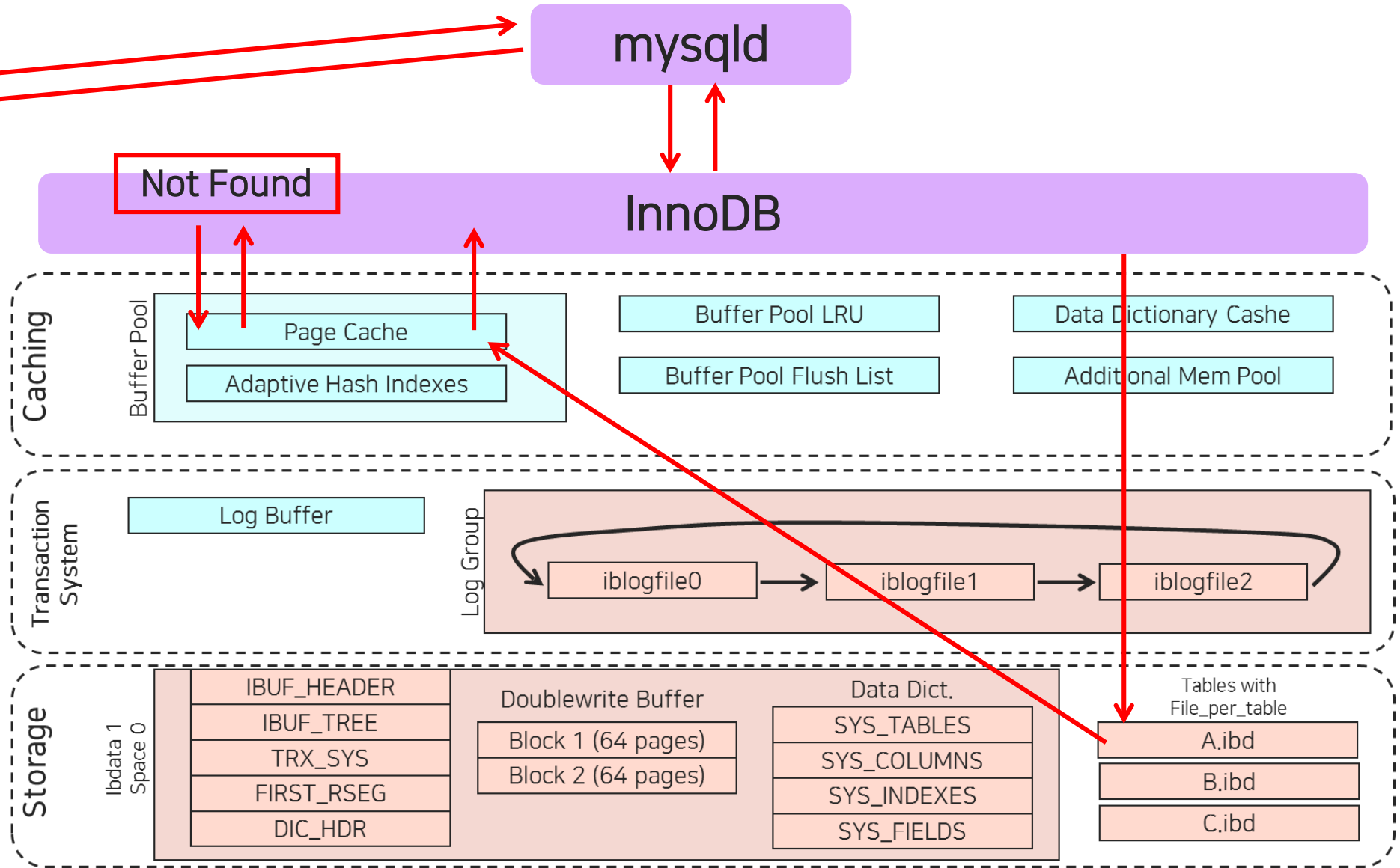
InnoDB High Level Overview





Query (pages in buffer pool)

SELECT * FROM a
WHERE id = 10;



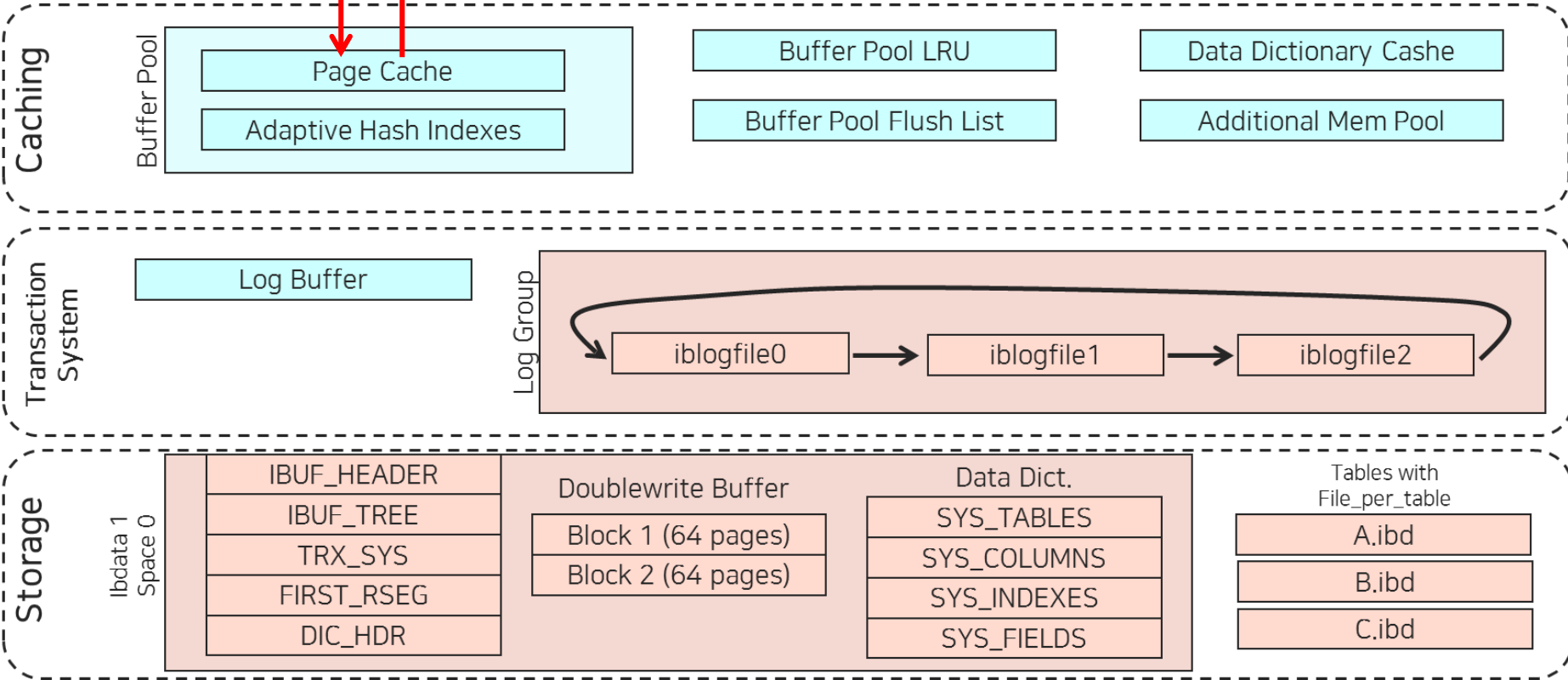


Query (pages not in buffer pool)

SELECT * FROM a
WHERE id = 10;

mysqld

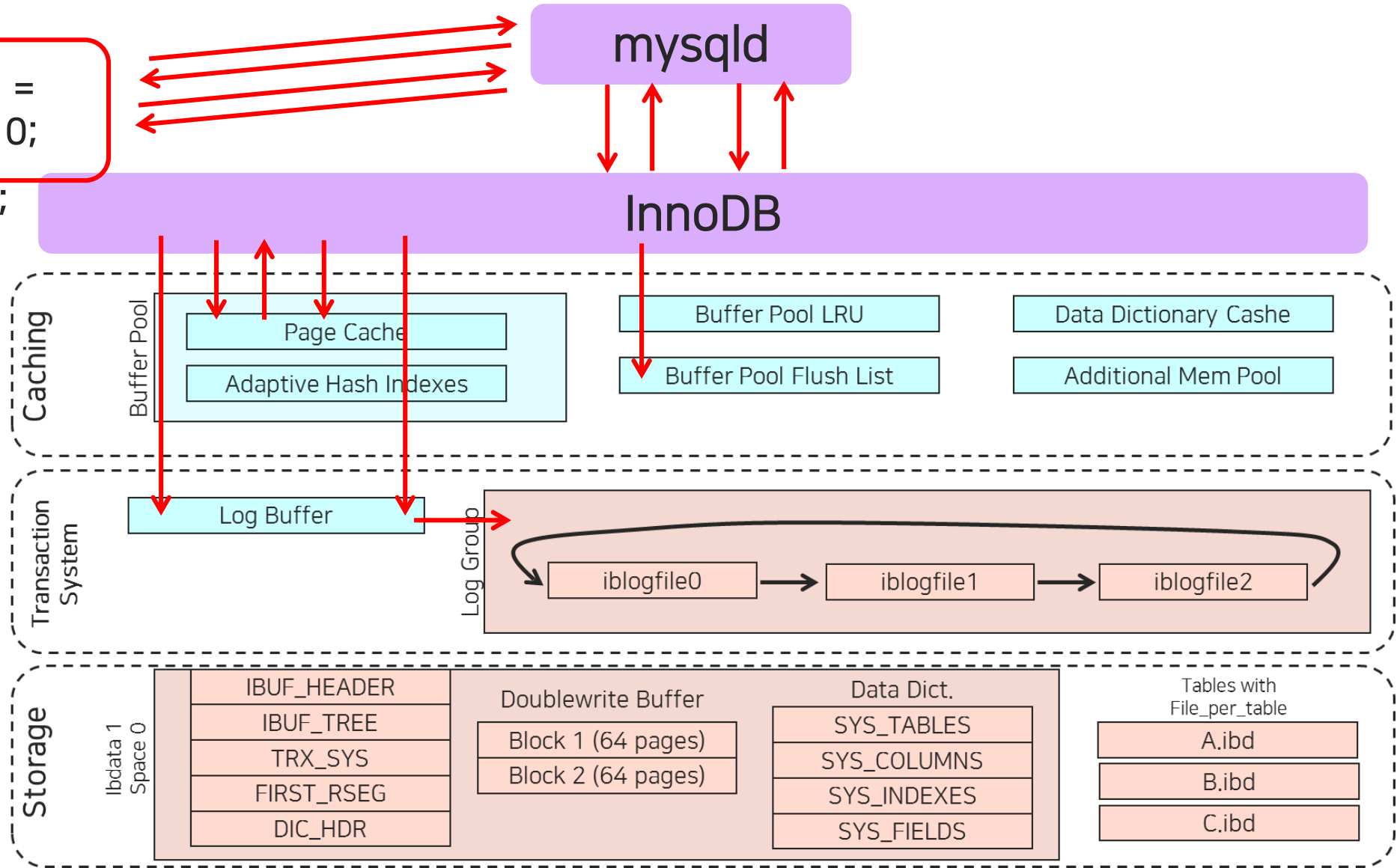
InnoDB



 Update Query in a Transaction (simplified)

```
UPDATE a SET col1 = 'new' WHERE id = 10;
```

Commit;

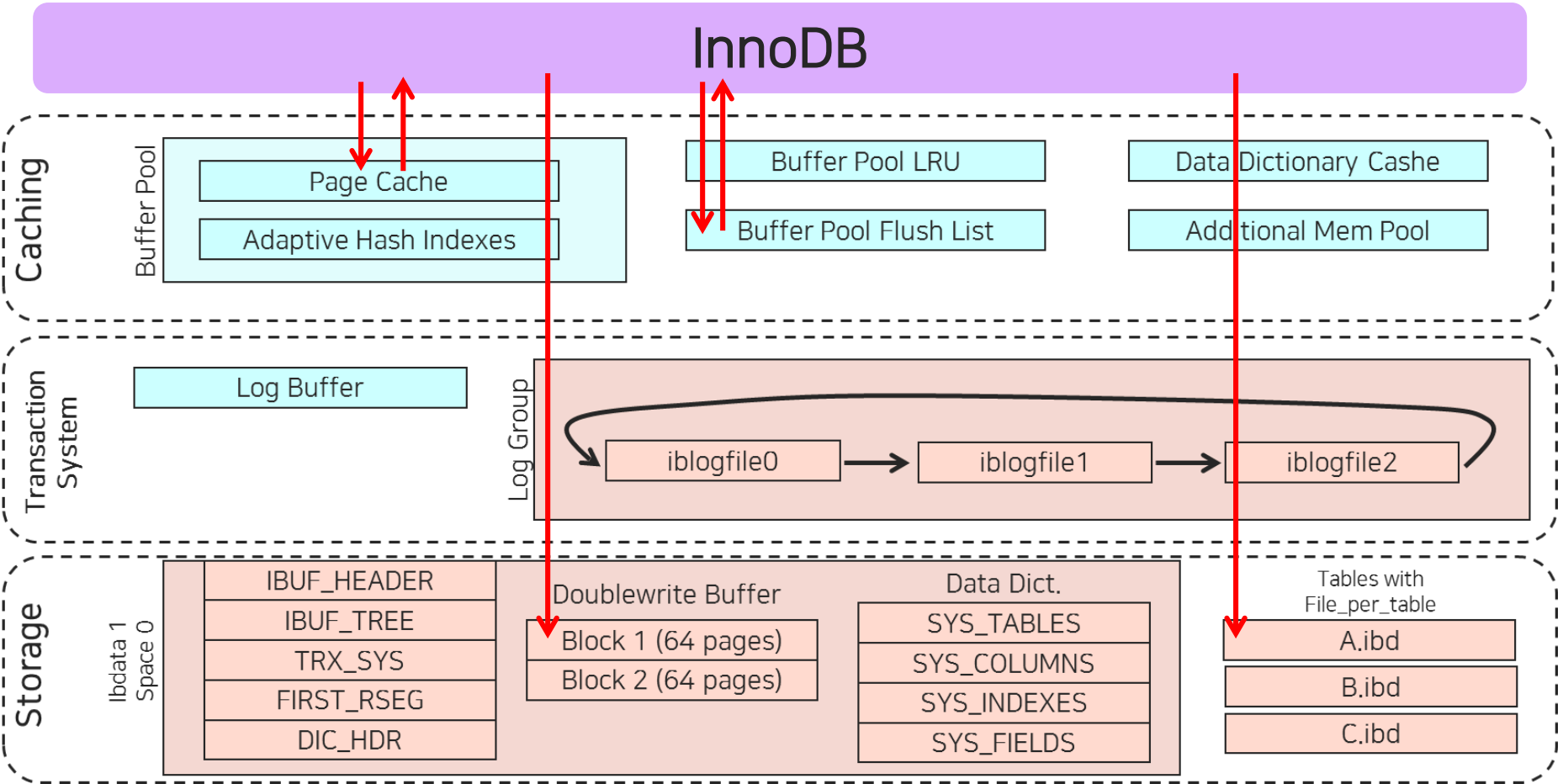




Checkpoint (Background Activity)

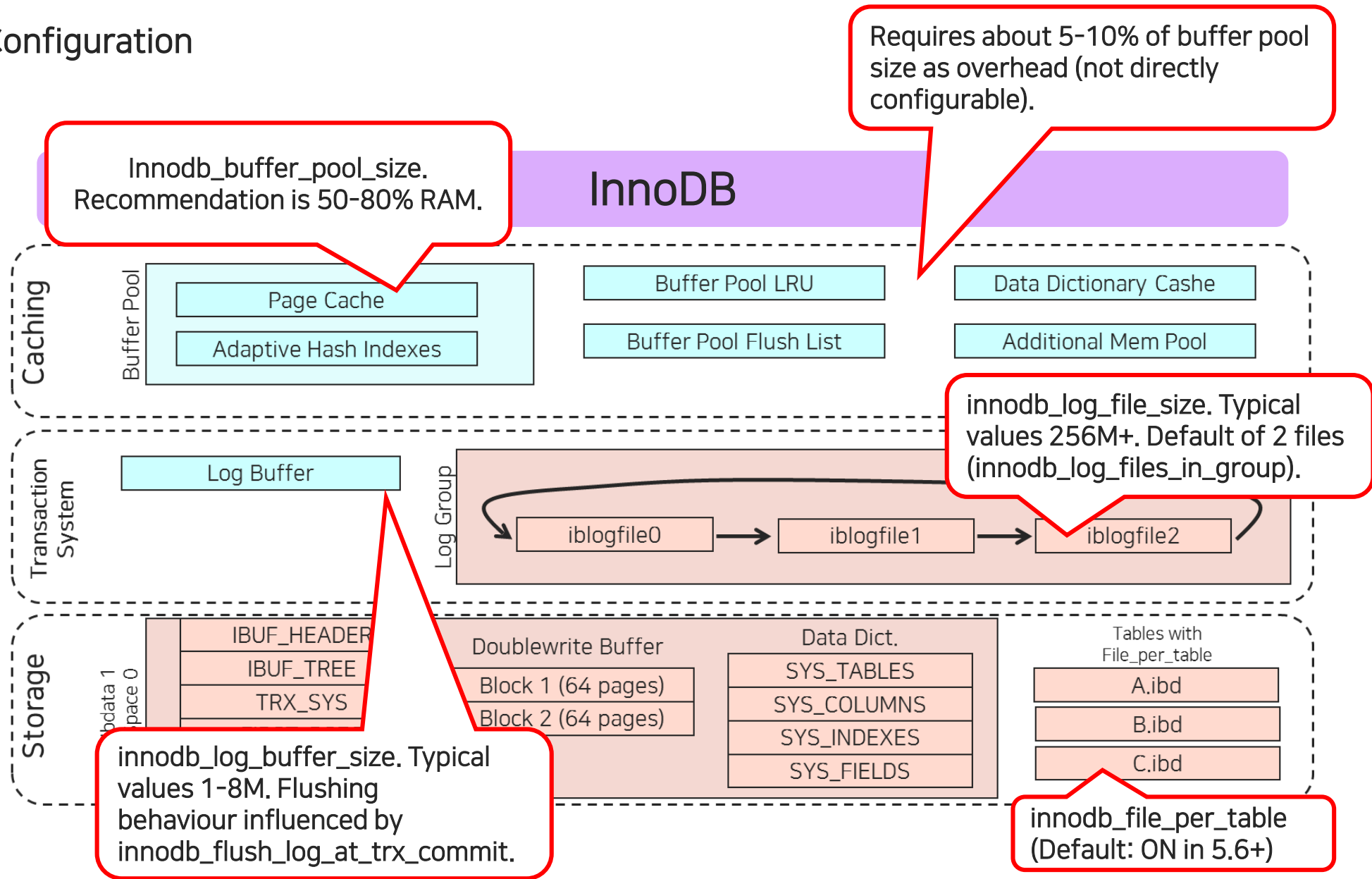
(nothing)

mysqld





Basic Configuration





InnoDB 특징

- 1. 역사
- 2. 아키텍처
- 3. 메모리
 - 3.1 메모리 아키텍처
 - 3.2 Buffer Pool
 - 3.3 Insert Buffer
 - 3.4 Redo Log Buffer
- 4. 디스크
- 5. 성능





InnoDB 특징

- 1. 역사
- 2. 아키텍처
- 3. 메모리
 - 3.1 메모리 아키텍처
 - 3.2 Buffer Pool
 - 3.3 Insert Buffer
 - 3.4 Redo Log Buffer
- 4. 디스크
- 5. 성능

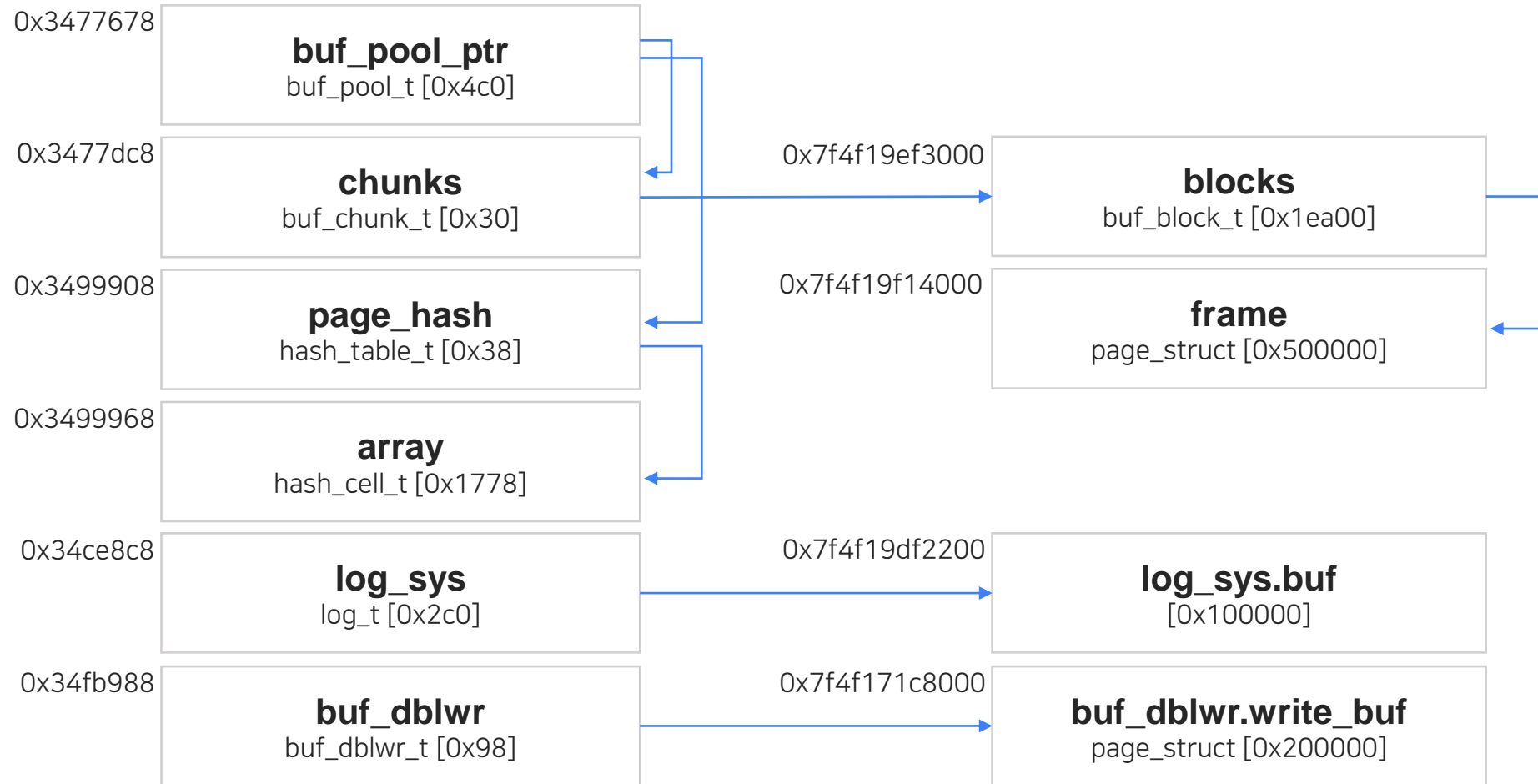




Memory Architecture

innodb_buffer_pool_size = 5MB

srv_page_size = 0x4000



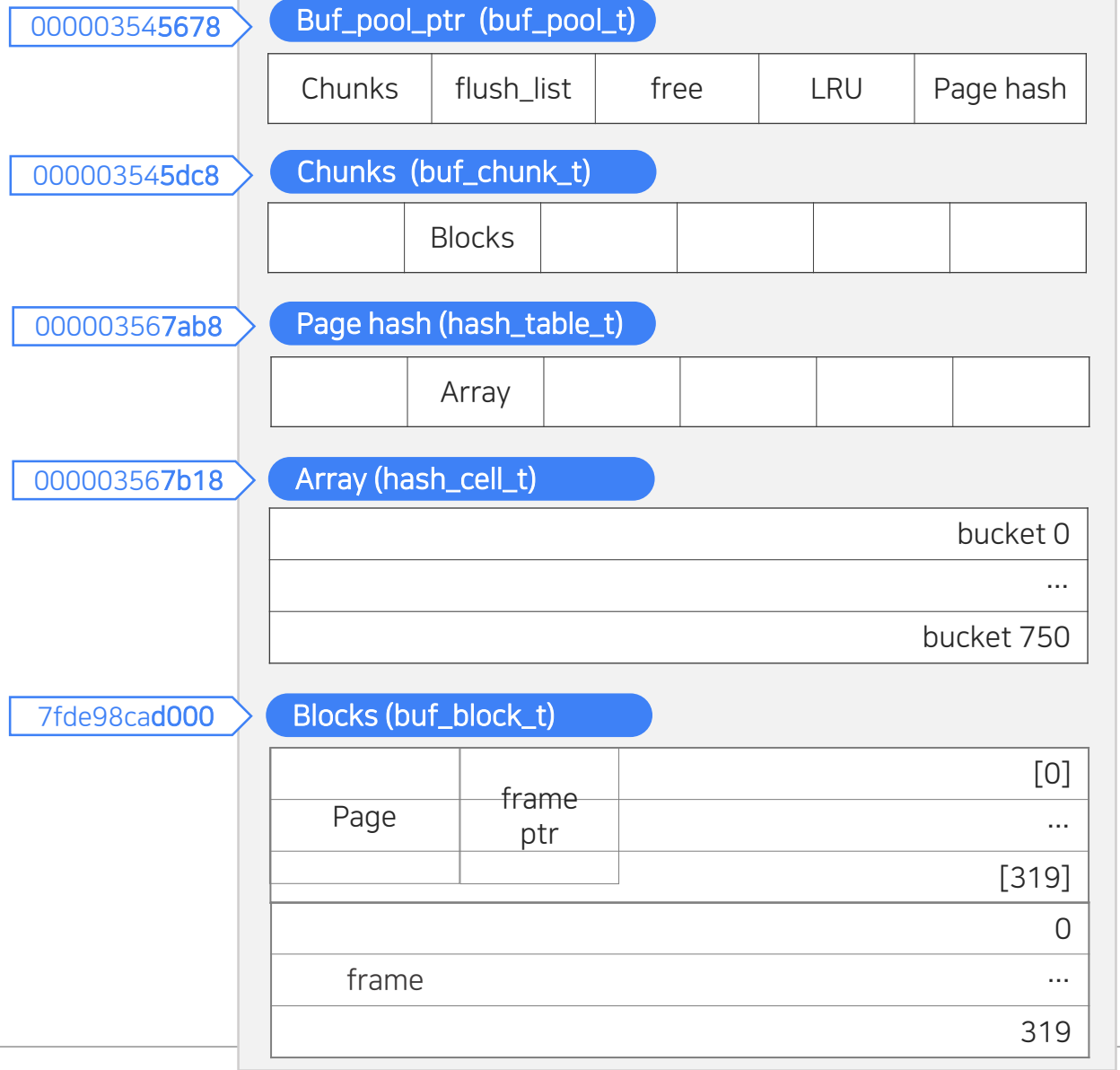


Buffer Pool Routines

1. LRU replacement algorithm
2. Flush algorithm
3. Buffer read
 - 해시 테이블 구조를 이용하여 읽고자하는 페이지가 버퍼 풀에 존재하는지 존재여부를 확인
4. Binary buddy allocator for compressed pages
 - 페이지 압축과 관련된 리스트 관리



Buffer Pool - 전반적 구조 (물리적 순서)





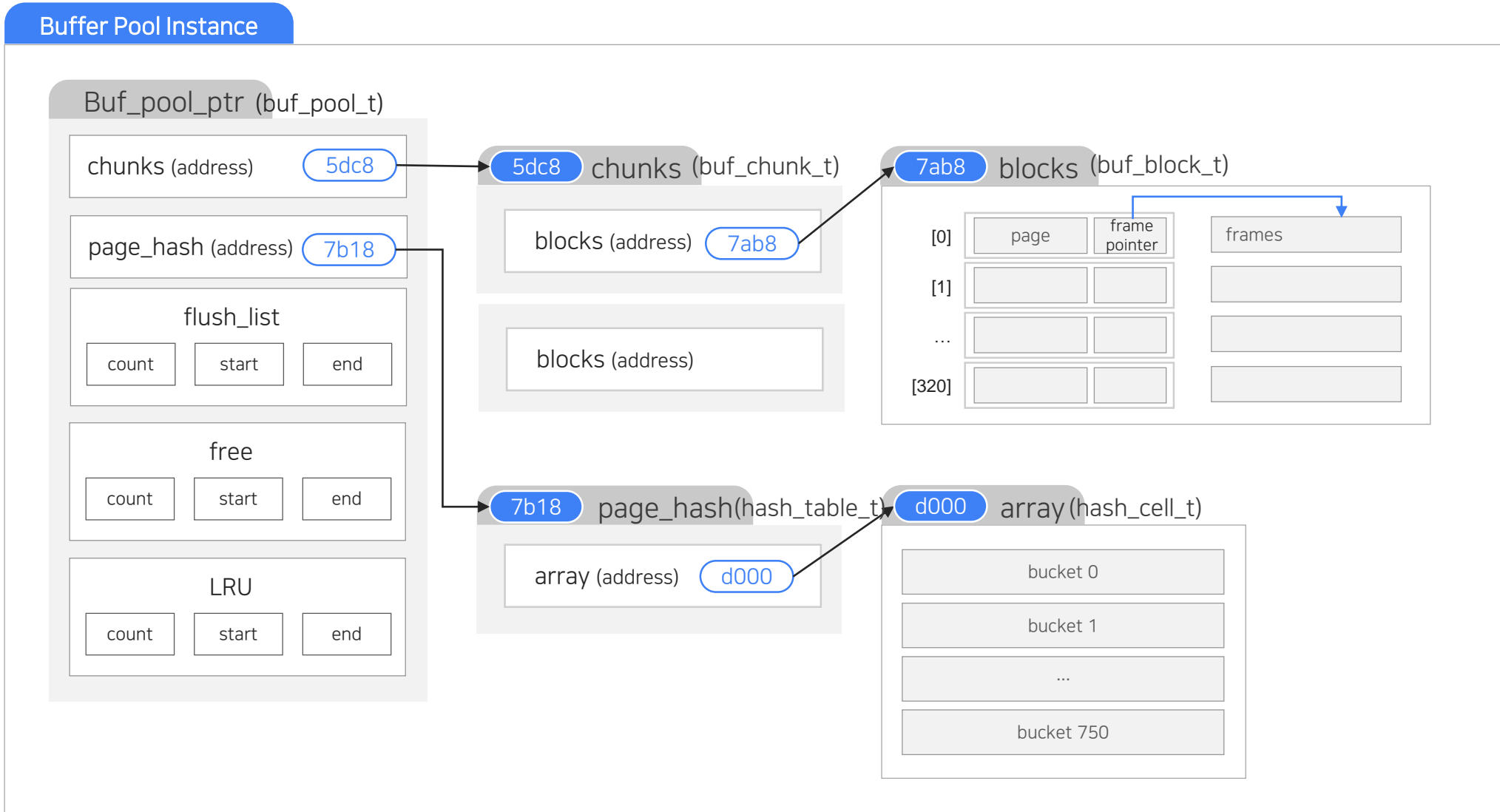
InnoDB 특징

- 1. 역사
- 2. 아키텍처
- 3. 메모리
 - 3.1 메모리 아키텍처
 - 3.2 Buffer Pool
 - 3.3 Insert Buffer
 - 3.4 Redo Log Buffer
- 4. 디스크
- 5. 성능





Buffer Pool - 전반적 구조





Buffer Pool Instance

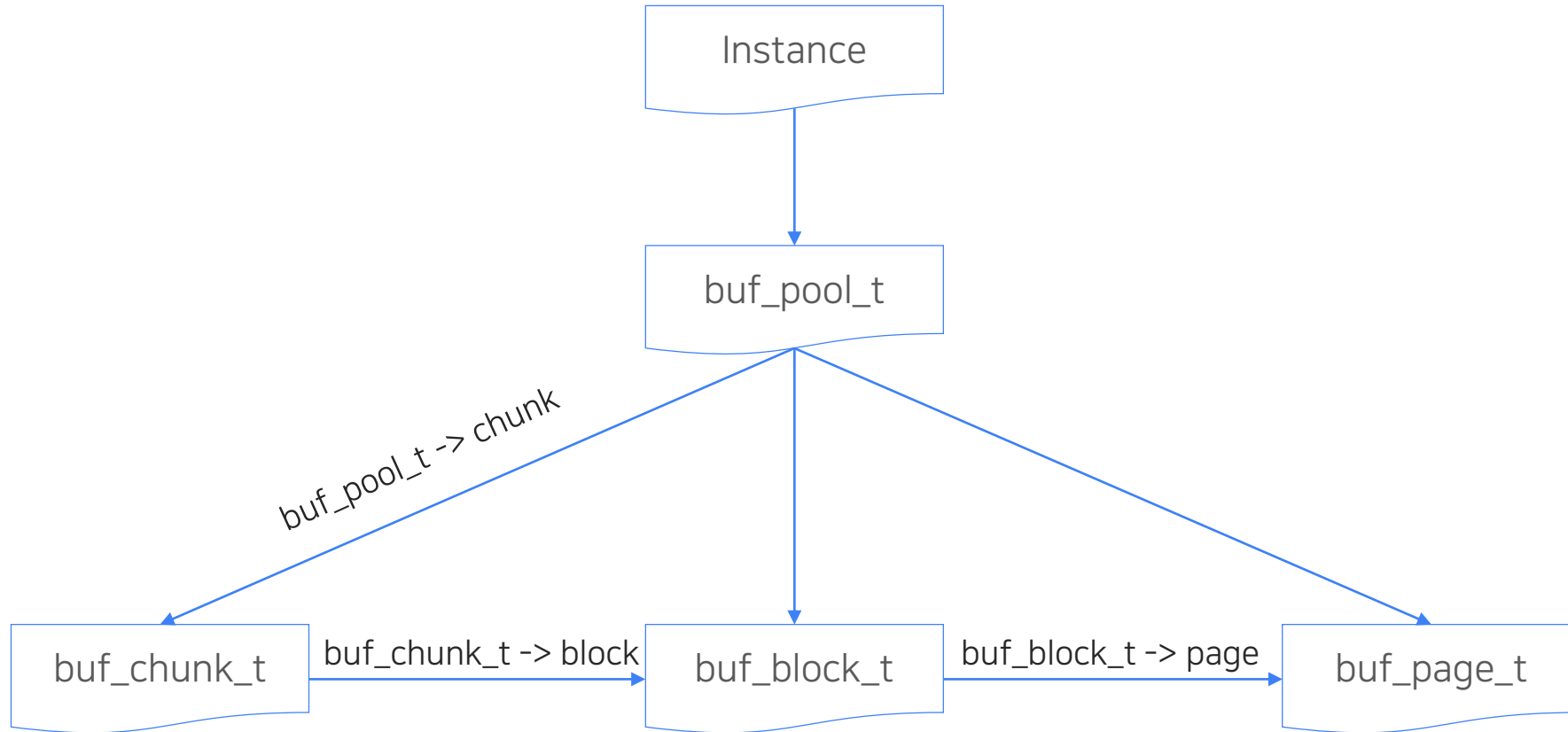
```
mysql> show global variables like 'innodb_buffer%' ;
```

Variable_name	Value
innodb_buffer_pool_chunk_size	134217728
innodb_buffer_pool_dump_at_shutdown	OFF
innodb_buffer_pool_dump_now	OFF
innodb_buffer_pool_dump_pct	25
innodb_buffer_pool_filename	ib_buffer_pool
innodb_buffer_pool_instances	8
innodb_buffer_pool_load_abort	OFF
innodb_buffer_pool_load_at_startup	OFF
innodb_buffer_pool_load_now	OFF
innodb_buffer_pool_size	1073741824
innodb_change_buffer_max_size	25
innodb_change_buffering	all
innodb_log_buffer_size	16777216
innodb_sort_buffer_size	1048576

- innodb_buffer_pool_size 1GB 이하
innodb_buffer_pool_instances = 1
- innodb_buffer_pool_size 1GB 이상
innodb_buffer_pool_instances = 8



Buffer Pool - 주요 구조체





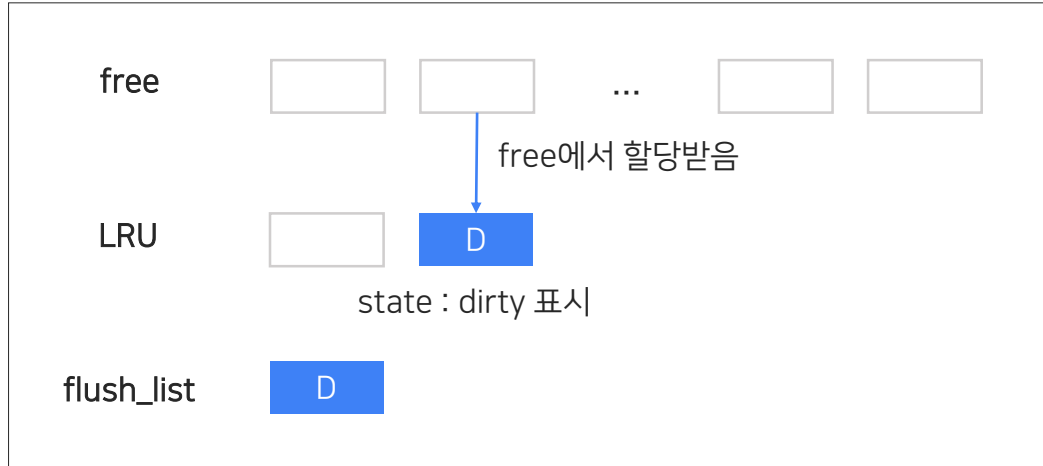
Buffer Pool - 주요 구조체

구조체명	설명
buf_pool_t	<ul style="list-style-type: none">• buffer pool instance 레벨의 제어• buffer pool instance의 mutex, instance_no, page_hash, old_list_pointer 등
buf_block_t	<ul style="list-style-type: none">• 데이터 페이지 구조체• 대부분의 정보는 buf_page_t 에 포함• 첫 번째 필드인 buf_page_t는 임의로 첫 번째가 아닌 필수적• 두 번째 필드인 frame은 실제 데이터가 있는 페이지를 포인트• Unzip LRU List 링크드 리스트 루트노드 저장
buf_page_t	<ul style="list-style-type: none">• 대부분의 데이터 페이지 정보 저장• space_id, page_no, page state, newest_modification, oldest_modifiction, access_time 및 zip page의 모든 정보 등이 저장• state 필드는 현재 페이지의 상태를 나타냄 (8 가지)



Buffer Pool

1) dirty 페이지 생성된 경우



List Name	Description
Free List	• 프리 페이지만 포함
LRU List	• 파일 페이지를 소유하는 모든 블록을 포함
Flush List	• 메모리에서 수정된 파일 페이지를 소유한 블록들을 포함 (단, 디스크에 내려쓰지 않음) • 오라클의 "LRUW"와 동일한 개념

2) Write Thread가 메모리에서 디스크로 내려쓴 경우





buf_pool_ptr

./storage/innobase/include/buf0buf.h

```

struct buf_pool_t{
    buf_chunk_t*   chunks;   /*!< buffer pool chunks */
    hash_table_t* page_hash; /*!< hash table of buf_page_t or buf_block_t file pages,
                             buf_page_in_file() == TRUE, indexed by (space_id, offset).
                             page_hash is protected by an array of mutexes.
                             Changes in page_hash are protected by buf_pool ->
                             mutex and the relevant page_hash mutex. Lookups can happen
                             while holding the buf_pool ->mutex or the relevant page_hash mutex.*/

    UT_LIST_BASE_NODE_T(buf_page_t) flush_list; /*!< base node of the modified block list */

    UT_LIST_BASE_NODE_T(buf_page_t) free; /*!< base node of the free block list */

    UT_LIST_BASE_NODE_T(buf_page_t) LRU; /*!< base node of the LRU list */
}

```

Member	Description
chunks	buffer pool chunks
page_hash	buf_page_t와 buf_block_t file pages의 hash table
flush_list	수정된 블록 리스트
free	프리 블록 리스트
LRU	LRU 리스트



buf_pool_ptr

buf_pool_ptr (buf_pool_t)

000003545708	01 00 00 00 00 00 00 00	c8 5d 54 03 00 00 00 00	>.....]T.....<
	...		
000003545738	b8 7a 56 03 00 00 00 00	00 00 00 00 00 00 00 00	>.zV.....<
	...		
000003545908	00 00 00 00 00 00 00 00	00 00 00 00 00 00 00 00	>.....<
000003545918	00 00 00 00 00 00 00 00	00 00 00 00 00 00 00 00	>.....<
	...		
000003545988	01 00 00 00 00 00 00 00	e8 00 00 00 00 00 00 00	>.....<
000003545998	c0 56 cb 98 de 7f 00 00	78 b8 cc 98 de 7f 00 00	>.V.....x.....<
	...		
000003545a18	00 00 00 00 00 00 00 00	58 00 00 00 00 00 00 00	>.....X.....<
000003545a28	38 55 cb 98 de 7f 00 00	00 d0 ca 98 de 7f 00 00	>8U.....<

0x3545708	chunks (8 Bytes)		
0x3545738	page_hash (8 Bytes)		
flush_list			
0x3545908-18	count (8 Bytes)	start (8 Bytes)	end (8 Bytes)
free			
0x3545988-98	count (8 Bytes)	start (8 Bytes)	end (8 Bytes)
LRU			
0x3545a18-28	count (8 Bytes)	start (8 Bytes)	end (8 Bytes)



buf_chunk_t

./storage/innobase/include/buf0buf.h

```
/** A chunk of buffers. The buffer pool is allocated in chunks. */
struct buf_chunk_t{

    ulint          size;          /*!< size of frames[] and blocks[] */

    unsigned char* mem;          /*!< pointer to the memory area which was allocated for the frames */

    ut_new_pfx_t   mem_pfx;      /*!< Auxiliary structure, describing "mem". It is filled by the allocator's
                                alloc method and later passed to the deallocate method. */

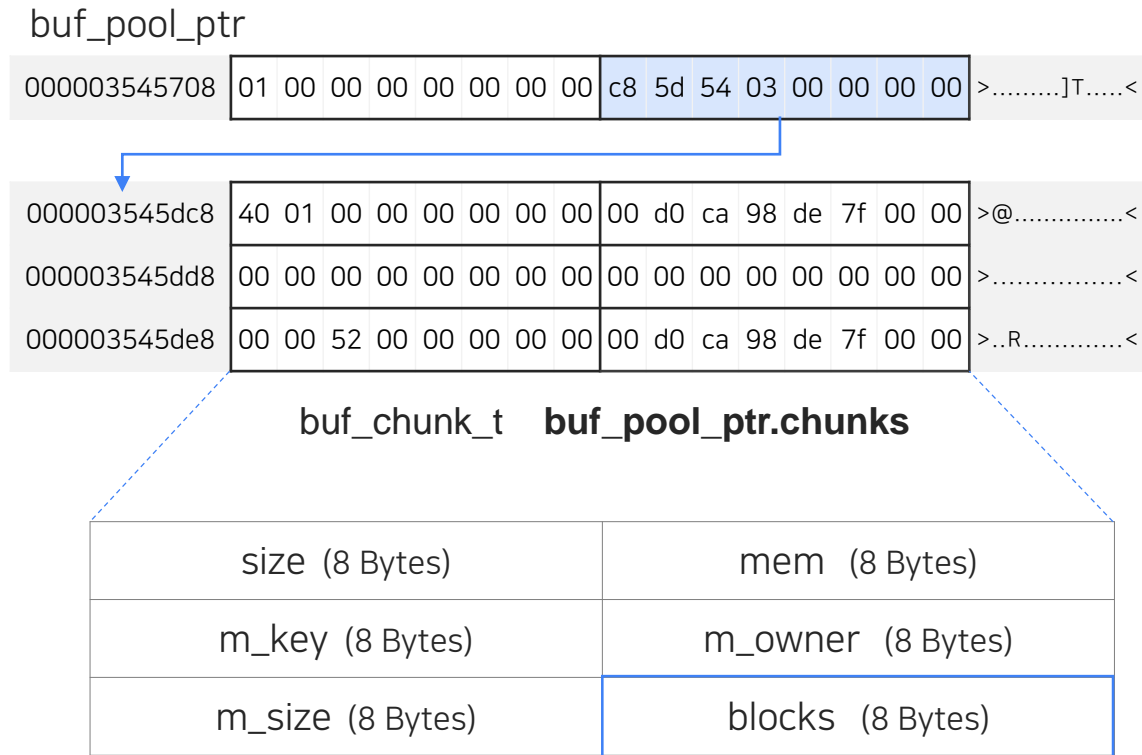
    buf_block_t*   blocks;       /*!< array of buffer control blocks */

    /** Get the size of 'mem' in bytes. */
    size_t mem_size() const {
        return(mem_pfx.m_size);
    }
};
```

Member	Description
size	frames[]와 blocks[]의 사이즈
mem	frames가 할당된 메모리 영역에 대한 포인터
mem_pfx	mem을 설명하는 보조적 구조
blocks	buffer control blocks의 array 시작 주소



buf_pool_ptr.chunks



```
*buf_pool_ptr.chunks
$23 = {
    size = 0x140, → "320"
    mem = 0x7fde98cad000,
    mem_pfx = {
        m_key = 0x0,
        m_owner = 0x0,
        m_size = 0x520000
    },
    blocks = 0x7fde98cad0
00
}
```



buf_block_t

./storage/innobase/include/buf0buf.h

```
struct buf_block_t{

    buf_page_t   page;    /*!< page information; this must be the first field, so that
                           buf_pool->page_hash can point to buf_page_t or buf_block_t */

    byte*        frame;   /*!< pointer to buffer frame which is of size UNIV_PAGE_SIZE, and
                           aligned to an address divisible by UNIV_PAGE_SIZE */

}
```

Member	Description
page (buf_page_t)	<ul style="list-style-type: none">• 페이지 정보, buf_page_t / buf_block_t를 포인트• id = { m_space, m_page_no, m_fold }• state• hash• list = { prev, next }• LRU = { prev, next }
frame	buffer frame 포인트



buf_page_t

./storage/innobase/include/buf0buf.h

```
class buf_page_t {  
  
    buf_page_t*  hash;      /*!< node used in chaining to buf_pool->page_hash or buf_pool->zip_hash */  
  
    UT_LIST_NODE_T(buf_page_t) list; /*!< based on state, this is a list node, protected either by  
                                     buf_pool->mutex or by buf_pool->flush_list_mutex,  
                                     in one of the following lists in buf_pool:  
  
                                     - BUF_BLOCK_NOT_USED: free, withdraw  
                                     - BUF_BLOCK_FILE_PAGE: flush_list  
                                     - BUF_BLOCK_ZIP_DIRTY: flush_list  
                                     - BUF_BLOCK_ZIP_PAGE: zip_clean  
  
                                     */  
  
    UT_LIST_NODE_T(buf_page_t) LRU; /*!< node of the LRU list */  
};
```

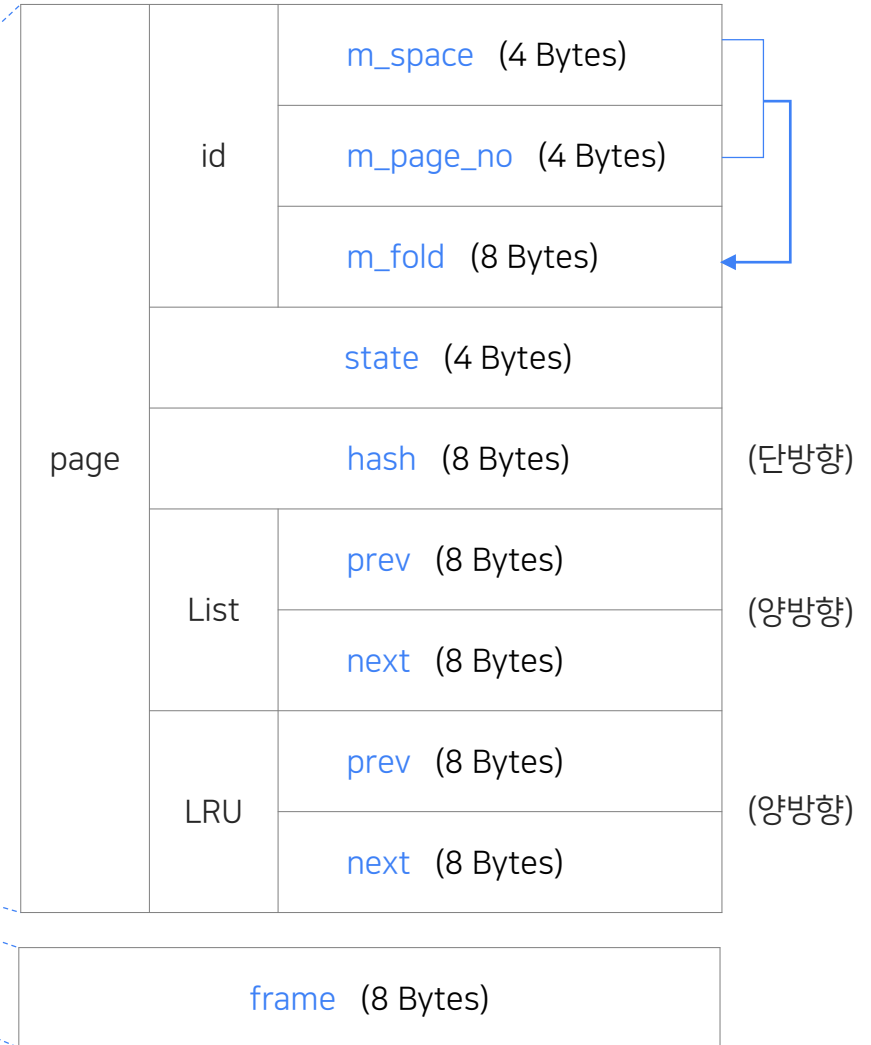
Member	Description
hash	buf_pool->page_hash or buf_pool->zip_hash
list	Free list
LRU	LRU list



Buf_pool_ptr.chunks.blocks

buf_pool_ptr

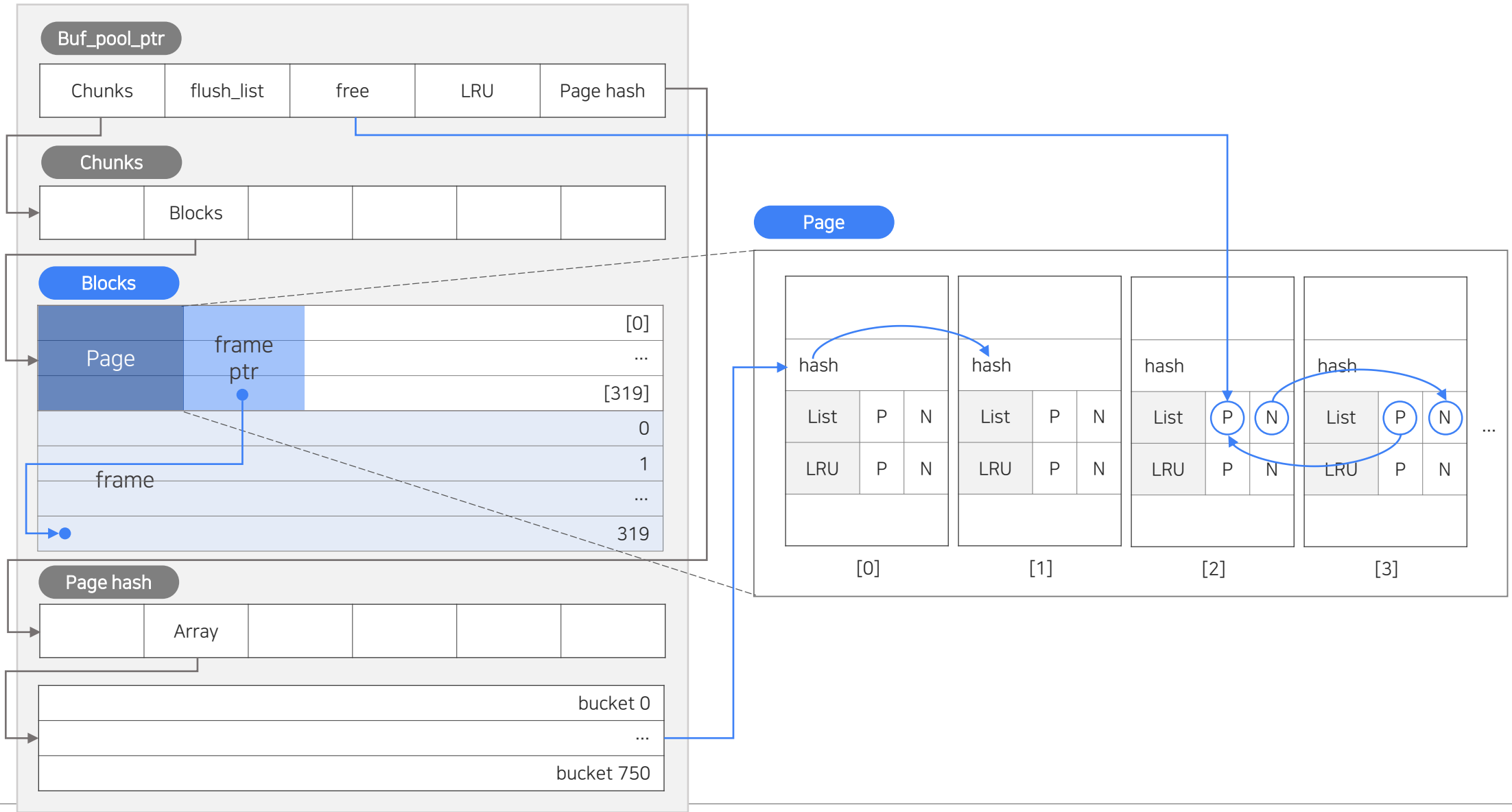
000003545de8	00 00 52 00 00 00 00 00	00 d0 ca 98 de 7f 00 00	>..R.....<
7fde98cad000	00 00 00 00 07 00 00 00	07 00 00 00 00 00 00 00	>.....<
7fde98cad010	00 40 00 00 00 40 00 00	00 00 00 00 00 00 00 00	>.@...@.....<
7fde98cad020	05 00 00 00 01 00 00 00	00 00 00 00 00 00 00 00	>.....<
7fde98cad030	00 00 00 00 00 00 00 00	00 00 00 00 00 00 00 00	>.....<
7fde98cad040	00 00 00 00 00 00 00 00	00 00 00 00 00 00 00 00	>.....<
7fde98cad050	00 00 00 00 00 00 00 00	ce 51 2e 00 00 00 00 00	>.....Q.....<
7fde98cad060	00 00 00 00 00 00 00 00	88 d1 ca 98 de 7f 00 00	>.....<
7fde98cad070	00 00 00 00 00 00 00 00	00 00 00 00 28 8f 2f 9b	>.....(./.<
7fde98cad080	00 c0 cc 98 de 7f 00 00	00 00 00 20 00 00 00 00	>.....<
	...		



buf_block_t **Buf_pool_ptr.chunks.blocks**

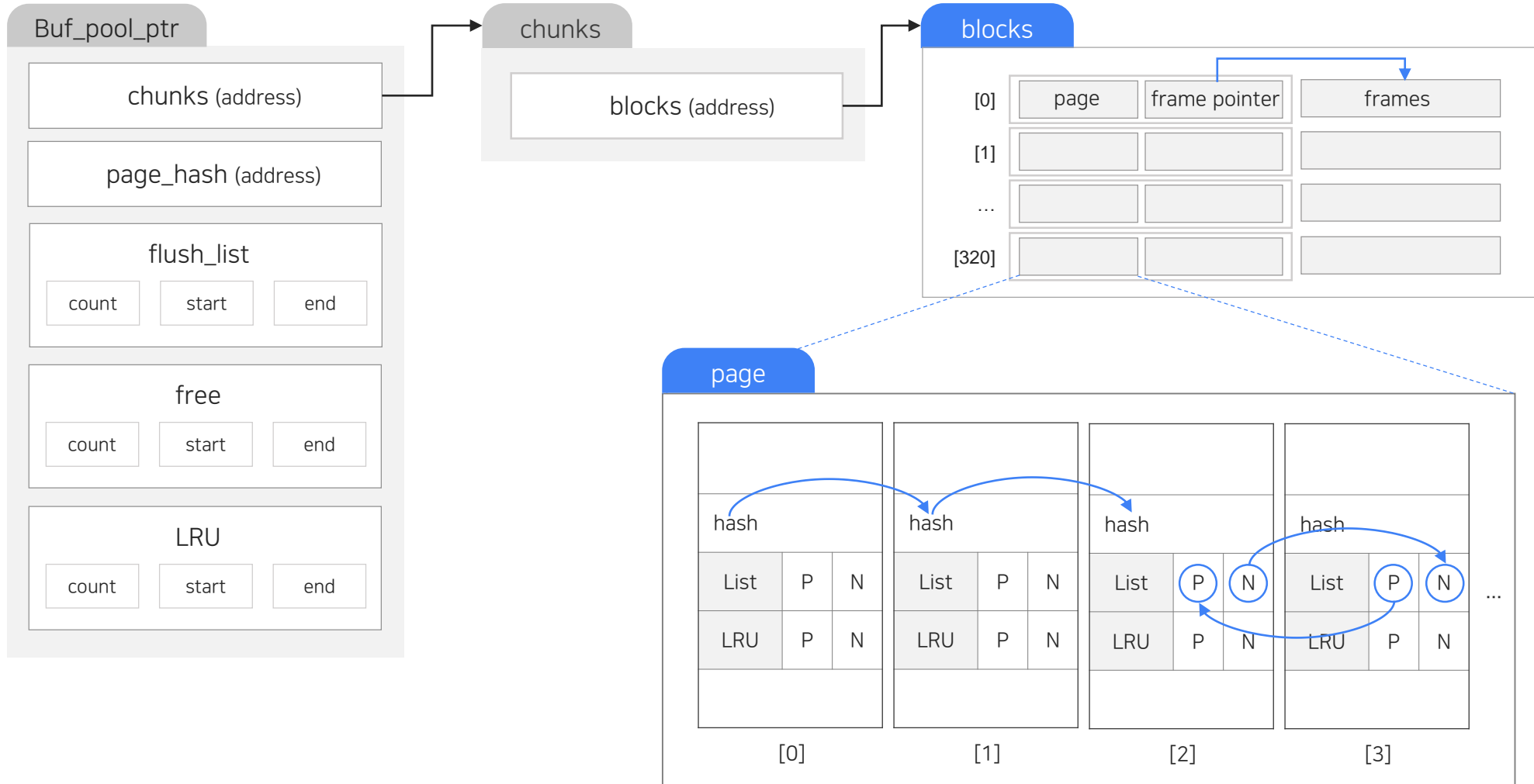


Buf_pool_ptr.chunks.blocks





Buf_pool_ptr.chunks.blocks





hash_table_t

./storage/innobase/include/hash0hash.h

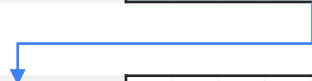
```
struct hash_table_t {  
    enum        hash_table_sync_t type; /*<! type of hash_table. */  
  
    ulint      n_cells; /* number of cells in the hash table */  
  
    hash_cell_t* array; /*!< pointer to cell array */  
  
    ulint      n_sync_obj; /* if sync_objs != NULL, then the number of either the number of  
                               mutexes or the number of rw_locks depending on the type. Must be a  
                               power of 2 */  
  
    union {  
  
        ib_mutex_t* mutexes; /* NULL, or an array of mutexes used to protect segments of the hash  
                                table */  
        rw_lock_t* rw_locks; /* NULL, or an array of rw_locks used to protect segments of the  
                                hash table */  
    } sync_obj;  
  
    mem_heap_t** heaps; /*!< if this is non-NULL, hash chain nodes for external chaining  
                           can be allocated from these memory heaps; there are then n_mutexes  
                           many of these heaps */  
  
    mem_heap_t* heap;  
  
};
```




Buf_pool_ptr.page_hash

buf_pool_ptr

000003545738	b8 7a 56 03 00 00 00 00	00 00 00 00 00 00 00 00	>.zV.....<
--------------	-------------------------	-------------------------	------------



000003567ab8	02 00 00 00 00 00 00 00	ef 02 00 00 00 00 00 00	>.....<
000003567ac8	18 7b 56 03 00 00 00 00	10 00 00 00 00 00 00 00	>.{V.....<
000003567ad8	b8 92 56 03 00 00 00 00	d8 ac 56 03 00 00 00 00	>..V.....V.....<
000003567ae8	00 00 00 00 00 00 00 00		>.....<

hash_table_t **buf_pool_ptr.page_hash**

type (8 Bytes)	n_cells (8 Bytes)
array (8 Bytes)	n_sync_obj (8 Bytes)
mutexes / rw_locks (8 Bytes)	heaps (8 Bytes)
heap (8 Bytes)	

02 ef = 751개



Buf_pool_ptr.page_hash.array

page_hash

000003567ac8	18 7b 56 03 00 00 00 00	10 00 00 00 00 00 00 00	>{V.....<
--------------	-------------------------	-------------------------	-----------

000003567b18	00 00 00 00 00 00 00 00	00 00 00 00 00 00 00 00	>.....<
000003567b28	00 00 00 00 00 00 00 00	00 00 00 00 00 00 00 00	>.....<
000003567b38	00 00 00 00 00 00 00 00	00 00 00 00 00 00 00 00	>.....<
000003567b48	00 00 00 00 00 00 00 00	00 00 00 00 00 00 00 00	>.....<
...			
000003567ba8	00 00 00 00 00 00 00 00	00 00 00 00 00 00 00 00	>.....<
000003567bb8	00 00 00 00 00 00 00 00	c8 dd ca 98 de 7f 00 00	>.....<
000003567bc8	40 dc ca 98 de 7f 00 00	00 00 00 00 00 00 00 00	>@.....<
...			

hash_cell_t **Buf_pool_ptr.page_hash.array**

0	0
1	0
2	0
...	...
21	7fde98caddc8
22	7fde98cad40
...	...
750	

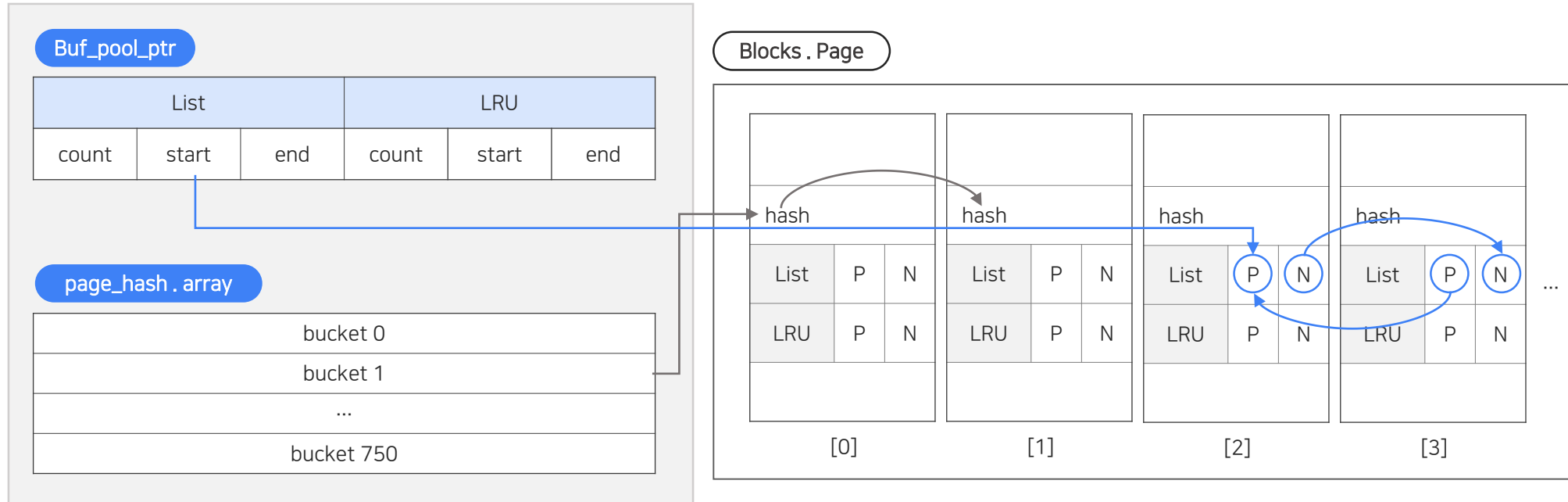
ut_find_prime

- (버퍼 개수 * 2) 보다 큰 소수
- $(320 * 2) = 640 < 751$

∴ 총 751개



Buf_pool_ptr.LRU, Buf_pool_ptr.free

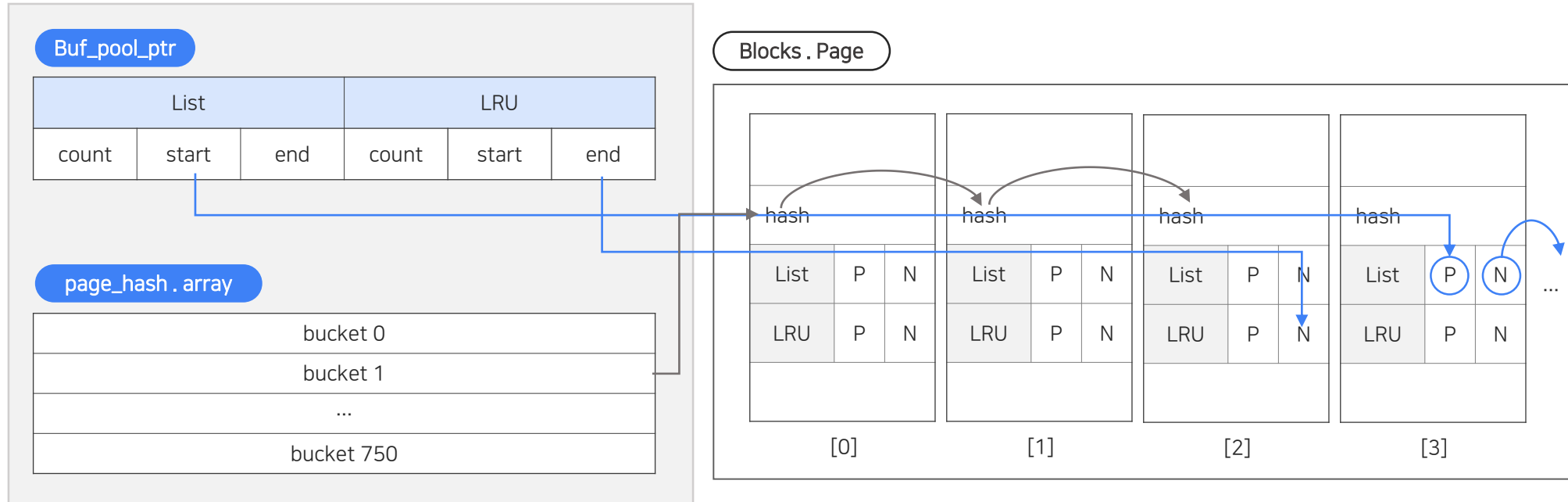


< 1번까지 할당된 경우 >

- 0번과 1번이 같은 버킷에 할당되었다면 hash로 연결 (단방향)
- 2번이 freelist의 시작, 2번에서 3번으로 이어짐



Buf_pool_ptr.LRU, Buf_pool_ptr.free

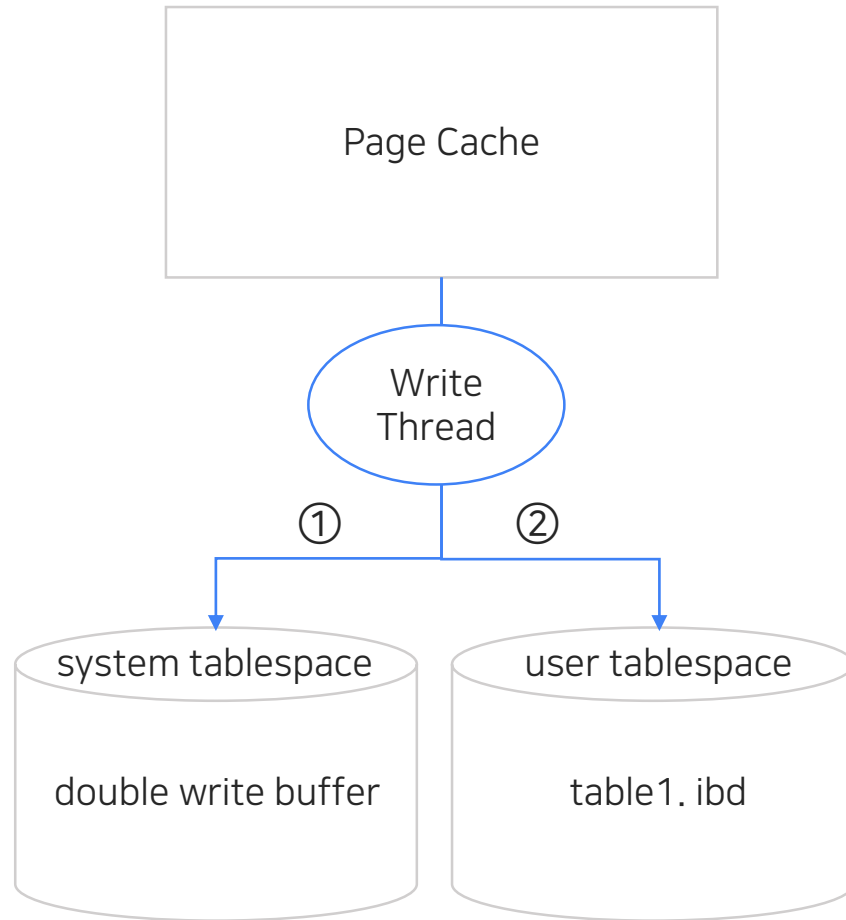


< 2번이 할당된 경우 >

- 0번, 1번, 2번이 같은 버킷에 할당되었다면, hash가 연결
- 2번은 freelist에서 LRU로 이동
- 3번은 freelist의 시작



[참고] Double Write Buffer



- innodb_doublewrite = ON
- 페이지가 데이터 파일의 적절한 위치에 쓰이기 전에 InnoDB 버퍼 풀에서 flush된 페이지를 내려쓰는 system tablespace에 위치한 스토리지 영역
- 페이지를 디스크에 내려 쓰다가 power failure 발생하여 half-written 페이지를 복구하기 위한 용도
- 데이터가 두 번 쓰이지만 doublewrite buffer는 두배의 I/O를 요구하지 않음 즉, sequential I/O로 처리되기 때문에 디스크 I/O가 2배가 되지 않음
- single fsync() call으로 large sequential chunk에서 데이터를 버퍼에 씀
- buffer 라는 이름을 가지고 있지만 디스크에 존재



시나리오

step 1	테이블 t1 select
step 2	Free Buffer 할당
step 3	변경을 위해 mutex 획득
step 4	tablespace, page no 등록 m_fold 생성
step 5	bucket에 등록
step 6	LRU에 등록

step 7	Disk I/O Frame에 데이터 load
step 8	Flush list 확인, hold io_write_thread
step 9	테이블 t1 Update 후, flush 대상 존재 여부 확인
step 10	double writer buffer 작동
step 11	release io_write_thread , flush list 와 double writer buffer 확인



step 1 > 테이블 t1 SELECT

Buf_pool_ptr

LRU		
count	start	end
96	0x7f7bd9fee178	0x7f7bd9fe5000

List		
count	start	end
224	0x7f7bd9fee300	0x7f7bda003878

Blocks

page.id		
m_space	m_page_no	m_fold
0	0	0

mutex		
m_lock_word		
0		

page_hash.array

node
0x0

session 2

THREAD_OS_ID	THREAD_ID	NAME
120705	18	thread/sql/one_connection
120689	6	thread/innodb/io_write_thread



step 2 > Free Buffer 할당

Buf_pool_ptr

LRU		
count	start	end
96	0x7f7bd9fee178	0x7f7bd9fe5000

List		
count	start	end
223	0x7f7bd9fee488	0x7f7bda003878

Blocks

page.id		
m_space	m_page_no	m_fold
0	0	0

mutex		
m_lock_word		
0		

page_hash.array

node
0x0



step 3 > 변경을 위한 mutex 획득

Buf_pool_ptr

LRU		
count	start	end
96	0x7f7bd9fee178	0x7f7bd9fe5000

List		
count	start	end
223	0x7f7bd9fee488	0x7f7bda003878

Blocks

page.id		
m_space	m_page_no	m_fold
0	0	0

mutex
m_lock_word
1

page_hash.array

node
0x0



step 4 > tablespace, page number 등록, fold 생성

Buf_pool_ptr

LRU		
count	start	end
96	0x7f7bd9fee178	0x7f7bd9fe5000

List		
count	start	end
223	0x7f7bd9fee488	0x7f7bda003878

Blocks

page.id		
m_space	m_page_no	m_fold
25	3	26214428

mutex		
m_lock_word		
1		

page_hash.array

node
0x0



step 5 > bucket에 등록

Buf_pool_ptr

LRU		
count	start	end
96	0x7f7bd9fee178	0x7f7bd9fe5000

List		
count	start	end
223	0x7f7bd9fee488	0x7f7bda003878

Blocks

page.id		
m_space	m_page_no	m_fold
25	3	26214428

mutex		
m_lock_word		
1		

page_hash.array

node
0x7f7bd9fee300



step 6 > LRU 에 등록

Buf_pool_ptr

LRU		
count	start	end
97	0x7f7bd9fee300	0x7f7bd9fe5000

List		
count	start	end
223	0x7f7bd9fee488	0x7f7bda003878

Blocks

page.id		
m_space	m_page_no	m_fold
25	3	26214428

mutex		
m_lock_word		
0		

page_hash.array

node
0x7f7bd9fee300



step 6 > LRU 에 등록

frame

7f1815b20000	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	> <	
7f1815b20010	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	> <
7f1815b20020	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	> <
7f1815b20030	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	> <
7f1815b20040	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	> <
7f1815b20050	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	> <
7f1815b20060	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	> <
7f1815b20070	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	> <
7f1815b20080	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	> <
7f1815b20090	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	> <
7f1815b200a0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	> <
--																			
7f1815b20190	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	> <
7f1815b201a0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	> <
7f1815b201b0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	> <
7f1815b201c0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	> <
7f1815b201d0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	> <
7f1815b201e0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	> <
7f1815b201f0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	> <
7f1815b20200	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	> <



step 7 > DISK I/O, Frame에 데이터 로드

frame

7f1815b20000	f7	c8	de	e0	00	00	00	03	ff	ff	ff	ff	ff	ff	ff	ff	> <
7f1815b20010	00	00	00	00	04	99	66	1e	45	bf	00	00	00	00	00	00	> f.E <
7f1815b20020	00	00	00	00	00	19	00	02	0a	78	80	07	00	00	00	00	> x <
7f1815b20030	08	83	00	02	00	04	00	05	00	00	00	00	00	00	00	00	> <
7f1815b20040	00	00	00	00	00	00	00	00	00	29	00	00	00	19	00	00	>) <
7f1815b20050	00	02	00	f2	00	00	00	19	00	00	00	02	00	32	01	00	> 2 .. <
7f1815b20060	02	00	20	69	6e	66	69	6d	75	6d	00	06	00	0b	00	00	> .. infimum <
7f1815b20070	73	75	70	72	65	6d	75	6d	72	71	f0	80	15	00	00	00	> supremumrq <
7f1815b20080	10	02	00	52	52	52	5f	5f	5f	5f	5f	5f	31	30	30	30	> ...RRR_____1000<
7f1815b20090	30	30	30	30	30	30	30	31	00	00	00	01	1a	0c	a1	00	> 00000001 <
7f1815b200a0	00	00	04	01	10	46	20	20	20	20	20	20	20	20	20	20	> F <
--																	
7f1815b20190	20	20	20	20	20	31	31	31	20	20	20	20	20	20	20	20	> 111 <
7f1815b201a0	20	20	20	20	20	20	20	20	20	20	20	20	20	20	20	20	> <
7f1815b201b0	20	20	20	20	20	20	20	20	20	20	20	20	20	20	20	20	> <
7f1815b201c0	20	20	20	20	20	20	20	20	20	20	20	20	20	20	20	20	> <
7f1815b201d0	20	20	20	20	20	20	20	20	20	20	20	20	20	20	20	20	> <
7f1815b201e0	20	20	20	20	20	20	20	20	20	20	20	20	20	20	20	20	> <
7f1815b201f0	20	20	20	20	20	20	20	20	20	20	20	20	20	20	20	20	> <
7f1815b20200	20	20	20	20	20	20	5e	5e	5e	20	20	20	20	20	20	20	> ^^^ <



step 8 > Flush 대상 존재 여부 확인, hold io_write_thread

Buf_pool_ptr

Flush		
count	start	end
0	0x0	0x0

```
session 2
+-----+-----+-----+
| THREAD_OS_ID | THREAD_ID | NAME |
+-----+-----+-----+
| 120705 | 18 | thread/sql/one_connection |
| 120689 | 6 | thread/innodb/io_write_thread |
+-----+-----+-----+
```

hold io_write_thread





step 9 > 테이블 t1 Update 후, flush 대상 존재 여부 확인

Buf_pool_ptr

Flush		
count	start	end
3	0x7f7bd9fe5c40	0x7f7bd9fe5dc8

0	0x7f7bd9fe5c40	1	3
1	0x7f7bd9fee300	25	3
2	0x7f7bd9fe5dc8	1	5

double write buffer

first_free	b_reserved
0	0



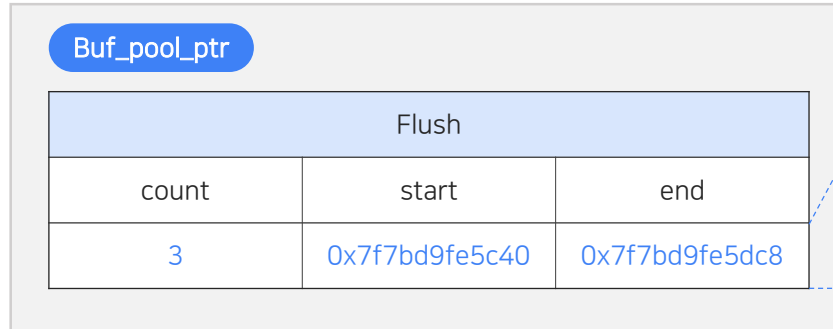
step 9 > 테이블 t1 Update 후, flush 대상 존재 여부 확인

frame

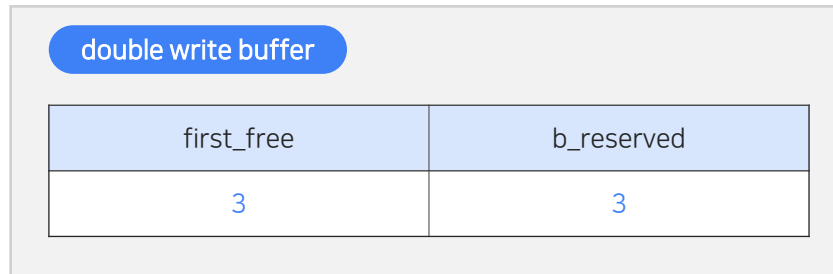
7f1815b20000	f7	c8	de	e0	00	00	00	03	ff	ff	ff	ff	ff	ff	ff	ff	> <
7f1815b20010	00	00	00	00	04	99	66	1e	45	bf	00	00	00	00	00	00	> f.E <
7f1815b20020	00	00	00	00	00	19	00	02	0a	78	80	07	00	00	02	2b	> x + <
7f1815b20030	08	83	00	05	00	00	00	05	00	00	00	00	00	00	00	00	> <
7f1815b20040	00	00	00	00	00	00	00	00	00	29	00	00	00	19	00	00	>) <
7f1815b20050	00	02	00	f2	00	00	00	19	00	00	00	02	00	32	01	00	> 2 .. <
7f1815b20060	02	00	20	69	6e	66	69	6d	75	6d	00	06	00	0b	00	00	>.. infimum..... <
7f1815b20070	73	75	70	72	65	6d	75	6d	03	71	f0	80	15	00	00	00	>supremum.q..... <
7f1815b20080	10	02	00	52	52	52	5f	5f	5f	5f	5f	5f	31	30	30	30	>...RRR_____1000<
7f1815b20090	30	30	30	30	30	30	30	31	00	00	00	01	1c	04	21	00	>00000001.....!.<
7f1815b200a0	00	00	05	20	a4	46	20	20	20	20	20	20	20	20	20	20	>... .F <
--																	
7f1815b20190	20	20	20	20	20	31	31	31	20	20	20	20	20	20	20	20	> 111 <
7f1815b201a0	20	20	20	20	20	20	20	20	20	20	20	20	20	20	20	20	> <
7f1815b201b0	20	20	20	20	20	20	20	20	20	20	20	20	20	20	20	20	> <
7f1815b201c0	20	20	20	20	20	20	20	20	20	20	20	20	20	20	20	20	> <
7f1815b201d0	20	20	20	20	20	20	20	20	20	20	20	20	20	20	20	20	> <
7f1815b201e0	20	20	20	20	20	20	20	20	20	20	20	20	20	20	20	20	> <
7f1815b201f0	20	20	20	20	20	20	20	20	20	20	20	20	20	20	20	20	> <
7f1815b20200	20	20	20	20	20	20	40	40	40	20	20	20	20	20	20	20	> @@@ <



step 10 > double writer buffer 작동



0	0x7f7bd9fe5c40	1	3
1	0x7f7bd9fee300	25	3
2	0x7f7bd9fe5dc8	1	5





step 10 > double writer buffer 작동

2	0x7f7bd9fe5dc8	1	5
1	0x7f7bd9fee300	25	3
0	0x7f7bd9fe5c40	1	3

frame

7f1812c54000	98	8a	53	0f	00	00	00	05	00	00	00	00	00	00	00	00	> ..S.....<
7f1812c54010	00	00	00	00	04	99	89	e9	00	02	00	00	00	00	00	00	>.....<
7f1812c54020	00	00	00	00	00	01	00	02	20	a4	23	c4	ff	ff	ff	ff	>..... .#.....<
7f1812c54030	00	00	ff	ff	ff	ff	00	00	00	02	1f	ea	00	00	00	01	>.....<
7f1812c54040	00	00	00	02	01	b2	00	00	00	01	00	00	00	05	00	2c	>....., <
--																	
7f1812c58000	6d	58	e2	73	00	00	00	03	ff	ff	ff	ff	ff	ff	ff	ff	> mX.s.....<
7f1812c58010	00	00	00	00	04	99	89	95	45	bf	00	00	00	00	00	00	>.....E.....<
7f1812c58020	00	00	00	00	00	19	00	02	0a	78	80	07	00	00	02	2b	>.....x.....+<
7f1812c58030	08	83	00	05	00	00	00	05	00	00	00	00	00	00	00	00	>.....<
7f1812c58040	00	00	00	00	00	00	00	00	00	29	00	00	00	19	00	00	>.....).....<
--																	
7f1812c5c000	20	10	4b	8b	00	00	00	03	00	00	00	00	00	00	00	00	> .K.....<
7f1812c5c010	00	00	00	00	04	99	89	e9	00	06	00	00	00	00	00	00	>.....<
7f1812c5c020	00	00	00	00	00	01	ff	ff	ff	fe	00	00	00	00	00	00	>.....<
7f1812c5c030	00	0b	00	00	00	05	20	0c	00	00	00	05	00	78	00	00	>..... .x..<
7f1812c5c040	00	01	00	00	00	02	00	32	00	00	00	05	00	00	00	04	>..... 2.....<



step 11 > flush list와 double writer buffer 확인, release io_write_thread

Buf_pool_ptr

Flush		
count	start	end
0	0x0	0x0

double write buffer

first_free	b_reserved
0	0

session 2

THREAD_OS_ID	THREAD_ID	NAME
120705	18	thread/sql/one_connection
120689	6	thread/innodb/io_write_thread

release io_write_thread





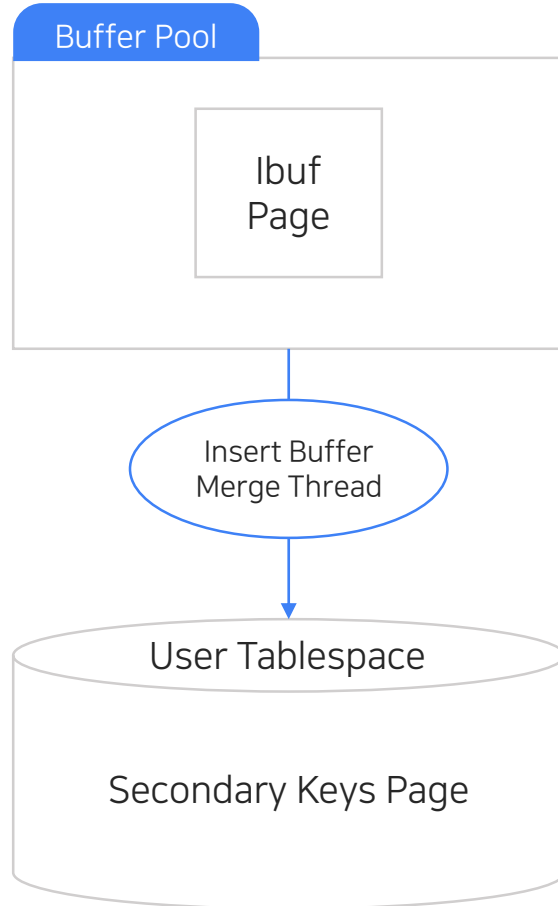
InnoDB 특징

- 1. 역사
- 2. 아키텍처
- 3. 메모리
 - 3.1 메모리 아키텍처
 - 3.2 Buffer Pool
 - 3.3 Insert Buffer
 - 3.4 Redo Log Buffer
- 4. 디스크
- 5. 성능





Insert Buffer



- DML이 발생할 때 Secondary Index에 변경이 발생하는 경우 성능 향상을 위해 사용하는 기능
- secondary index의 disk I/O를 줄이기 위해 사용
 - 일반적으로 PK key 순으로 이루어지는 경우가 많아 PK index는 random read가 많이 발생하지 않음
 - secondary index는 non-unique 경우가 많고, random order 작업이 많이 발생하기 때문에 disk I/O 많이 발생



Insert Buffer

Value	Description
none	Do not buffer any operations
inserts	Buffer insert operations
deletes	Buffer delete marking operations; strictly speaking, the writes that mark index records for later deletion during a purge operation
changes	Buffer inserts and delete-marking operations
purges	Buffer the physical deletion operations that happen in the background
all	(default) Buffer inserts, delete-marking operations, and purges



Insert Buffer

./storage/innobase/include/ibuf0ibuf.ic

```
/** Insert buffer struct */
struct ibuf_t{
  ulint          size;      /*!< current size of the ibuf index tree, in pages */
  dict_index_t* index;     /*!< insert buffer index */
  ulint          n_merges;  /*!< number of pages merged */
  ulint          n_merged_ops[IBUF_OP_COUNT];
                  /*!< number of operations of each type merged to index pages */
  ulint          n_discarded_ops[IBUF_OP_COUNT];
                  /*!< number of operations of each type discarded without merging due to the
tablespace being deleted or the index being dropped */
};
```




Insert Buffer

Member	Description
size	Insert Buffer 로 할당된 페이지 수
index	Insert Buffer 로 할당된 페이지의 인덱스
n_merges	Merge 된 페이지 수
n_merged_ops	Merge operation의 상세(insert, delete mark, delete 개수)
n_discarded_ops	Merge 취소 operation의 상세(insert, delete mark, delete 개수)

```
-----  
INSERT BUFFER AND ADAPTIVE HASH INDEX  
-----
```

```
Ibuf: size 1, free list len 126, seg size 128, 14 merges  
merged operations: insert 24, delete mark 28, delete 0  
discarded operations: insert 0, delete mark 0, delete 0
```



InnoDB 특징

- 1. 역사
- 2. 아키텍처
- 3. 메모리
 - 3.1 메모리 아키텍처
 - 3.2 Buffer Pool
 - 3.3 Insert Buffer
 - 3.4 Redo Log Buffer
- 4. 디스크
- 5. 성능



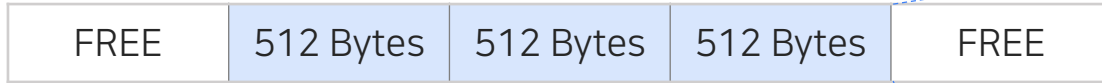


Redo Log Buffer

Member	Description
innodb_log_buffer_size	리두 로그 버퍼의 사이즈
innodb_flush_log_at_trx_commit	리두 로그가 디스크에 기록되는 방법 지정 <ul style="list-style-type: none">• 0 = write + flush per second• 1 = write + flush per transaction (default)• 2 = write per transaction + flush per innodb_flush_log_at timeout
innodb_flush_log_at_timeout	리두 로그 flush의 빈도 설정



Redo Log Buffer



로그 버퍼

- 512 Bytes
- 리두 로그 파일의 블록 사이즈와 동일

LOG_BLOCK_HDR_SIZE

LOG_BLOCK_TRL_SIZE

LOG_BLOCK_HDR_NO (4 Bytes) 해당 블록의 LSN을 블록 번호로 전환
LOG_BLOCK_HDR_DATA_LEN (2 Bytes) 해당 블록에 쓰여지는 로그 길이
LOG_BLOCK_FIRST_REC_GROUP (2 Bytes) 해당 로그 블록의 mtr 로그 레코드 그룹의 첫 번째 시작 offset
LOG_BLOCK_CHECKPOINT_NO (4 Bytes) 하나의 블록을 다 쓸 경우 혹은 가장 마지막 한 개 블록이 파일에 쓰일 경우, 해당 위치에 log_sys -> Next_checkpoint_no로 쓴다
로그 기록
LOG_BLOCK_CHECKSUM (4 Bytes) 블록의 체크섬 값을 저장



Global In-memory Redo Log Buffer

- 버퍼 크기 설정 파라미터 : `innodb_log_buffer_size`
- 버퍼 크기 기본 값 : 8MB
- redo log buffer가 가득차면, `log_buffer_flush_to_disk()`를 사용하여 동기화 flush
 - `log_sys -> buf_next_to_write` 에서 `log_sys -> buff_free`
 - `log_sys -> flushed_to_disk_lsn` 에서 `log_sys -> lsn`



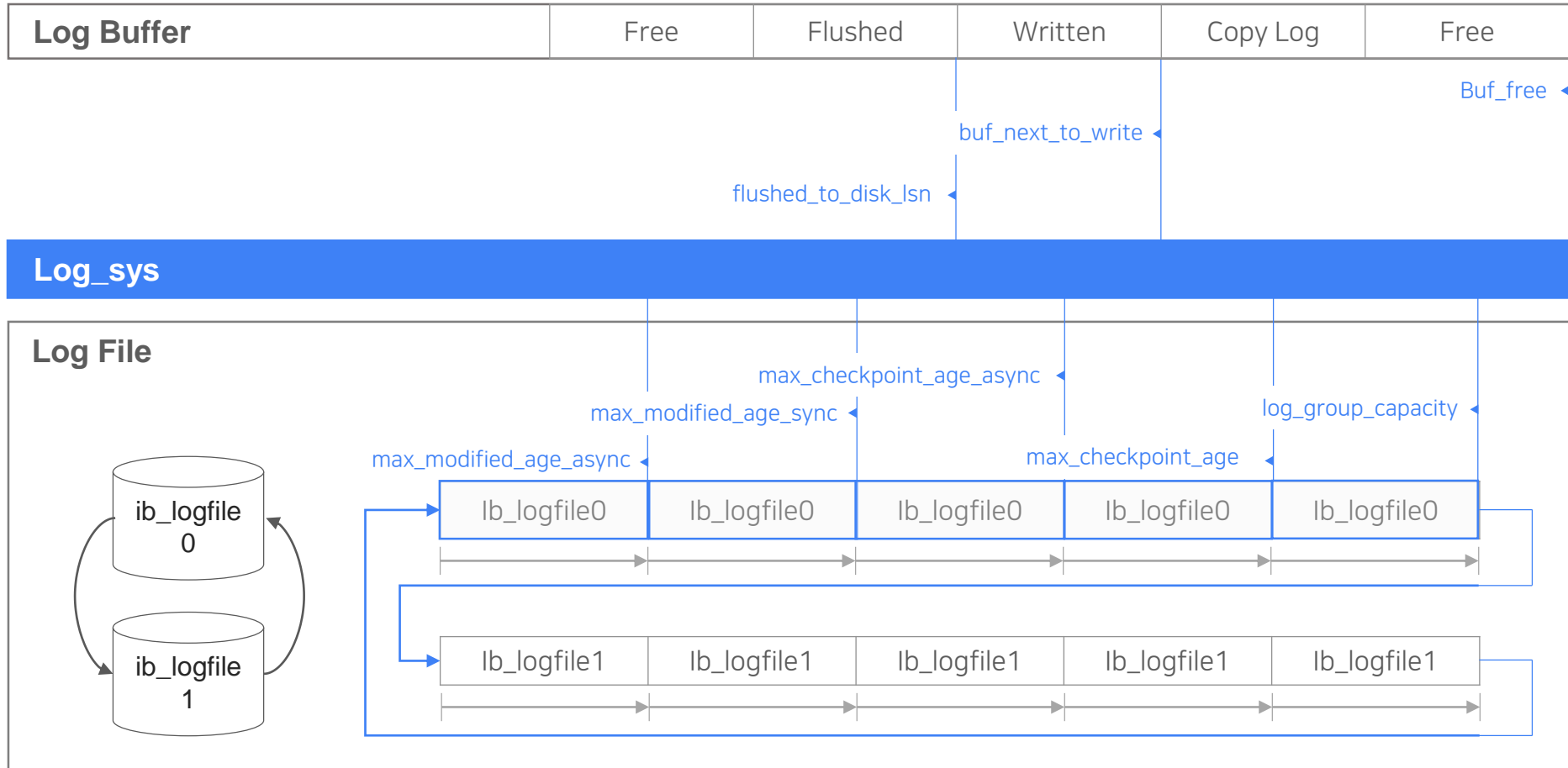
Global Log System Object

- `log_sys` (`log_t` 타입)
- redo log buffer, redo log file과 연관된 정보를 포함
- log buffer의 `Active area`를 식별 (Active area : 디스크에 쓰이기 전의 redo log를 저장)

member	Description
<code>log_sys -> buf</code>	in-memory redo log buffer
<code>log_sys -> buf_free</code>	in-memory redo log buffer내의 offset
<code>log_sys -> buf_next_to_write</code>	redo log file에 쓰여지지 않은 redo log record의 offset
<code>log_sys -> flushed_to_disk_lsn</code>	redo log file에 기록하고 flush 되어야 하는 곳까지 lsn 표기
<code>log_sys -> lsn</code>	현재 lsn
<code>log_sys -> write_lsn</code>	현재 동작중인 redo log buffer 쓰기 작업의 마지막 lsn 표기
<code>log_sys -> current_flush_lsn</code>	현재 동작중인 redo log buffer 쓰기 작업과 flush 작업의 마지막 lsn



Log_sys, Log Buffer, Log File의 관계

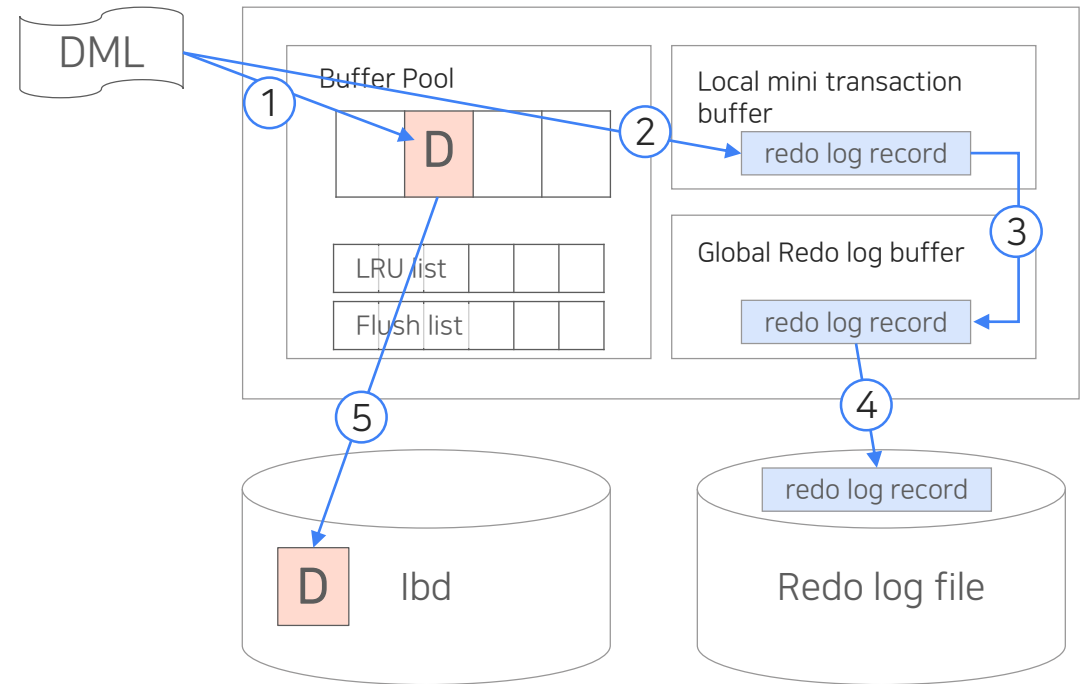




Redo Log Generation

Redo logging Process (Write-ahead logging)

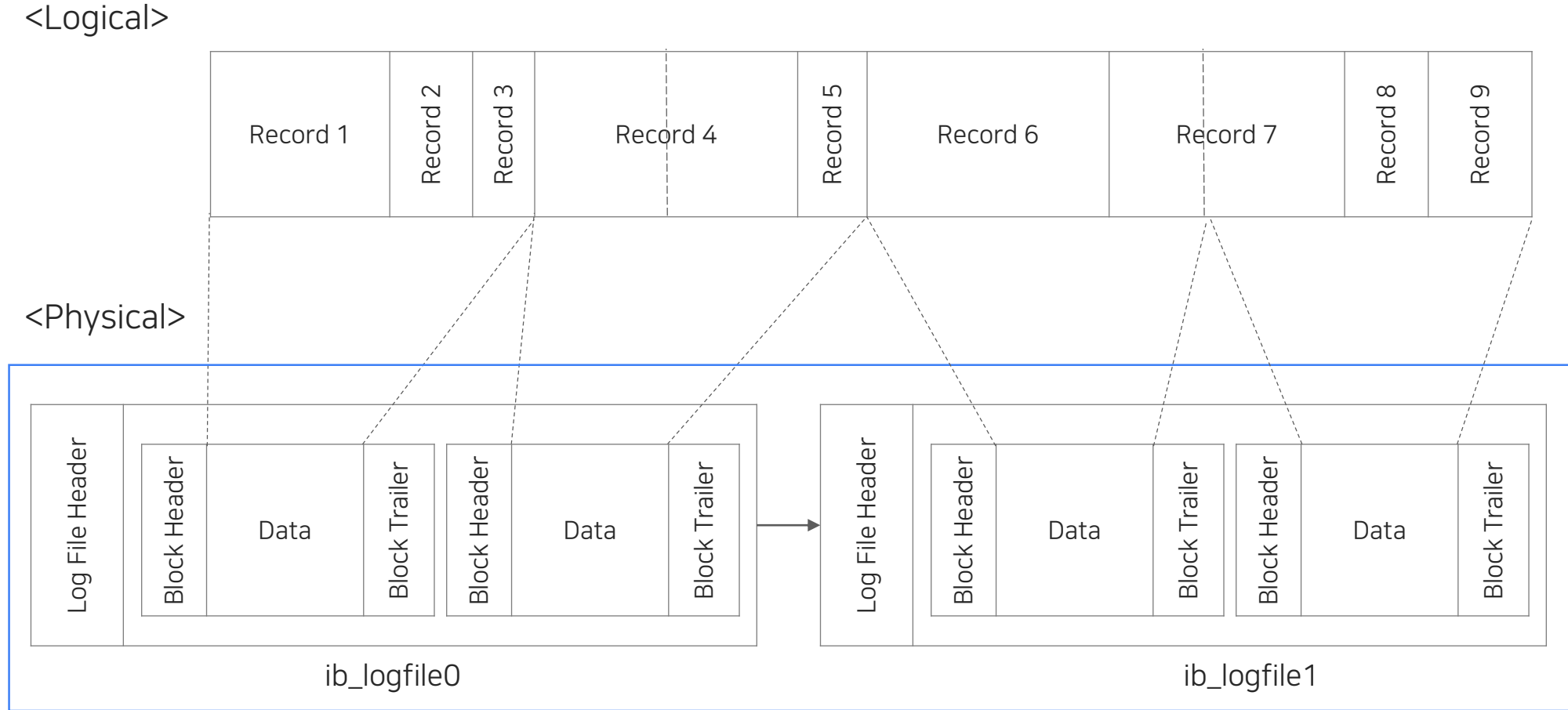
1. 페이지가 변경되면, 가장 먼저 in-memory에 페이지가 복사됨. 메모리에서 페이지가 수정되고, 디스크에 플러시 되기 전까지 해당 페이지는 dirty page로 메모리에 남아있음
2. 메모리에 생성된 리두 로그는 mini transaction buffer에 위치
3. global in-memory redo log buffer로 이동
4. redo log record는 메모리내의 redo log buffer에서 디스크의 redo log file로 쓰여짐
5. Dirty page는 체크포인트 작업 중에 메모리에서 디스크로 flush됨



참고 : <https://cafe.naver.com/mysqlpg>



Log Group 구조



Log Group 1



InnoDB 특징

- 1. 역사
- 2. 아키텍처
- 3. 메모리
- 4. 디스크
- 5. 성능

- 4.1 Redo Log File
- 4.2 Logical Structure
 - 4.2.1 Tablespace
 - 4.2.1.1 IBD File의 구조
 - 4.2.2 Segment, Extent
 - 4.2.3 Page
 - 4.2.3.1 Page Layout
 - 4.2.3.2 Directory Slot
 - 4.2.3.3 Page 내부 동작원리
- 4.3 Undo, MVCC Architecture
- 4.4 Purge
 - 4.4.1 Purge 정의 및 필요성
 - 4.4.2 Purge Update Undo
 - 4.4.3 Purge Garbage Space



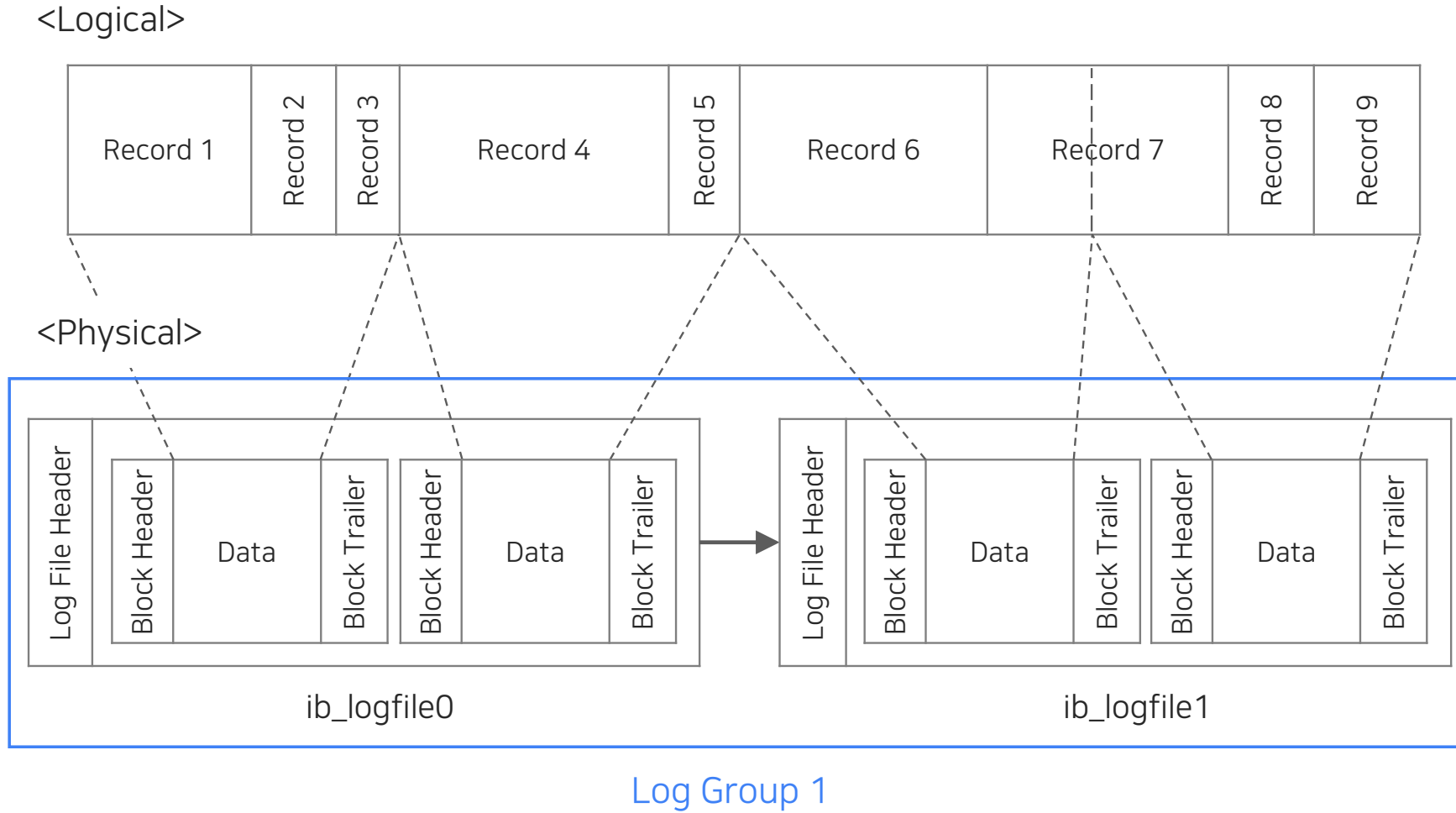
InnoDB 특징

- 1. 역사
- 2. 아키텍처
- 3. 메모리
- 4. 디스크
- 5. 성능

- 4.1 Redo Log File
- 4.2 Logical Structure
 - 4.2.1 Tablespace
 - 4.2.1.1 IBD File의 구조
 - 4.2.2 Segment, Extent
 - 4.2.3 Page
 - 4.2.3.1 Page Layout
 - 4.2.3.2 Directory Slot
 - 4.2.3.3 Page 내부 동작원리
- 4.3 Undo, MVCC Architecture
- 4.4 Purge
 - 4.4.1 Purge 정의 및 필요성
 - 4.4.2 Purge Update Undo
 - 4.4.3 Purge Garbage Space



Redo Log File



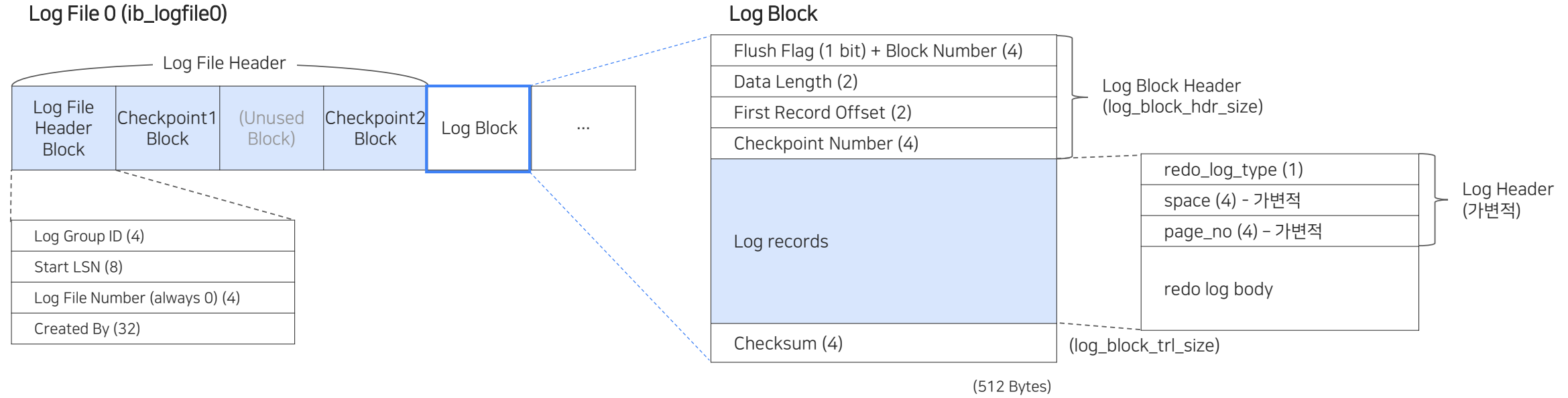


Redo Log File 특징

- 기본적으로 2개 파일 생성 (ib_logfile0, ib_logfile1)
 - 한 그룹의 파일 개수 파라미터 : innodb_log_files_in_group
- 각 사이즈는 48MB (default)
 - 그룹 내 각 파일의 사이즈 : innodb_log_file_size
- redo log 전체 사이즈 = innodb_log_files_in_group * innodb_log_file_size
- 5.6 버전에서 InnoDB가 하나의 log group만 지원
- 순환 방식으로 사용됨
- log blocks (OS_FILE_LOG_BLOCK_SIZE)
- log file (LOG_FILE_HDR_SIZE = 4 * OS_FILE_LOG_BLOCK_SIZE)
- fsync() System Call을 이용하여 디스크에 flush



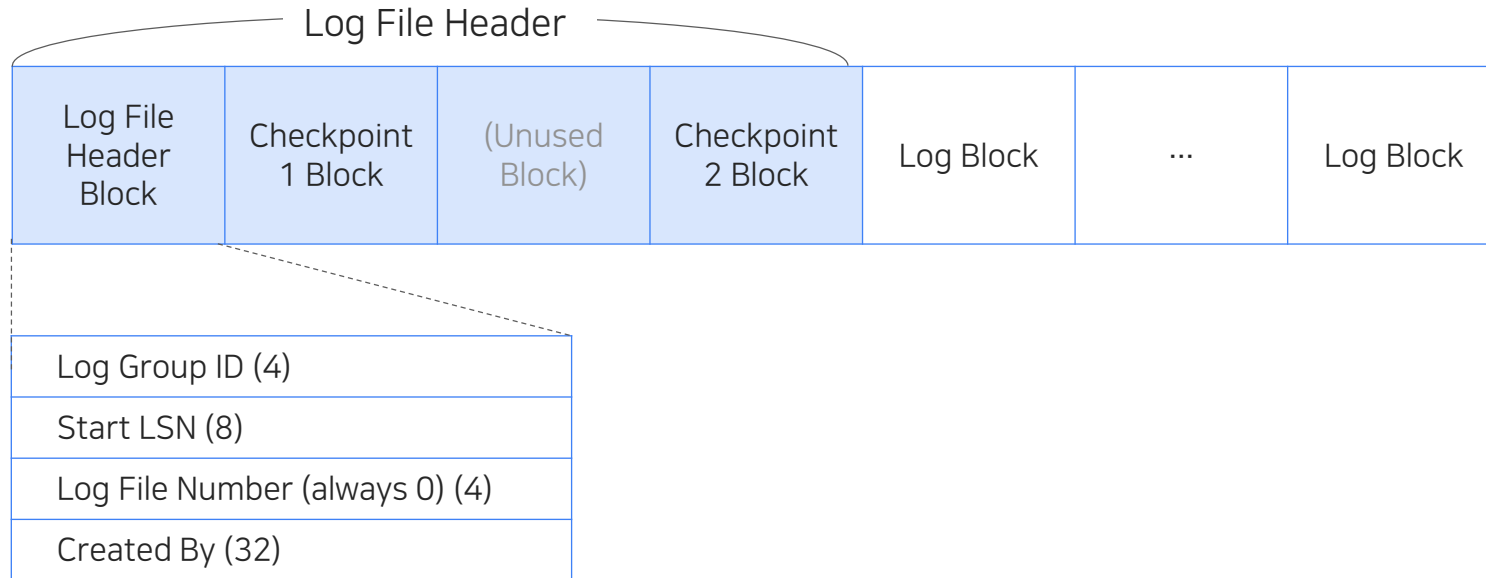
Redo Log File





Redo Log File 구조

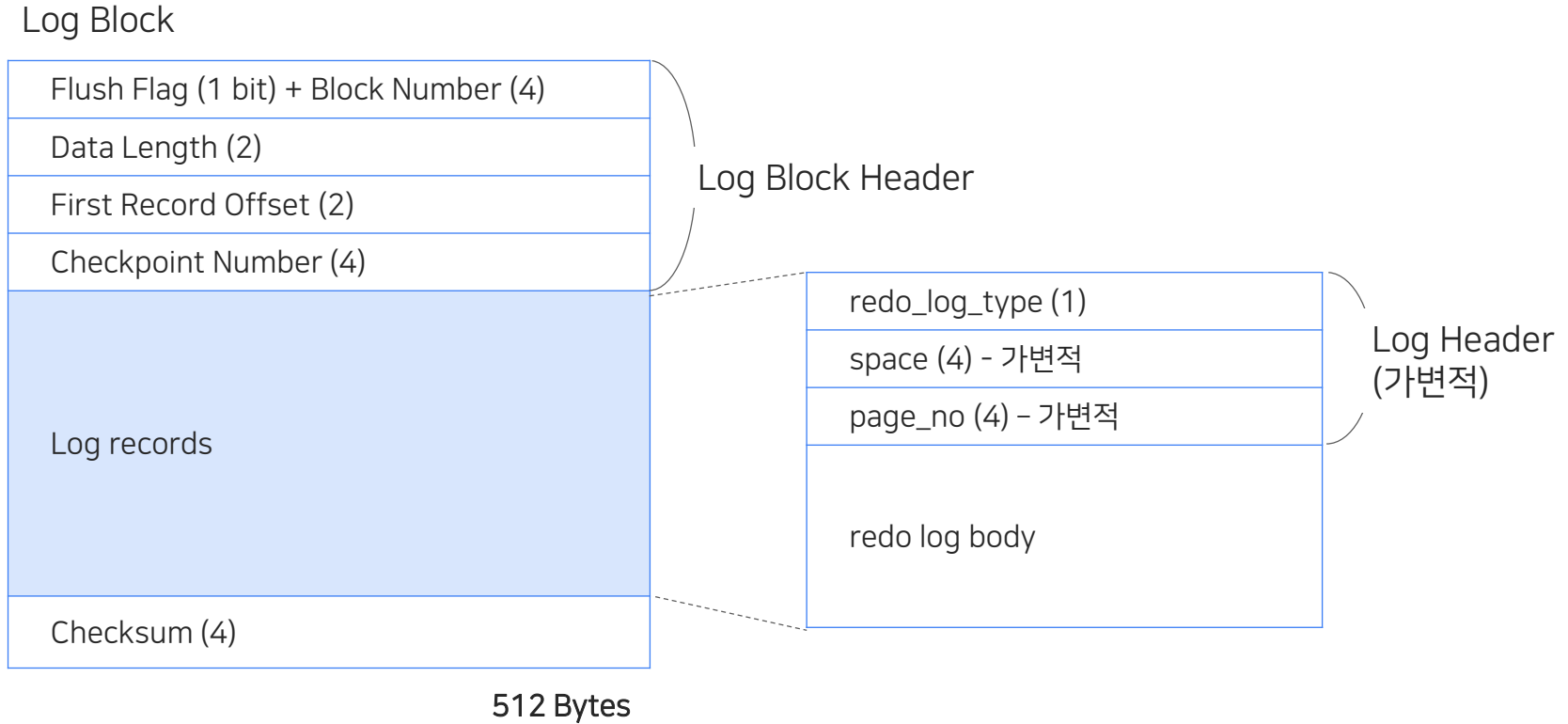
Log File 0 (ib_logfile0)



- 4개의 log block
- 두 번째 로그 블록 시작점에 첫 번째 체크포인트
- 네 번째 로그 블록 시작점에 두 번째 체크포인트

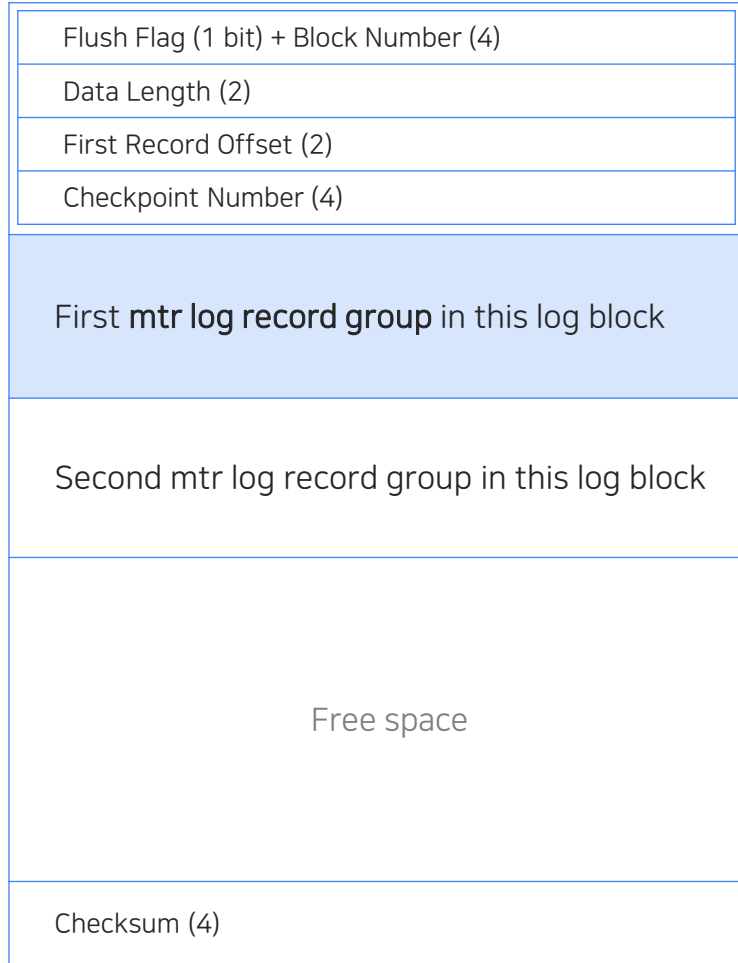


Redo Log Block 및 Log Record의 구조



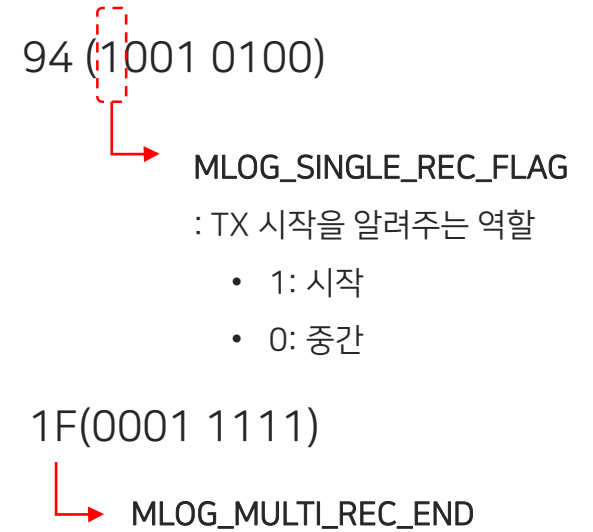


Mini Transaction (MTR)



1 Byte

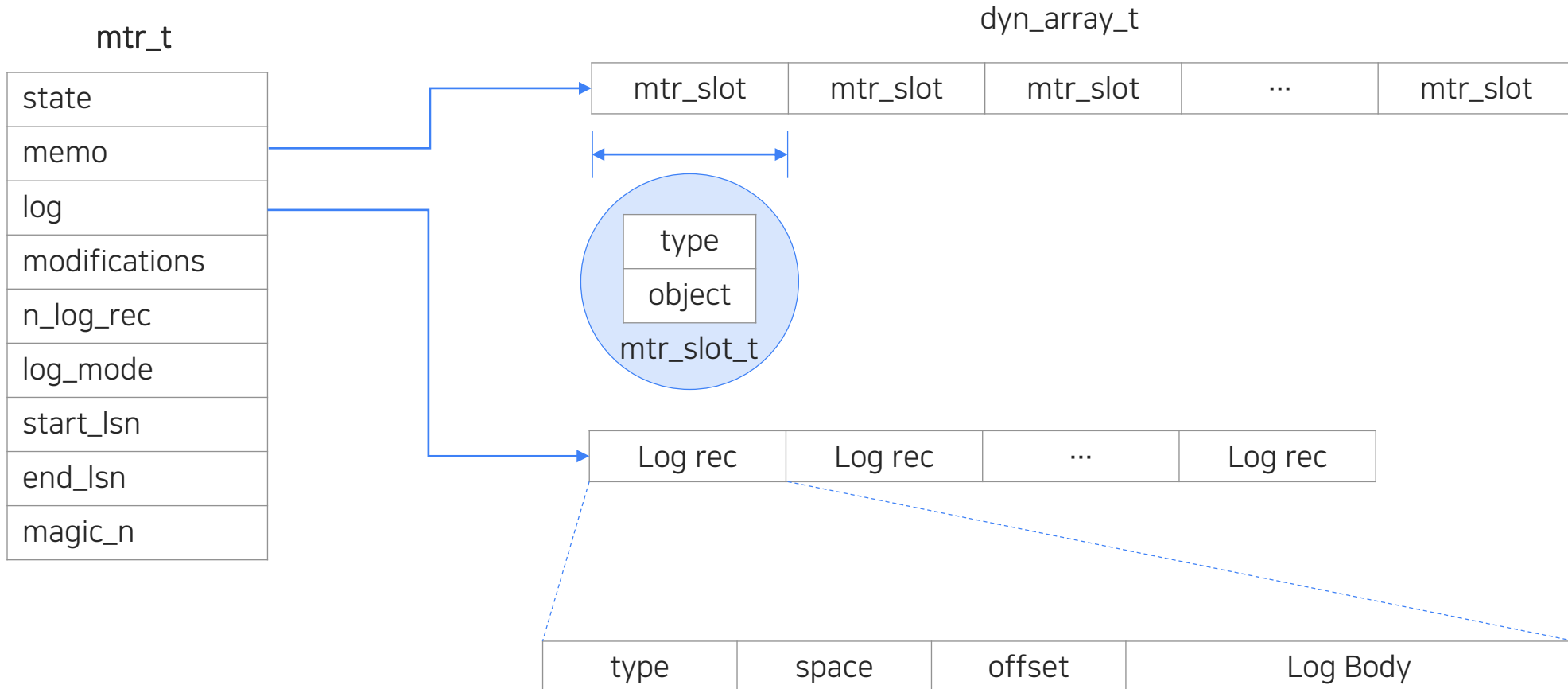
mtr1	94	
mtr2	26	
mtr3	37	
		1F



- mtr_t 타입
- 모든 redo log record 생성에 사용
- Dirty Page (수정된 페이지의 리스트) 포함
- redo log record 그룹 생성 시, 하나의 mini transaction으로 처리



Mini Transaction (MTR)





Mini Transaction (MTR)

./storage/innobase/include/mtr0mtr.h

```
/* Mini-transaction handle and buffer */
struct  mtr_t{
    dyn_array_t  memo;           /*!< memo stack for locks etc. */
    dyn_array_t  log;           /*!< mini-transaction log */
    unsigned     inside_ibuf:1; /*!< TRUE if inside ibuf changes */
    unsigned     modifications:1; /*!< TRUE if the mini-transaction modified buffer pool pages */
    unsigned     made_dirty:1;  /*!< TRUE if mtr has made at least one buffer pool page dirty */
    ulint        n_log_recs;    /* count of how many page initial log records have been written
                                to the mtr log */
    ulint        n_freed_pages; /* number of pages that have been freed in this mini-
                                transaction */
    ulint        log_mode;     /* specifies which operations should be logged; default value
                                MTR_LOG_ALL */
    lsn_t        start_lsn;    /* start lsn of the possible log entry for this mtr */
    lsn_t        end_lsn;     /* end lsn of the possible log entry for this mtr */
};
```



Mini Transaction (MTR)

mtr_start()	mini transaction을 시작, mini transaction buffer를 초기화
mlog_write_ulint()	redo log record 생성
mtr_commit()	<ul style="list-style-type: none">• mini transaction을 commit• redo log record가 mini transaction에서 global redo log buffer로 이동• dirty pages 리스트는 buffer pool flush 리스트에 추가



Log Sequence Number (LSN)

- redo log file 내부의 [offset](#)
- lsn_t 타입 (8-byte unsigned integer)
- dirty page, redo log record, redo log file 연결
- in memory log buffer에 복사될 때, 연관된 LSN 값을 가짐
- 페이지의 LSN은 각 페이지의 헤더에 저장
- 페이지의 LSN은 페이지 flush 작업 이전에 flush 되어야 하는 redo log file을 알려줌

LSN	Description
log_sys -> lsn	다음에 생성될 redo log record가 사용할 lsn
log_sys -> flushed_to_disk_lsn	해당 lsn까지 flush 작업이 수행 lsn이 flushed_to_disk_lsn보다 작으면 모든 redo log record는 디스크에 안전하게 저장
log_sys -> write_lsn	현재 쓰기 작업은 해당 lsn까지 수행
log_sys -> current_flush_lsn	현재 쓰기 작업과 flush 작업은 해당 lsn까지 수행



InnoDB 특징

- 1. 역사
- 2. 아키텍처
- 3. 메모리
- 4. 디스크
- 5. 성능

4.1 Redo Log File

4.2 Logical Structure

4.2.1 Tablespace

4.2.1.1 IBD File의 구조

4.2.2 Segment, Extent

4.2.3 Page

4.2.3.1 Page Layout

4.2.3.2 Directory Slot

4.2.3.3 Page 내부 동작원리

4.3 Undo, MVCC Architecture

4.4 Purge

4.4.1 Purge 정의 및 필요성

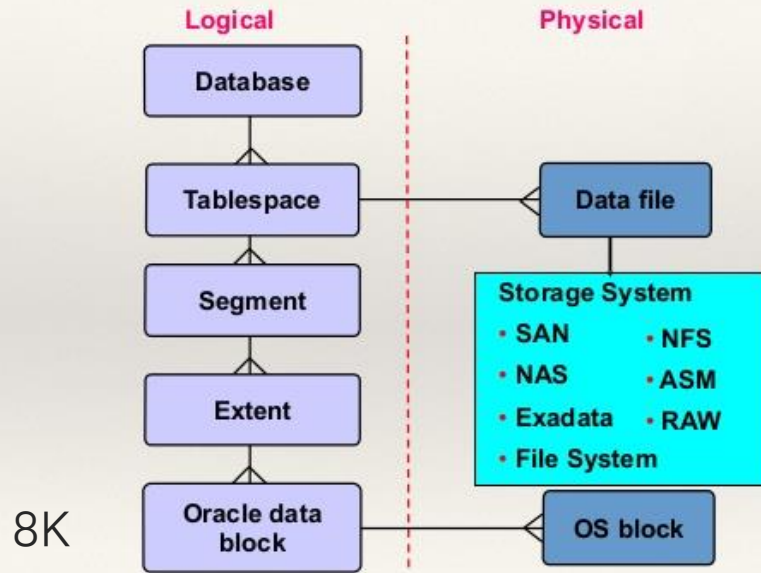
4.4.2 Purge Update Undo

4.4.3 Purge Garbage Space

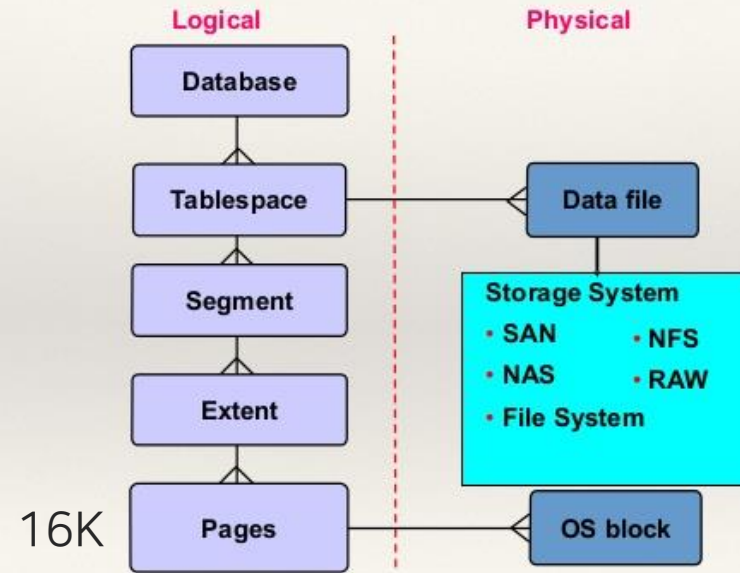


Logical Structure

Logical and Physical Database Structures in Oracle

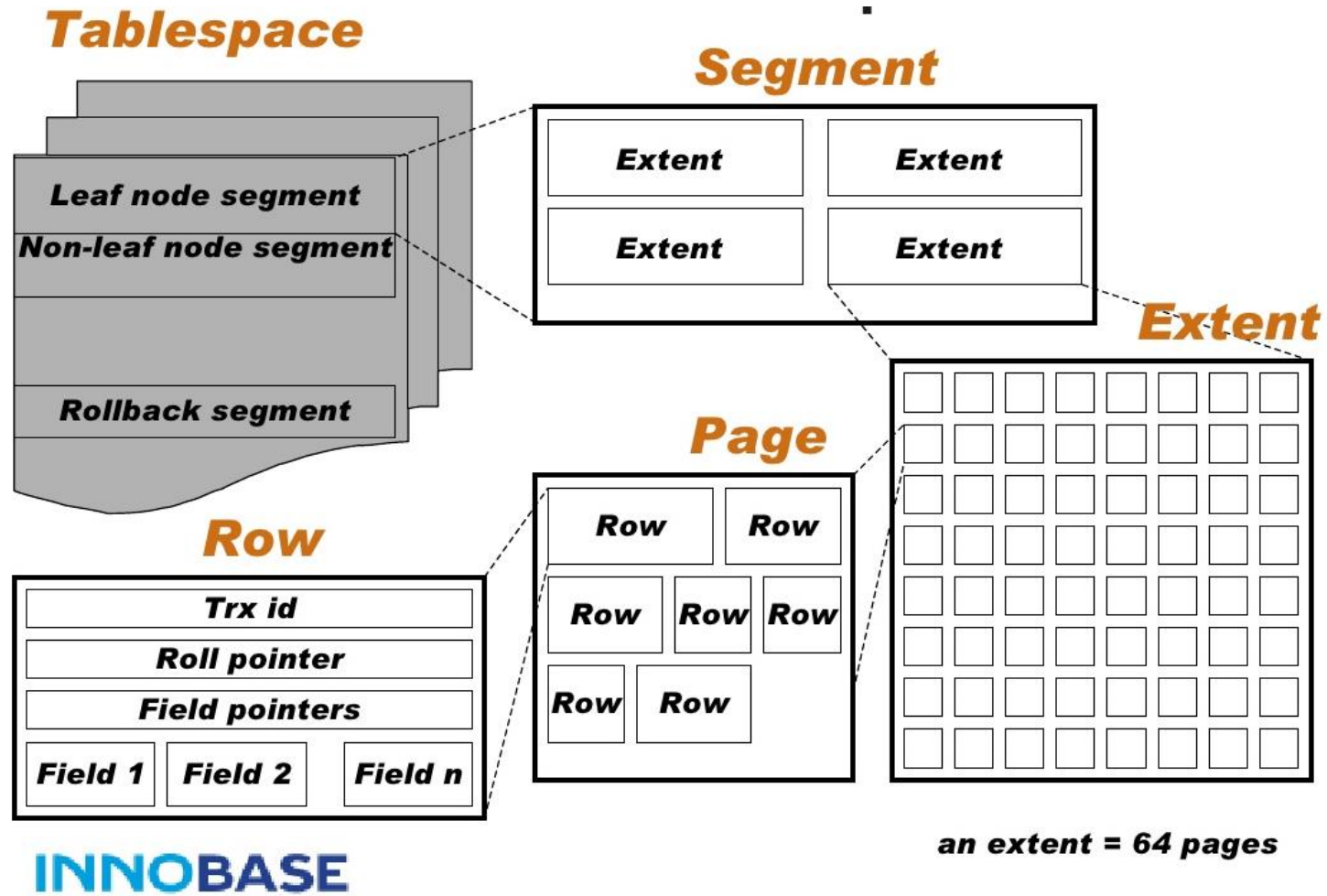


Logical and Physical Database Structures in InnoDB





Logical Structure





InnoDB 특징

- 1. 역사
- 2. 아키텍처
- 3. 메모리
- 4. 디스크
- 5. 성능

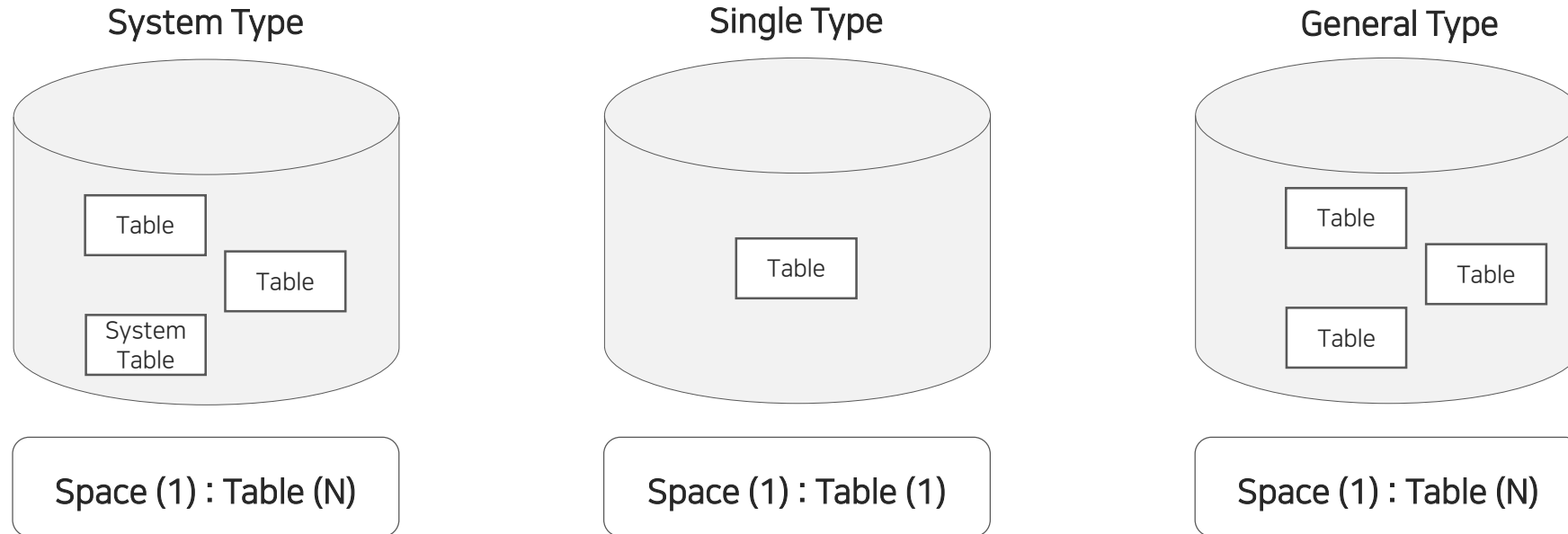
- 4.1 Redo Log File
- 4.2 Logical Structure
 - 4.2.1 Tablespace
 - 4.2.1.1 IBD File의 구조
 - 4.2.2 Segment, Extent
 - 4.2.3 Page
 - 4.2.3.1 Page Layout
 - 4.2.3.2 Directory Slot
 - 4.2.3.3 Page 내부 동작원리
- 4.3 Undo, MVCC Architecture
- 4.4 Purge
 - 4.4.1 Purge 정의 및 필요성
 - 4.4.2 Purge Update Undo
 - 4.4.3 Purge Garbage Space



Tablespace (InnoDB)

InnoDB의 Tablespace는 용도와 생성 방법에 따라 3가지로 분류

- 1) System Type (ibdata1) : InnoDB의 시스템 정보와 일반 테이블들을 저장. Tablespace ID는 0으로 고정, 테이블(N)과 1:N 관계
- 2) Single Type (Table_name.ibd) : 'InnoDB_file_per_table' 시스템 변수가 ON으로 설정된 후 생성된 테이블의 Tablespace, 테이블과 1:1 관계
Amazon Aurora: innodb_file_per_table (default ON, modifiable TRUE)
- 3) General Type (Tablespace_name.ibd) : 'CREATE TABLESPACE ...'를 이용해서 수동으로 생성, 테이블(N)과 1:N 관계 (Aurora 생성 권한 X)





General Tablespace

임의로 Tablespace 생성

CREATE TABLESPACE ... 를 통해 Tablespace를 생성한 경우, Type은 General이며 파일명은 구문에서 명시한대로 생성

MySQL 5.7

```
mysql> create tablespace exem add datafile 'good.ibd';
Query OK, 0 rows affected (0.01 sec)

mysql> select name as table_name, space_type, file_name from innodb_sys_tablespaces t, files f where name='exem' and t.space=f.file_id;
+-----+-----+-----+
| table_name | space_type | file_name |
+-----+-----+-----+
| exem      | General   | ./good.ibd |
+-----+-----+-----+
1 row in set (0.00 sec)
```

Amazon Aurora: create tablespace 권한 없음

Aurora

```
create tablespace ts_user01 add datafile 'ts_user01_001.ibd' ;
ERROR 1227 (42000): Access denied; you need (at least one of) the CREATE TABLESPACE privilege(s) for this
operation
```



Single Tablespace

Database와 Table 생성 (innodb_file_per_table = ON)

```
E1> create database exem_i;
Query OK, 1 row affected (0.02 sec)

E1> use exem_i;
Database changed

E1> create table t1 ( c1 char(14), c2 char(2), c3 char(6), c4 char(3), primary key(c1)) engine=innodb row_format=COMPACT;
Query OK, 0 rows affected (0.04 sec)

E1> select * from information_schema.INNODB_SYS_TABLES where name like 'exem%' ;
+-----+-----+-----+-----+-----+-----+-----+-----+
| TABLE_ID | NAME          | FLAG | N_COLS | SPACE | FILE_FORMAT | ROW_FORMAT | ZIP_PAGE_SIZE |
+-----+-----+-----+-----+-----+-----+-----+-----+
| 95 | exem_i/T2 | 1 | 5 | 37 | Antelope | Compact | 0 |
| 85 | exem_i/t1 | 1 | 7 | 27 | Antelope | Compact | 0 |
| 99 | exem_i/t3 | 1 | 4 | 41 | Antelope | Compact | 0 |
| 98 | exem_i/t4 | 1 | 4 | 40 | Antelope | Compact | 0 |
| 100 | exem_i/t5 | 1 | 4 | 42 | Antelope | Compact | 0 |
+-----+-----+-----+-----+-----+-----+-----+-----+
5 rows in set (0.01 sec)
```

- innodb_file_per_table 시스템 변수
ON: 각 테이블마다 고유의 Tablespace 생성
OFF: System Tablespace(ibdata1) 에 모든 테이블 할당



Single Tablespace

해당 Table의 Tablespace의 번호와 File ID, Datafile 확인

MySQL 5.7

```
mysql> select name,space,space_type from information_schema.innodb_sys_tables where name='exem_i/t1';
+-----+-----+-----+
| name      | space | space_type |
+-----+-----+-----+
| exem_i/t1 | 20    | Single     |
+-----+-----+-----+
1 row in set (0.00 sec)
```

5.7부터 space_type이 보임

```
mysql> select file_id, file_name, file_type, tablespace_name
-> from information_schema.files where file_id=20;
+-----+-----+-----+-----+
| FILE_ID | FILE_NAME          | FILE_TYPE | TABLESPACE_NAME |
+-----+-----+-----+-----+
| 20      | ./exem_i/t1.ibd   | TABLESPACE | innodb_file_per_table_20 |
+-----+-----+-----+-----+
1 row in set (0.00 sec)
```

space: tablespace 번호

space_type: single

FILE_ID: 파일 번호 (tablespace 번호와 동일)

FILE_NAME: [테이블 이름].ibd

TABLESPACE_NAME: innodb_file_per_table_N

cf) Amazon Aurora: information_schema.files가 없음 -> 권한이 없음



Single Tablespace

Aurora

```
E1> select * from information_schema.INNODB_SYS_TABLES where name like 'exem%' ;
+-----+-----+-----+-----+-----+-----+-----+-----+
| TABLE_ID | NAME          | FLAG | N_COLS | SPACE | FILE_FORMAT | ROW_FORMAT | ZIP_PAGE_SIZE |
+-----+-----+-----+-----+-----+-----+-----+-----+
| 95 | exem_i/T2 | 1 | 5 | 37 | Antelope | Compact | 0 |
| 85 | exem_i/t1 | 1 | 7 | 27 | Antelope | Compact | 0 |
| 99 | exem_i/t3 | 1 | 4 | 41 | Antelope | Compact | 0 |
| 98 | exem_i/t4 | 1 | 4 | 40 | Antelope | Compact | 0 |
| 100 | exem_i/t5 | 1 | 4 | 42 | Antelope | Compact | 0 |
+-----+-----+-----+-----+-----+-----+-----+-----+
5 rows in set (0.01 sec)
```

```
E1> select * from information_schema.INNODB_SYS_DATAFILES where space = 37;
+-----+-----+
| SPACE | PATH          |
+-----+-----+
| 37 | ./exem_i/T2.ibd |
+-----+-----+
```



참고 - 실습1

해당 Table을 System tablespace로 이동

MySQL 5.7

```
mysql> ALTER TABLE t1 TABLESPACE = innodb_system;
Query OK, 0 rows affected (0.02 sec)
Records: 0 Duplicates: 0 Warnings: 0

mysql> select name,space,space_type from information_schema.innodb_sys_tables where name='exem_i/t1';
+-----+-----+-----+
| name      | space | space_type |
+-----+-----+-----+
| exem_i/t1 | 0     | System     |
+-----+-----+-----+
1 row in set (0.00 sec)
```

Space 컬럼의 값(Tablespace ID)이 0일 경우 Tablespace의 종류는 System(ibdata1)



참고 - 실습1

Aurora

```
E1> alter table exem_i.T2 tablespace innodb_system ;
Query OK, 0 rows affected (0.10 sec)
Records: 0 Duplicates: 0 Warnings: 0
```

```
E1> select * from information_schema.INNOODB_SYS_DATAFILES where path like '%exem%' ;
```

datafile에는 변경없이 존재

```
+-----+-----+
| SPACE | PATH                |
+-----+-----+
| 27    | ./exem_i/t1.ibd    |
| 37    | ./exem_i/T2.ibd    |
| 40    | ./exem_i/t4.ibd    |
| 41    | ./exem_i/t3.ibd    |
| 42    | ./exem_i/t5.ibd    |
+-----+-----+
5 rows in set (0.00 sec)
```

```
E1> select * from information_schema.INNOODB_SYS_TABLES where name like 'exem%' ;
```

5.6 에서는 space_type 확인 불가

```
+-----+-----+-----+-----+-----+-----+-----+-----+
| TABLE_ID | NAME          | FLAG | N_COLS | SPACE | FILE_FORMAT | ROW_FORMAT | ZIP_PAGE_SIZE |
+-----+-----+-----+-----+-----+-----+-----+-----+
| 95        | exem_i/T2    | 1    | 5      | 37    | Antelope    | Compact    | 0              |
| 85        | exem_i/t1    | 1    | 7      | 27    | Antelope    | Compact    | 0              |
| 99        | exem_i/t3    | 1    | 4      | 41    | Antelope    | Compact    | 0              |
| 98        | exem_i/t4    | 1    | 4      | 40    | Antelope    | Compact    | 0              |
| 100       | exem_i/t5    | 1    | 4      | 42    | Antelope    | Compact    | 0              |
+-----+-----+-----+-----+-----+-----+-----+-----+
5 rows in set (0.00 sec)
```

명령어 실행 여부에 대해서 확인 불가. datafile에 존재하는 것으로 보아 옮겨지지 않는 것으로 보임.



참고 - 실습2

해당 Table의 Tablespace의 번호와 File ID, Datafile 확인

MySQL 5.7

```
mysql> select FILE_ID, FILE_NAME, FILE_TYPE, TABLESPACE_NAME from information_schema.files;
+-----+-----+-----+-----+
| FILE_ID | FILE_NAME                | FILE_TYPE | TABLESPACE_NAME |
+-----+-----+-----+-----+
|      0 | /app/mysql/ibdata/ibdata1 | TABLESPACE | innodb_system    |
|      0 | /app/mysql/ibdata/ibdata2 | TABLESPACE | innodb_system    |
|     18 | /app/mysql/ibdata/ibtmp1  | TEMPORARY  | innodb_temporary |
+-----+-----+-----+-----+
3 rows in set (0.00 sec)

mysql> \! ls /app/mysql/data/exem_i/
db.opt  t1.frm
```

해당 Table의 Tablespace를 변경할 경우 기존 Tablespace(FILE_ID 20)와 .ibd File은 사라짐



참고 - 실습3

해당 Table을 Innodb_file_per_table로 이동

MySQL 5.7

```
mysql> alter table t1 tablespace = innodb_file_per_table;
Query OK, 0 rows affected (0.04 sec)
Records: 0 Duplicates: 0 Warnings: 0

mysql> select name,space,space_type from information_schema.innodb_sys_tables where name = 'exem_i/t1';
+-----+-----+-----+
| name      | space | space_type |
+-----+-----+-----+
| exem_i/t1 | 24    | Single     |
+-----+-----+-----+
1 row in set (0.00 sec)
```



참고 - 실습4

해당 Table의 Tablespace의 번호와 File ID, Datafile 확인

MySQL 5.7

```
mysql> select file_id, file_name, file_type, tablespace_name from information_schema.files where file_id=24;
+-----+-----+-----+-----+
| file_id | file_name          | file_type | tablespace_name          |
+-----+-----+-----+-----+
|      24 | ./exem_i/t1.ibd   | TABLESPACE | innodb_file_per_table_24 |
+-----+-----+-----+-----+
1 row in set (0.00 sec)

mysql> \! ls /app/mysql/data/exem_i/
db.opt _t1.frm t1.ibd
```

해당 Table의 Tablespace는 기존 방식대로 innodb_file_per_table_N 으로 변경되고 새로운 .ibd File 생성

Aurora의 경우 접근 불가로 확인 불가



InnoDB 특징

- 1. 역사
- 2. 아키텍처
- 3. 메모리
- 4. 디스크
- 5. 성능

- 4.1 Redo Log File
- 4.2 Logical Structure
 - 4.2.1 Tablespace
 - 4.2.1.1 IBD File의 구조
 - 4.2.2 Segment, Extent
 - 4.2.3 Page
 - 4.2.3.1 Page Layout
 - 4.2.3.2 Directory Slot
 - 4.2.3.3 Page 내부 동작원리
- 4.3 Undo, MVCC Architecture
- 4.4 Purge
 - 4.4.1 Purge 정의 및 필요성
 - 4.4.2 Purge Update Undo
 - 4.4.3 Purge Garbage Space



IBD File의 구조

t1 테이블의 IBD 파일의 구조

1. innodb_file_per_table = ON

2. t1 테이블 생성

```
mysql> create table exem_i.t1 ( c1 char(14), c2 char(11), c3 text(4000), primary key(c1) ) engine=innodb row_format=COMPACT
```

Query OK, 0 rows affected (0.03 sec)

3. 메모리에 캐시된 IBD 파일 구조 조회

```
mysql> select space,page_number,page_type,table_name,index_name from information_schema.INNODB_BUFFER_PAGE_LRU where space =462;
```

space	page_number	page_type	table_name	index_name
462	0	FILE_SPACE_HEADER	NULL	NULL
462	1	IBUF_BITMAP	NULL	NULL
462	2	INODE	NULL	NULL
462	3	INDEX	`exem_i`.`t1`	PRIMARY

4 rows in set (0.00 sec)

Page(462,0) FSP_HDR	Page(462,1) IBUF_BITMAP	Page(462,2) INODE	Page(462,3) INDEX (PRIMARY)	Page(462,4) ALLOCATED (Reserved)	Page(462,5) ALLOCATED (Reserved)
------------------------	----------------------------	----------------------	-----------------------------------	--	--

0번 페이지 : File space Header

1번 페이지 : Insert Buffer Bitmap

2번 페이지 : INODE

3번 페이지 : Primary Key인 c1 컬럼의 인덱스 페이지

4번 페이지 : Reserved 페이지 (미리 할당해두고 필요할 때 사용)

5번 페이지 : Reserved 페이지 (미리 할당해두고 필요할 때 사용)



IBD File의 구조

테이블 생성시 공간할당 정보

t1 테이블의 .ibd 파일은 총 100K 할당됨

1) INNODB_SYS_TABLESPACES 디렉터리 테이블

```
mysql> select space, name, allocated_size, file_size, page_size, fs_block_size
from INNODB_SYS_TABLESPACES where name = 'exem_i/t1';
```

SPACE	NAME	ALLOCATED_SIZE	FILE_SIZE	PAGE_SIZE	FS_BLOCK_SIZE
462	exem_i/t1	102400	98304	16384	4096

1 row in set (0.00 sec)

- ALLOCATED_SIZE : 해당 파일에 할당된 총 사이즈
- FILE_SIZE : 파일 시스템 블록(4k)를 제외한 DBMS 블록의 사이즈
- FS_BLOCK_SIZE : Hole Punching을 위한 파일 시스템 블록의 사이즈

2) OS단의 du명령어

```
mysql> W! du -h /app/mysql/data/5.7/exem_i/t1.ibd
```

```
100K /app/mysql/data/5.7/exem_i/t1.ibd
```

t1.ibd (Space ID : 462) 100K (+4K)

Page 16K	Page 16K	Page 16K	Page 16K	Page 16K	Page 16K
----------	----------	----------	----------	----------	----------

Page(462,0) FSP_HDR	Page(462,1) IBUF_BITMAP	Page(462,2) INODE	Page(462,3) INDEX (PRIMARY)	Page(462,4) ALLOCATED (Reserved)	Page(462,5) ALLOCATED (Reserved)
------------------------	----------------------------	----------------------	-----------------------------------	--	--



IBD File의 구조

Secondary Index 생성시 공간할당

1. c2 컬럼에 Secondary Index를 생성

```
mysql> create index t1_idx on exem_i.t1(c2);
```

Query OK, 0 rows affected (0.03 sec)

Records: 0 Duplicates: 0 Warnings: 0

2. 메모리에 캐시된 IBD 파일 구조 조회

```
mysql> select space,page_number,page_type,table_name,index_name from
      INNODB_BUFFER_PAGE_LRU where space =462;
```

space	page_number	page_type	table_name	index_name
462	0	FILE_SPACE_HEADER	NULL	NULL
462	1	IBUF_BITMAP	NULL	NULL
462	2	INODE	NULL	NULL
462	3	INDEX	`exem_i`.`t1`	PRIMARY
462	4	INDEX	`exem_i`.`t1`	t1_idx

5 rows in set (0.00 sec)

t1.ibd (Space ID : 462) 116K (+4K)

Page 16K	Page 16K	Page 16K	Page 16K	Page 16K	Page 16K	Page 16K
----------	----------	----------	----------	----------	----------	----------

Page(462,0) FSP_HDR	Page(462,1) IBUF_BITMA P	Page(462,2) INODE	Page(462,3) INDEX (PRIMARY)	Page(462,4) INDEX (t1_idx)	Page(462,5) ALLOCATED (Reserved)	Page(462,6) ALLOCATED (Reserved)
------------------------	--------------------------------	----------------------	-----------------------------------	---	--	--



IBD File의 구조

데이터 삽입 시 공간할당 (PK Index)

1. 327건 데이터 삽입 (FULL)

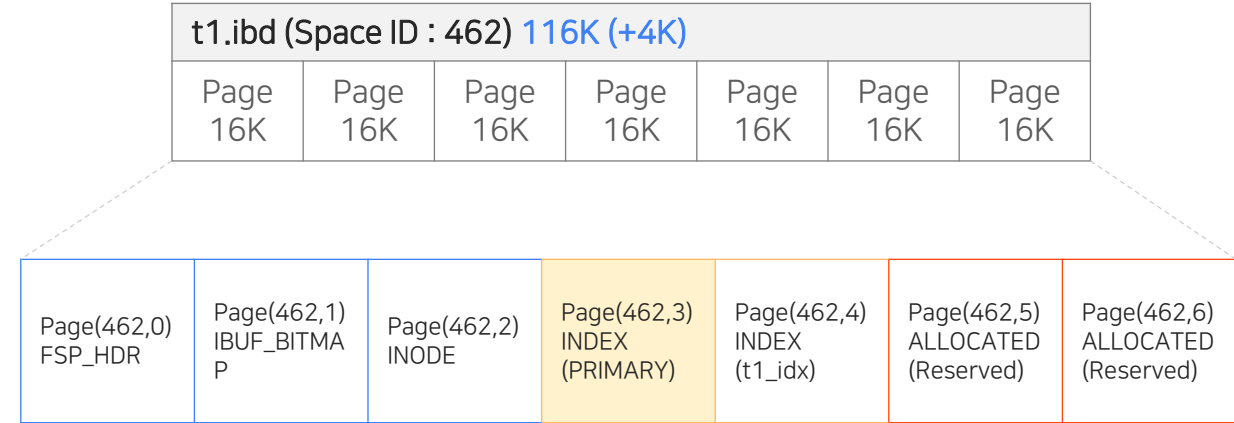
```
mysql> INSERT 327 Rows.. Rec0001 ~ Rec0327
(count 327 rows)
```

2. 메모리에 캐시된 IBD 파일 구조 조회

```
mysql> select space,page_number,page_type,table_name,index_name from
INNODB_BUFFER_PAGE_LRU where space =462;
```

space	page_number	page_type	table_name	index_name
462	0	FILE_SPACE_HEADER	NULL	NULL
462	1	IBUF_BITMAP	NULL	NULL
462	2	INODE	NULL	NULL
462	3	INDEX	`exem_i`.`t1`	PRIMARY
462	4	INDEX	`exem_i`.`t1`	t1_idx

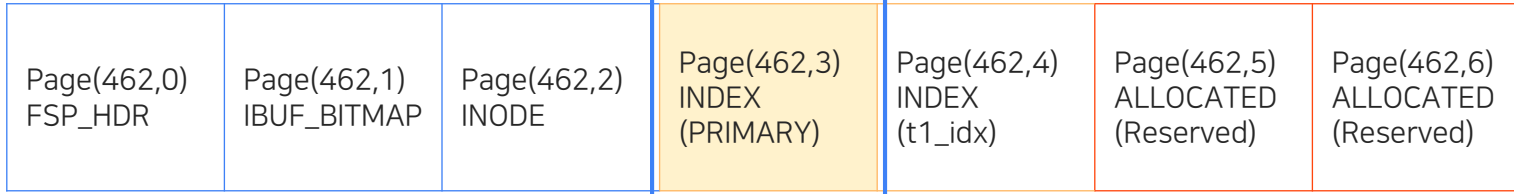
5 rows in set (0.00 sec)





IBD File의 구조

3. 327건 Insert 후, PK INDEX 페이지 (462,3) 덤프



Page (462,3)

10	67	38	e9	00	00	00	03	ff	ff	ff	ff	ff	ff	ff	ff	>.g8.....<
00	00	00	00	10	3a	f3	06	45	bf	00	00	00	00	00	00	>.....:..E.....<
00	00	00	00	01	ce	00	52	3b	3a	81	49	00	00	00	00	>.....R;:..I....<
3b	14	00	02	01	46	01	47	00	00	00	00	00	00	00	00	>;....F.G.....<
00	00	00	00	00	00	00	00	01	ed	00	00	01	ce	00	00	>.....<
00	02	00	f2	00	00	01	ce	00	00	00	02	00	32	01	00	>.....2..<
02	00	1d	69	6e	66	69	6d	75	6d	00	08	00	0b	00	00	>...infimum.....<
73	75	70	72	65	6d	75	6d	0b	0e	02	00	00	10	00	2e	>supremum.....<
52	65	63	30	30	30	30	30	31	20	20	20	20	20	20	00	>Rec000001 ..<
00	02	93	96	a1	00	00	00	04	01	10	56	20	20	20	20	>.....V <
20	20	20	20	20	20	0b	0e	02	00	00	18	00	2e	52	65	>Re<
63	30	30	30	30	30	32	20	20	20	20	20	00	00	00	02	>c000002<
93	97	a1	00	00	00	04	01									> <

Elements	Size	Value
FIL_PAGE_OFFSET	4	0x3
PAGE_N_DIR_SLOTS	2	0x52 (82)
PAGE_N_RECS	2	0x147(327)
PAGE_LEVEL	2	0x0



IBD File의 구조

4. 1건 데이터 삽입

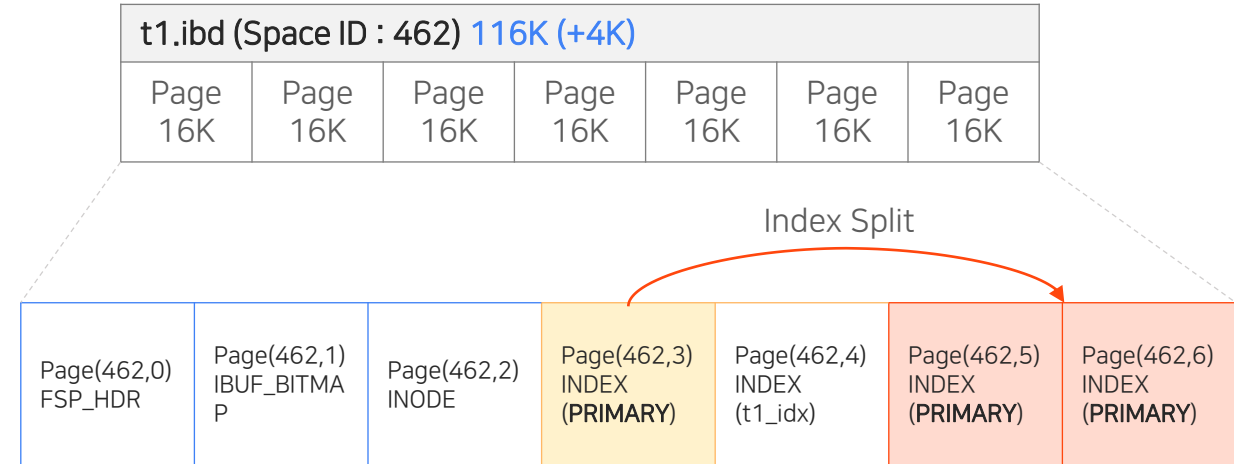
```
mysql> INSERT 1 Rows.. Rec0328
(count 328 rows)
```

5. 메모리에 캐시된 IBD 파일 구조 조회

```
mysql> select space,page_number,page_type,table_name,index_name
from information_schema.INNODB_BUFFER_PAGE_LRU
where space =462;
```

space	page_number	page_type	table_name	index_name
462	0	FILE_SPACE_HEADER	NULL	NULL
462	1	IBUF_BITMAP	NULL	NULL
462	2	INODE	NULL	NULL
462	3	INDEX	`exem_j`.`t1`	PRIMARY
462	4	INDEX	`exem_j`.`t1`	t1_idx
462	5	INDEX	`exem_j`.`t1`	PRIMARY
462	6	INDEX	`exem_j`.`t1`	PRIMARY

7 rows in set (0.01 sec)



3번 페이지 : B-tree 구조의 root node

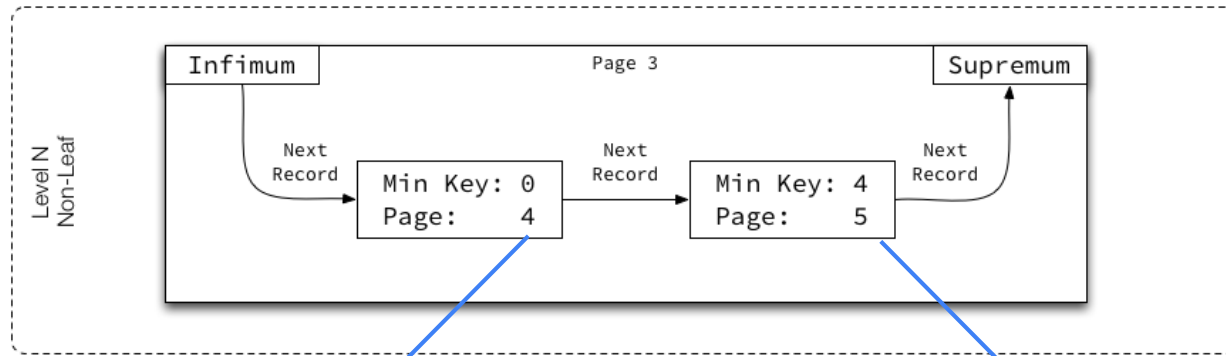
5번 페이지 : B-tree 구조의 leaf node

6번 페이지 : B-tree 구조의 leaf node

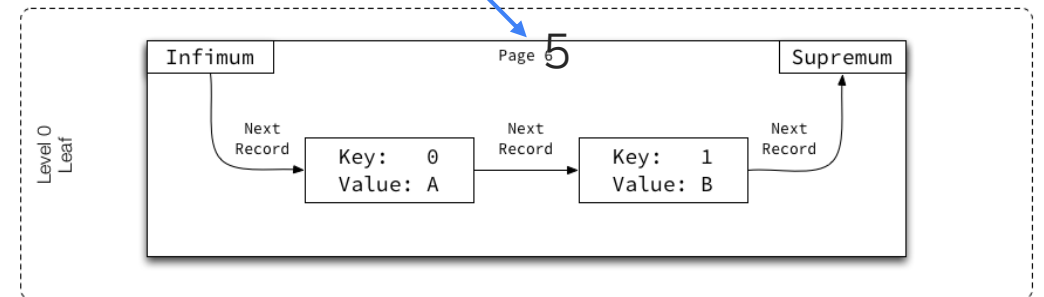
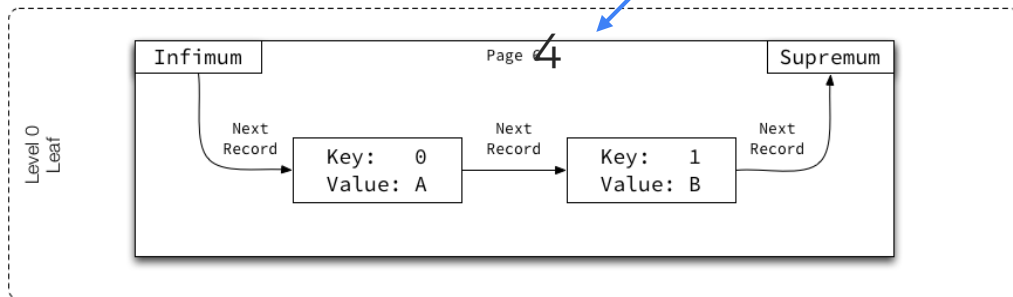


IBD File의 구조

B+Tree Simplified Non-Leaf Page



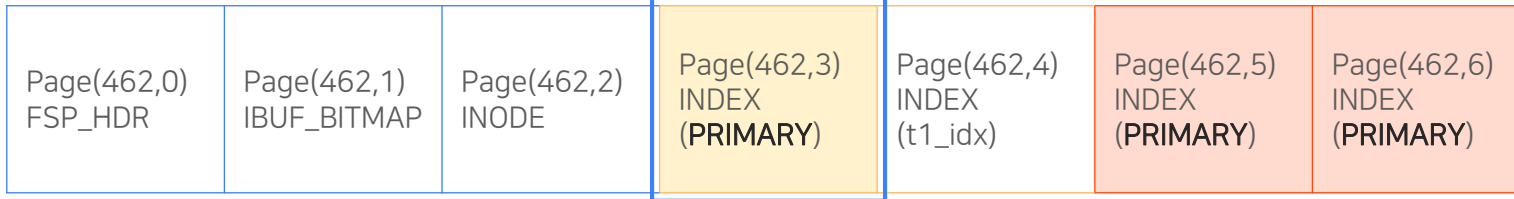
B+Tree Simplified Leaf Page





IBD File의 구조

6. 1건 Insert 후, PK INDEX 페이지 (462,3) 덤프



Page (462,3)

8b	8a	12	25	00	00	00	03	ff	ff	ff	ff	ff	ff	ff	ff	>...%.....<
00	00	00	00	10	3b	37	68	45	bf	00	00	00	00	00	00	>.....;7hE.....<
00	00	00	00	01	ce	00	02	00	aa	80	04	00	00	00	00	>.....<
00	98	00	02	00	01	00	02	00	00	00	00	00	00	00	00	>.....<
00	01	00	00	00	00	00	00	01	ed	00	00	01	ce	00	00	>.....<
00	02	00	f2	00	00	01	ce	00	00	00	02	00	32	01	00	>.....2..<
02	00	1c	69	6e	66	69	6d	75	6d	00	03	00	0b	00	00	>...infimum.....<
73	75	70	72	65	6d	75	6d	0e	00	10	00	11	00	19	52	>supremum.....R<
65	63	30	30	30	30	30	31	20	20	20	20	20	00	00	00	>ec000001 ...<
05	0e	00	00	00	19	ff	d8	52	65	63	30	31	36	34	20	>.....Rec0164 <
20	20	20	20	20	20	00	00	00	06	00	00	00	00	00	00	>.....<

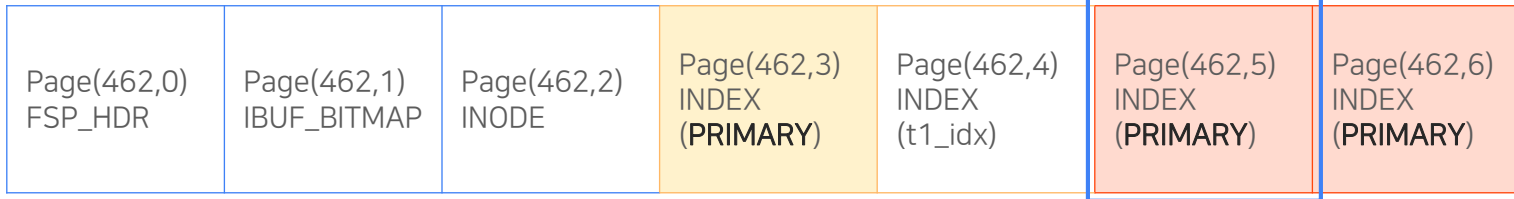
Elements	Size	Value
FIL_PAGE_OFFSET	4	0x3
PAGE_N_DIR_SLOTS	2	0x2
PAGE_N_RECS	2	0x2
PAGE_LEVEL	2	0x1

Key	Node Pointer
Rec000001	0x5
Rec0164	0x6



IBD File의 구조

7. 1건 Insert 후, PK INDEX 페이지 (462,5) 덤프



Page (462,5)

7a	4d	8f	0e	00	00	00	05	ff	ff	ff	ff	00	00	00	06	> z M.....<
00	00	00	00	10	3b	37	68	45	bf	00	00	00	00	00	00	>.....;7 hE.....<
00	00	00	00	01	ce	00	2a	3b	3a	81	49	1d	ca	1d	78	>.....*;;.l...x<
00	00	00	05	00	00	00	a3	00	00	00	00	00	00	00	00	>.....<
00	00	00	00	00	00	00	00	01	ed	00	00	00	00	00	00	>.....<
00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	01	>.....<
02	00	1d	69	6e	66	69	6d	75	6d	00	04	00	0b	00	00	>...infimum.....<
73	75	70	72	65	6d	75	6d	0b	0e	02	00	00	10	00	2e	>supremum.....<

Elements	Size	Value
FIL_PAGE_OFFSET	4	0x5
FIL_PAGE_PREV	4	0xffffffff
FIL_PAGE_NEXT	4	0x6
PAGE_N_DIR_SLOTS	2	0x2a (42)
PAGE_N_RECS	2	0xa3 (163)
PAGE_LEVEL	2	0x0



IBD File의 구조

8. 1건 Insert 후, PK INDEX 페이지 (462,6) 덤프

Page(462,0) FSP_HDR	Page(462,1) IBUF_BITMAP	Page(462,2) INODE	Page(462,3) INDEX (PRIMARY)	Page(462,4) INDEX (t1_idx)	Page(462,5) INDEX (PRIMARY)	Page(462,6) INDEX (PRIMARY)
------------------------	----------------------------	----------------------	-----------------------------------	----------------------------------	-----------------------------------	-----------------------------------

Page (462,6)

42	61	69	18	00	00	00	06	00	00	00	05	ff	ff	ff	ff	> Bai.....<
00	00	00	00	10	3b	37	68	45	bf	00	00	00	00	00	00	>.....;7hE.....<
00	00	00	00	01	ce	00	2a	1e	1e	80	a7	00	00	00	00	>.....*.....<
1d	f8	00	05	00	00	00	a5	00	00	00	00	00	00	00	00	>.....<
00	00	00	00	00	00	00	00	01	ed	00	00	00	00	00	00	>.....<
00	00	00	00	00	00	00	00	00	00	00	00	00	00	01	00	>.....<
02	00	1d	69	6e	66	69	6d	75	6d	00	06	00	0b	00	00	>...infimum.....<
73	75	70	72	65	6d	75	6d	0b	0e	02	00	00	10	00	2e	>supremum.....<

Elements	Size	Value
FIL_PAGE_OFFSET	4	0x6
FIL_PAGE_PREV	4	0x5
FIL_PAGE_NEXT	4	0xffffffff
PAGE_N_DIR_SLOTS	2	0x2a (42)
PAGE_N_RECS	2	0xa5 (165)
PAGE_LEVEL	2	0x0



IBD File의 구조

데이터 삽입 시 공간할당 (Secondary Index)

1. 157건 데이터 삽입 (FULL)

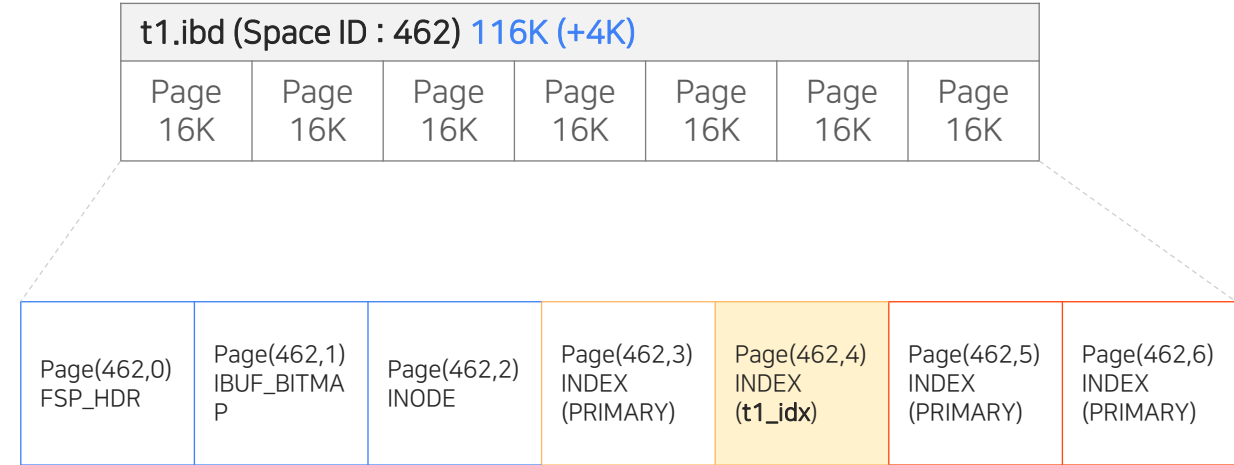
```
mysql> INSERT 157 Rows..
(count 485 rows)
```

2. 메모리에 캐시된 IBD 파일 구조 조회

```
mysql> select space,page_number,page_type,table_name,index_name
from information_schema.INNODB_BUFFER_PAGE_LRU
where space =462;
```

space	page_number	page_type	table_name	index_name
462	0	FILE_SPACE_HEADER	NULL	NULL
462	1	IBUF_BITMAP	NULL	NULL
462	2	INODE	NULL	NULL
462	3	INDEX	`exem_j`.`t1`	PRIMARY
462	4	INDEX	`exem_j`.`t1`	t1_idx
462	5	INDEX	`exem_j`.`t1`	PRIMARY
462	6	INDEX	`exem_j`.`t1`	PRIMARY

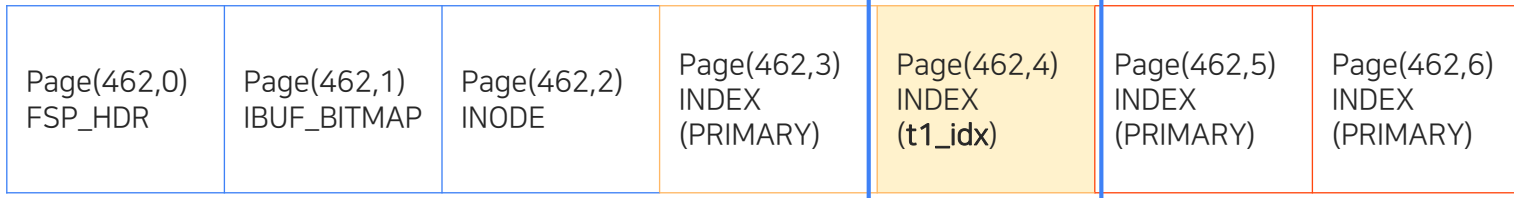
7 rows in set (0.00 sec)





IBD File의 구조

3. 1건 Insert 후, Secondary INDEX 페이지 (462,4) 덤프



Page (462,4)

f0	db	23	df	00	00	00	04	ff	ff	ff	ff	ff	ff	ff	ff	>..#.....<
00	00	00	00	10	3c	05	08	45	bf	00	00	00	00	00	00	>.....<..E.....<
00	00	00	00	01	ce	00	6a	3e	fd	81	e7	00	00	00	00	>.....j>.....<
3e	e4	00	05	00	00	01	e5	00	00	00	00	00	02	95	82	>>.....<
00	00	00	00	00	00	00	00	01	ee	00	00	01	ce	00	00	>.....<
00	02	02	72	00	00	01	ce	00	00	00	02	01	b2	01	00	>...r.....<
02	09	02	69	6e	66	69	6d	75	6d	00	07	00	0b	00	00	>...infimum.....<
73	75	70	72	65	6d	75	6d	0e	0b	00	00	00	10	01	ce	>supremum.....<

Elements	Size	Value
FIL_PAGE_OFFSET	4	0x4
PAGE_N_DIR_SLOTS	2	0x6a (106)
PAGE_N_RECS	2	0x1e5 (485)
PAGE_LEVEL	2	0x0



IBD File의 구조

4. 1건 데이터 삽입

mysql> INSERT 1 Rows..
(count 486 rows)

5. 메모리에 캐시된 IBD 파일 구조 조회

mysql> select space,page_number,page_type,table_name,index_name
from information_schema.INNODB_BUFFER_PAGE_LRU
where space =462;

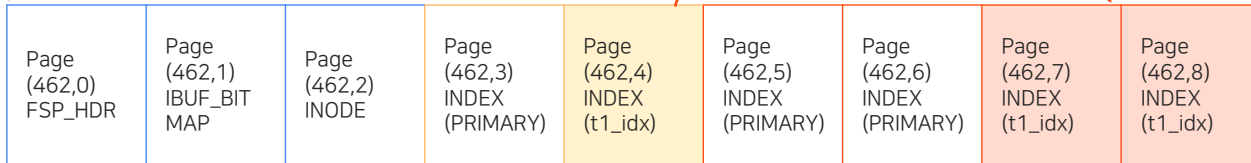
space	page_number	page_type	table_name	index_name
462	0	FILE_SPACE_HEADER	NULL	NULL
462	1	IBUF_BITMAP	NULL	NULL
462	2	INODE	NULL	NULL
462	3	INDEX	`exem_i`.`t1`	PRIMARY
462	5	INDEX	`exem_i`.`t1`	PRIMARY
462	6	INDEX	`exem_i`.`t1`	PRIMARY
462	4	INDEX	`exem_i`.`t1`	t1_idx
462	7	INDEX	`exem_i`.`t1`	t1_idx
462	8	INDEX	`exem_i`.`t1`	t1_idx

9 rows in set (0.02 sec)

t1.ibd (Space ID : 462) 148K (+4K)

Page 16K	Page 16K	Page 16K	Page 16K	Page 16K	Page 16K	Page 16K	Page 16K	Page 16K
----------	----------	----------	----------	----------	----------	----------	----------	----------

Index Split



4번 페이지 : B-tree 구조의 root node

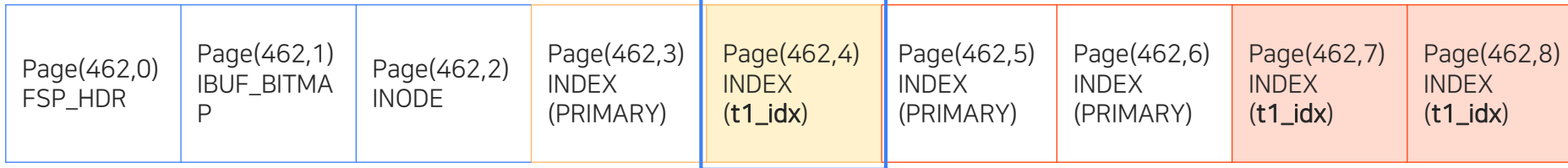
7번 페이지 : B-tree 구조의 leaf node

8번 페이지 : B-tree 구조의 leaf node



IBD File의 구조

6. 1건 Insert 후, Secondary INDEX 페이지 (462,4) 덤프



Page (462,4)

41	5d	24	ad	00	00	00	04	ff	ff	ff	ff	ff	ff	ff	ff	>A]\$.....<
00	00	00	00	10	3c	31	a4	45	bf	00	00	00	00	00	00	>.....<1.E.....<
00	00	00	00	01	ce	00	02	00	c2	80	04	00	00	00	00	>.....<
00	a5	00	02	00	01	00	02	00	00	00	00	00	00	00	00	>.....<
00	01	00	00	00	00	00	00	01	ee	00	00	01	ce	00	00	>.....<
00	02	02	72	00	00	01	ce	00	00	00	02	01	b2	01	00	>...r.....<
02	00	1d	69	6e	66	69	6d	75	6d	00	03	00	0b	00	00	>...infimum.....<
73	75	70	72	65	6d	75	6d	0e	0b	00	10	00	11	00	25	>supremum.....%<
41	20	20	20	20	20	20	20	20	20	20	52	65	63	30	30	>A Rec00<
30	37	30	20	20	20	20	20	20	00	00	00	07	0e	0b	00	>070<
00	00	19	ff	cb	4e	20	20	20	20	20	20	20	20	20	20	>....N <
52	65	63	30	31	33	36	20	20	20	20	20	20	20	00	00	>Rec0136 ..<
00	08	00	00	00	00	00	00									>.....<

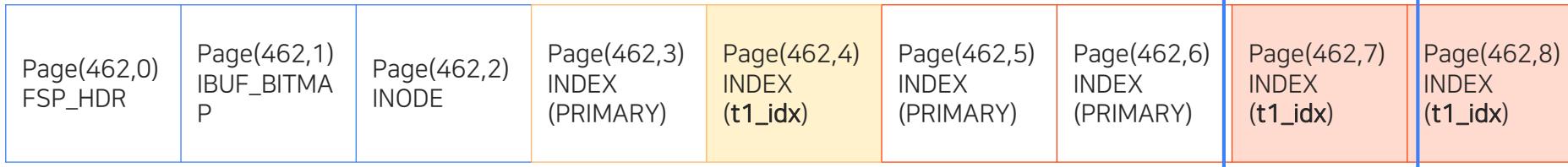
Elements	Size	Value
FIL_PAGE_OFFSET	4	0x4
PAGE_N_DIR_SLOTS	2	0x2
PAGE_N_RECS	2	0x2
PAGE_LEVEL	2	0x1

Key	Node Pointer
A	0x7
N	0x8



IBD File의 구조

7. 1건 Insert 후, Secondary INDEX 페이지 (462,7) 덤프



Page (462,7)

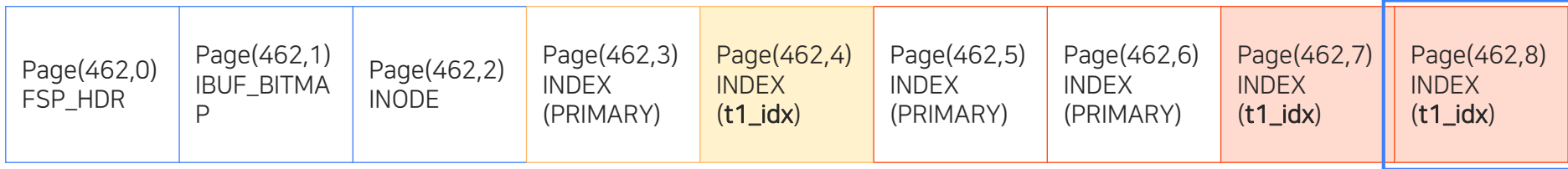
1d	0f	3d	9f	00	00	00	07	ff	ff	ff	ff	00	00	00	08	>..=.....<
00	00	00	00	10	3c	31	a4	45	bf	00	00	00	00	00	00	>.....<1.E.....<
00	00	00	00	01	ce	00	3e	3e	fd	81	e7	1f	b2	1f	53	>.....>>.....S<
00	00	00	05	00	00	00	f2	00	00	00	00	00	02	95	82	>.....<
00	00	00	00	00	00	00	00	01	ee	00	00	00	00	00	00	>.....<
00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	01	>.....<
02	00	1d	69	6e	66	69	6d	75	6d	00	03	00	0b	00	00	>...infimum.....<
73	75	70	72	65	6d	75	6d	0e	0b	00	00	00	10	00	21	>supremum.....!<
41	20	20	20	20	20	20	20	20	20	20	52	65	63	30	30	>A Rec00<
30	37	30	20	20	20	20	20	20	0e	0b	00	00	00	18	00	>070<

Elements	Size	Value
FIL_PAGE_OFFSET	4	0x7
FIL_PAGE_PREV	4	0xffffffff
FIL_PAGE_NEXT	4	0x8
PAGE_N_DIR_SLOTS	2	0x3e (62)
PAGE_N_RECS	2	0xf2 (242)
PAGE_LEVEL	2	0x0



IBD File의 구조

8. 1건 Insert 후, Secondary INDEX 페이지 (462,8) 덤프



Page (462,8)

f5	fa	39	8f	00	00	00	08	00	00	00	07	ff	ff	ff	ff	>...9.....<
00	00	00	00	10	3c	31	a4	45	bf	00	00	00	00	00	00	>.....<1.E.....<
00	00	00	00	01	ce	00	3d	1f	ec	80	f6	00	00	00	00	>.....=.....<
1f	d3	00	05	00	00	00	f4	00	00	00	00	00	02	95	87	>.....<
00	00	00	00	00	00	00	00	01	ee	00	00	00	00	00	00	>.....<
00	00	00	00	00	00	00	00	00	00	00	00	00	00	01	00	>.....<
02	00	1d	69	6e	66	69	6d	75	6d	00	08	00	0b	00	00	>...infimum.....<
73	75	70	72	65	6d	75	6d	0e	0b	00	00	00	10	00	21	>supremum.....!<
4e	20	20	20	20	20	20	20	20	20	20	52	65	63	30	31	>N Rec01<
33	36	20	20	20	20	20	20	20	0e	0b	00	00	00	18	00	>36<

Elements	Size	Value
FIL_PAGE_OFFSET	4	0x8
FIL_PAGE_PREV	4	0x7
FIL_PAGE_NEXT	4	0xffffffff
PAGE_N_DIR_SLOTS	2	0x3d (61)
PAGE_N_RECS	2	0xf4 (244)
PAGE_LEVEL	2	0x0



InnoDB 특징

- 1. 역사
- 2. 아키텍처
- 3. 메모리
- 4. 디스크**
- 5. 성능

- 4.1 Redo Log File
- 4.2 Logical Structure
 - 4.2.1 Tablespace
 - 4.2.1.1 IBD File의 구조
 - 4.2.2 Segment, Extent
 - 4.2.3 Page
 - 4.2.3.1 Page Layout
 - 4.2.3.2 Directory Slot
 - 4.2.3.3 Page 내부 동작원리
- 4.3 Undo, MVCC Architecture
- 4.4 Purge
 - 4.4.1 Purge 정의 및 필요성
 - 4.4.2 Purge Update Undo
 - 4.4.3 Purge Garbage Space



Segment, Extent

File Segments

- ❖ A tablespace can contain many file segments
- ❖ Information of segment header
 - ❖ The space to which the inode belongs
 - ❖ The `page_no` of the inode
 - ❖ The byte offset of the inode
 - ❖ The length of the file segment header (in bytes)

Extents

- ❖ An extent is 1MB of consecutive pages
- ❖ `#define FSP_EXTENT_SIZE (1048576U / UNIV_PAGE_SIZE)`



InnoDB 특징

- 1. 역사
- 2. 아키텍처
- 3. 메모리
- 4. 디스크
- 5. 성능

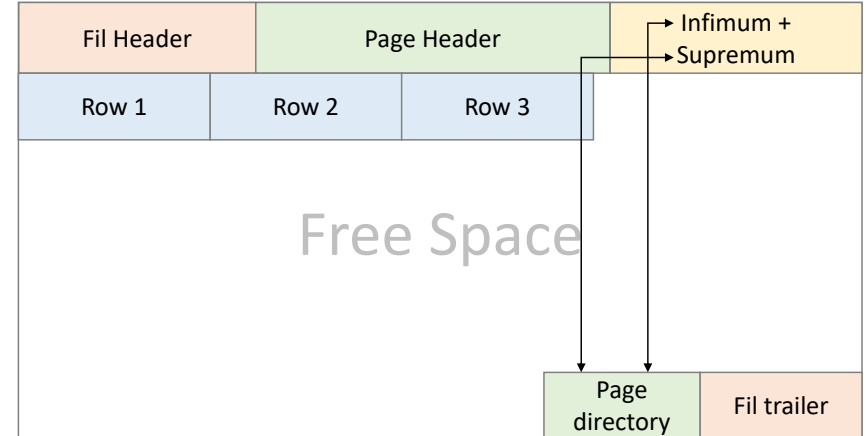
- 4.1 Redo Log File
- 4.2 Logical Structure
 - 4.2.1 Tablespace
 - 4.2.1.1 IBD File의 구조
 - 4.2.2 Segment, Extent
 - 4.2.3 Page
 - 4.2.3.1 Page Layout
 - 4.2.3.2 Directory Slot
 - 4.2.3.3 Page 내부 동작원리
- 4.3 Undo, MVCC Architecture
- 4.4 Purge
 - 4.4.1 Purge 정의 및 필요성
 - 4.4.2 Purge Update Undo
 - 4.4.3 Purge Garbage Space



Page Layout

MySQL/InnoDB Page Structure

- 'Page'라고 불리는 고정된 사이즈 단위 안에 모든 레코드를 저장
- 모든 페이지는 동일한 사이즈인 **16KB** (default)
- 페이지 안에는 레코드, 헤더, 트레일러를 포함, 총 **7가지 파트**로 구성
- 페이지 안에는 두 쌍의 헤더/트레일러
 - inner pair = Page Header + Page Directory
 - outer pair = Fil Header + Fil Trailer
- 헤더와 트레일러 사이에 있는 것은 'record'와 'free space'
- 페이지는 항상 바뀌지 않는 레코드인 'infimum'과 'supremum'으로 시작
- 새로운 레코드를 위한 공간은 사용자 레코드(아래로 생성)와 페이지 디렉토리(위로 생성) 사이에 위치

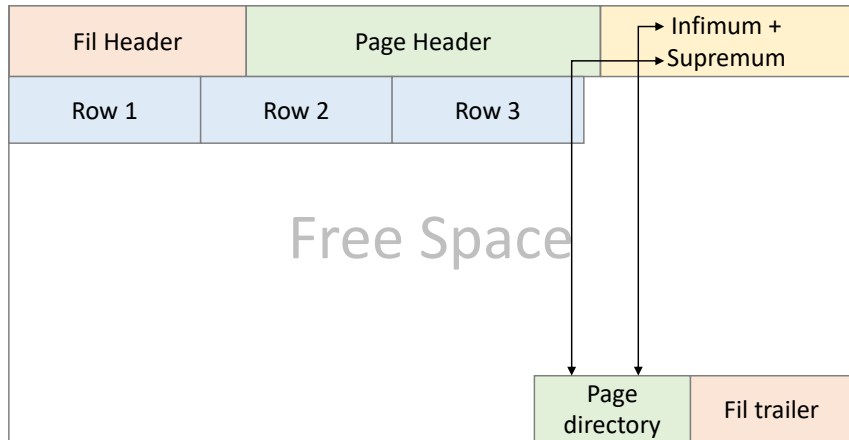


1 page = 16K

• Fil Headers	파일 페이지 정보
• Page Header	페이지의 요약정보
• Infimum, Supremum	시스템 레코드
• User records	유저 레코드
• Free Space	프리 스페이스
• Page Directory	레코드 포인터
• Fil Trailer	체크섬



Page Layout



1 page = 16K

- Fil Headers 파일 페이지 정보
- Page Header 페이지의 요약정보
- Infimum, Supremum 시스템 레코드
- User records 유저 레코드
- Free Space 프리 스페이스
- Page Directory 레코드 포인터
- Fil Trailer 체크섬

00c000	05 6b c0 e0	00 00 00 03	ff ff ff ff	ff ff ff ff	>.k.....<
00c010	00 00 00 00	05 4e b0 a3	45 bf 00 00	00 00 00 00	>....N..E.....<
00c020	00 00 00 00	01 c1 00 02	01 08 80 05	00 00 00 00	>.....<
00c030	00 e2 00 02	00 02 00 03	00 00 00 00	00 00 00 00	>.....<
00c040	00 00 00 00	00 00 00 00	01 d3 00 00	01 c1 00 00	>.....<
00c050	00 02 00 f2	00 00 01 c1	00 00 00 02	00 32 01 00	>.....2..<
00c060	02 00 1f 69	6e 66 69 6d	75 6d 00 04	00 0b 00 00	>...infimum.....<
00c070	73 75 70 72	65 6d 75 6d	03 06 02 0e	00 00 00 10	>supremum.....<
00c080	00 30 4a 4a	31 30 30 31	20 20 20 20	20 20 20 20	>.0JJ1001 <
00c090	00 00 00 01	bb aa 00 00	00 00 3d 1d	8c 44 20 44	>.....=.D D<
00c0a0	20 20 20 20	20 44 20 20	03 06 02 0e	00 00 00 18	> D<
00c0b0	00 30 4a 4a	31 30 30 32	20 20 20 20	20 20 20 20	>.0JJ1002 <
00c0c0	00 00 00 01	bb a1 80 00	00 00 2d 01	27 41 20 41	>.....-'A A<
00c0d0	20 20 20 20	20 41 20 20	03 06 02 0e	00 00 00 20	> A<
00c0e0	ff 8e 4a 4a	31 30 30 33	20 20 20 20	20 20 20 20	>..JJ1003 <
00c0f0	00 00 00 01	bb a1 80 00	00 00 2d 01	3e 41 20 41	>.....-'A A<
00c100	20 20 20 20	20 41 20 20	00 00 00 00	00 00 00 00	> A<
00c110	00 00 00 00	00 00 00 00	00 00 00 00	00 00 00 00	>.....<
*					
00fff0	00 00 00 00	00 70 00 63	05 6b c0 e0	05 4e b0 a3	>.....p.c.k...N..<



Page Layout

Fil Header

NO	Element	Size (byte)	Description
1	FIL_PAGE_SPACE_OR_CHKSUM	4	Checksum (~ MySQL 4.0.14: space ID)
2	FIL_PAGE_OFFSET	4	ordinal page number from start of space
3	FIL_PAGE_PREV	4	offset of previous page in key order
4	FIL_PAGE_NEXT	4	offset of next page in key order
5	FIL_PAGE_LSN	8	log serial number of page's latest log record
6	FIL_PAGE_TYPE	2	current defined types are: FIL_PAGE_INDEX, FIL_PAGE_UNDO_LOG, FIL_PAGE_INODE, FIL_PAGE_IBUF_FREE_LIST
7	FIL_PAGE_FILE_FLUSH_LSN	8	"the file has been flushed to disk at least up to this lsn" (log serial number), valid only on the first page of the file
8	FIL_PAGE_ARCH_LOG_NO_OR_SPACE_ID	4	Space ID (~ MySQL 4.1.X: the latest archived log file number at the time)



Page Layout

Page Header

NO	Element	Size (byte)	Description
1	page_n_dir_slots	2	number of directory slots in the Page Directory part; initial value = 2
2	page_heap_top	2	the beginning of free space
3	page_n_heap	2	number of heap records; initial value = 2
4	page_free	2	record pointer to first free record
5	page_garbage	2	"number of bytes in deleted records"
6	page_last_insert	2	record pointer to the last inserted record
7	page_direction	2	either PAGE_LEFT, PAGE_RIGHT, or PAGE_NO_DIRECTION
8	page_n_direction	2	number of consecutive inserts in the same direction, for example, "last 5 were all to the left"
9	page_n_recs	2	number of user records
10	page_max_trx_id	8	the highest ID of a transaction which might have changed a record on the page (only set for secondary indexes)
11	page_level	2	level within the index (0 for a leaf page)
12	page_index_id	8	identifier of the index the page belongs to
13	page_btr_seg_leaf	10	"file segment header for the leaf pages in a B-tree" (this is irrelevant here)
14	page_btr_seg_top	10	"file segment header for the non-leaf pages in a B-tree" (this is irrelevant here)



Page Layout

The Infimum and Supremum Records

NO	Element	Size (byte)	Description
1	Info Flags	4 bits	infimum, supremum(0)
2	Number of Records Owned	4 bits	The number of records "owned" by the current record in the page directory.
3	Order	13 bits	The order in which this record was inserted into the heap. infimum (0), supremum (1), user records(2~)
4	Record Type	3 bits	conventional (0), node pointer (1), infimum (2), supremum (3)
5	Next Record Offset	2	A relative offset from the current record to the origin of the next record within the page in ascending order by key.
6	Infimum	8	Index system records
12	Supremum	8	Index system records



Page Layout

User Records

NO	Element	Size (byte)	Description
1	variable field lengths	1-2	Variable-length field to store the length of the field data for that field.
2	Nullable field bitmap	1bit	Nullable field to store whether the field is NULL, rounded up to a whole number of bytes.
3	Info Flags	4bits	min_rec (1) : minimum record in a non-leaf level of the B+Tree deleted (2) : delete-marked (and will be actually deleted by a purge operation in the future)
4	Number of Records Owned	4bits	The number of records "owned" by the current record in the page directory.
5	Order	13bits	infimum (0), supremum (1), user record (2~)
6	Record Type	3bits	conventional (0), node pointer (1), infimum (2), supremum (3)
7	Next Record Offset	2	A relative offset from the current record to the origin of the next record within the page in ascending order by key.
8	Cluster Key Fields	N	The cluster key fields, concatenated together
9	Transaction ID	6	Transaction ID of the transaction which last modified this record.
10	Roll Pointer	7	insert flag (1bit), rollback segment (7bits), page number (4), offset (2)
11	Non-key Fields	K	All non-key fields concatenated together into a single byte stream.



Page Layout

Fil Trailer

NO	Element	Size (byte)	Description
1	Fil_Page_End_Lsn	8	앞 4 bytes = 페이지 체크섬 (05 6b c0 e0) 뒤 4 bytes = fil_page_lsn와 동일 (05 4e b0 a3)

- integrity를 위한 아키텍처 구조
- 장애를 대비하여 체크섬을 하고, 맨 처음 페이지와 같은 값인지 확인



InnoDB 특징

- 1. 역사
- 2. 아키텍처
- 3. 메모리
- 4. 디스크
- 5. 성능

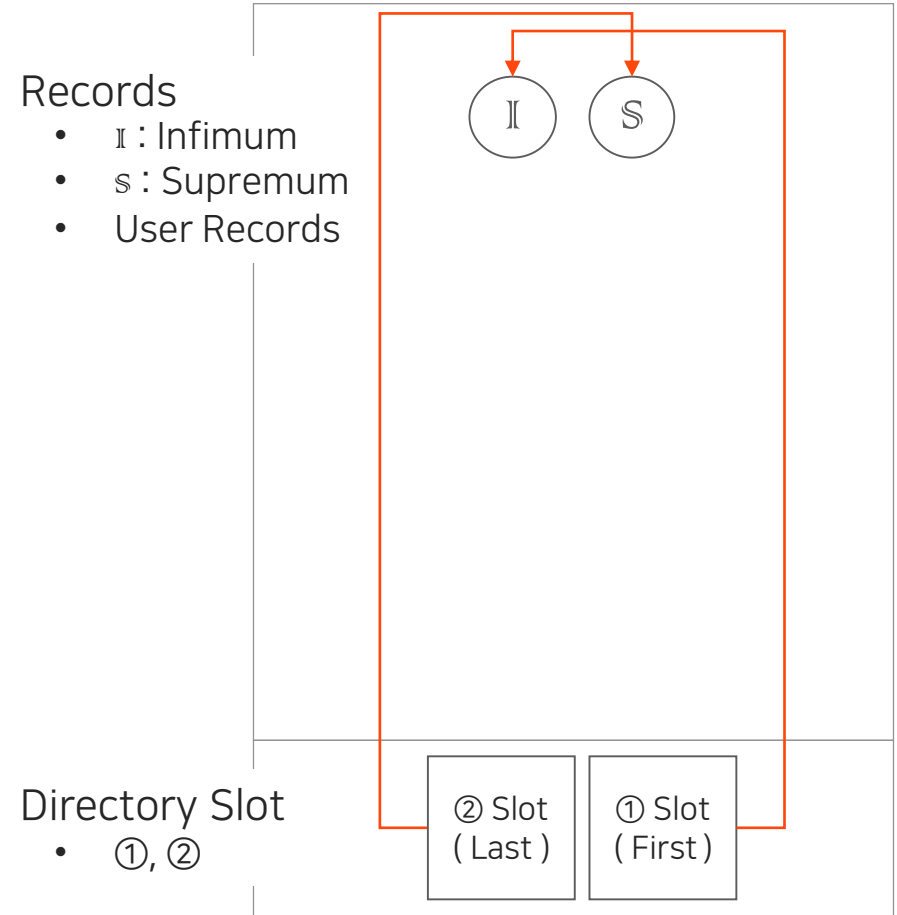
- 4.1 Redo Log File
- 4.2 Logical Structure
 - 4.2.1 Tablespace
 - 4.2.1.1 IBD File의 구조
 - 4.2.2 Segment, Extent
 - 4.2.3 Page
 - 4.2.3.1 Page Layout
 - 4.2.3.2 Directory Slot
 - 4.2.3.3 Page 내부 동작원리
- 4.3 Undo, MVCC Architecture
- 4.4 Purge
 - 4.4.1 Purge 정의 및 필요성
 - 4.4.2 Purge Update Undo
 - 4.4.3 Purge Garbage Space



Directory Slot

정의 및 목적

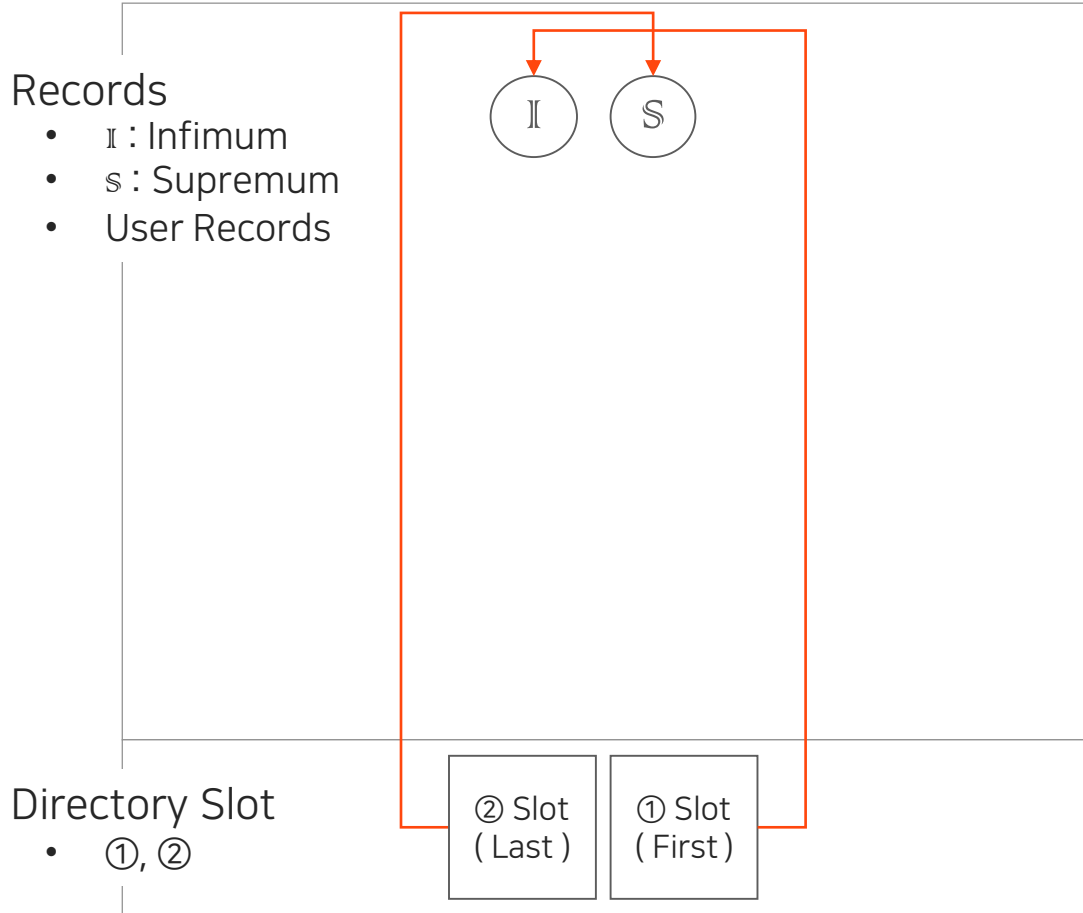
- 페이지에서 **이진탐색**을 이용하여 해당 레코드를 효율적으로 찾기 위해 사용
- 페이지 하단에 존재하는 **Fil Trailer 영역** 앞에 위치정보를 가지고 있는 슬롯 (2bytes)
- infimum을 가리키는 first slot과 supremum을 가리키는 last slot의 위치는 처음과 끝으로 고정
- 한 슬롯당 소유하는 레코드 수는 최소 4개에서 최대 8개까지 소유하고, 가장 첫 슬롯은 Infimum 1개, 가장 마지막 슬롯은 1~8개를 소유
- 페이지의 레코드가 많아져 슬롯이 추가될 경우, **아래에서 위로 증가**





Directory Slot

1. create table exem_i.t1



① Slot (First) 1 Row	② Slot (Last) 1 Row
Infimum	Supremum



Directory Slot

1. create table exem_i.t1

Page Header

- Slot Count : 2
- Record Count : 0

Directory Slot 1

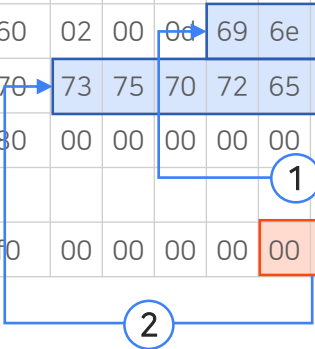
- offset : Infimum
- n_owned : 1

Directory Slot 2

- offset : Supremum
- n_owned : 1

T1.ibd space (Space ID : e0)					
Page 0	Page 1	...	Page 3	...	Page 64

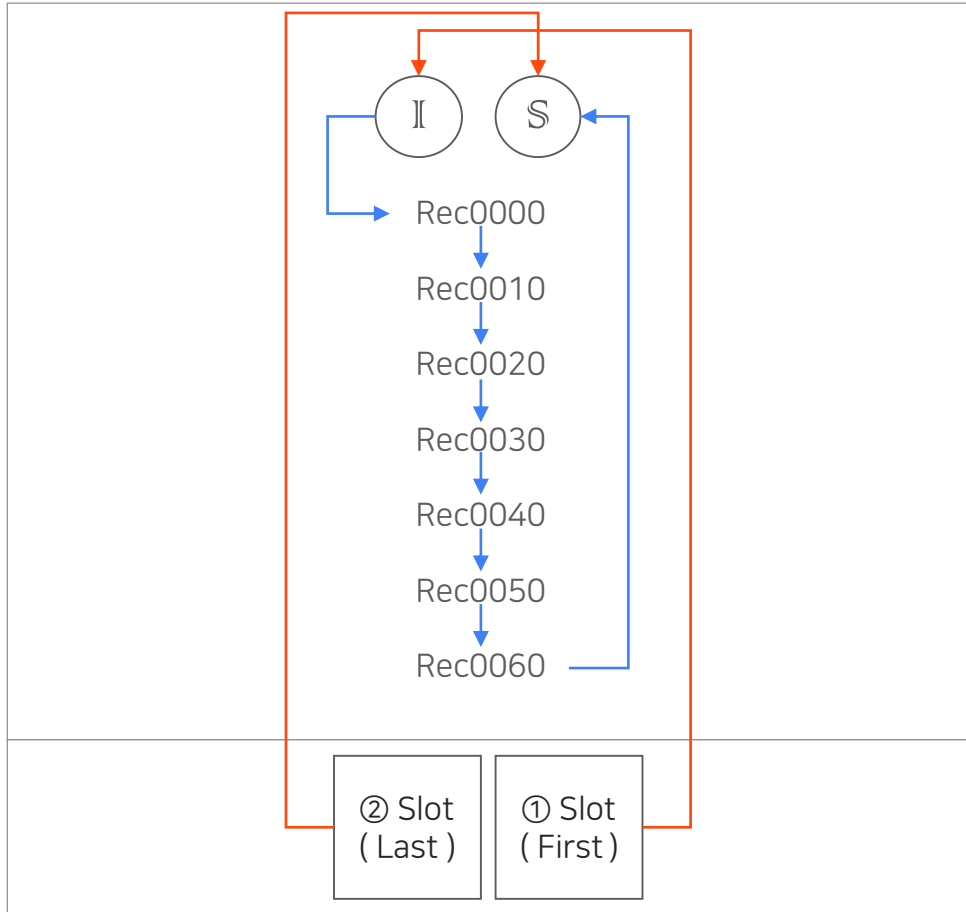
000000	a8	09	5a	2f	00	00	00	03	ff	ff	ff	ff	ff	ff	ff	>..Z/.....<
000010	00	00	00	00	0f	f2	d4	83	45	bf	00	00	00	00	00	>.....E.....<
000020	00	00	00	00	01	68	00	02	00	78	80	02	00	00	00	>....h...x.....<
000030	00	00	00	05	00	00	00	00	00	00	00	00	00	00	00	>.....<
000040	00	00	00	00	00	00	00	01	ac	00	00	01	68	00	00	>.....h..<
000050	00	02	00	f2	00	00	01	68	00	00	00	02	00	32	01	>.....h.....2..<
000060	02	00	00	69	6e	66	69	6d	75	6d	00	01	00	0b	00	>...infimum.....<
000070	73	75	70	72	65	6d	75	6d	00	00	00	00	00	00	00	>supremum.....<
000080	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	>.....<
*																
003ff0	00	00	00	00	00	70	00	63	a8	09	5a	2f	0f	f2	d4	>.....p.c..Z/....<





Directory Slot

2. Insert into exem_i.t1 ... 7 Records



① Slot (First) 1 Row	② Slot (Last) 8 Rows
Inifimum	Rec0000
	Rec0010
	Rec0020
	Rec0030
	Rec0040
	Rec0050
	Rec0060
	Supremum



Directory Slot

2. Insert into exem_i.t1 ... 7 Records

Page Header

- Slot Count : 2
- Record Count : 7

Directory Slot 2

- offset : Supremum
- n_owned : 8

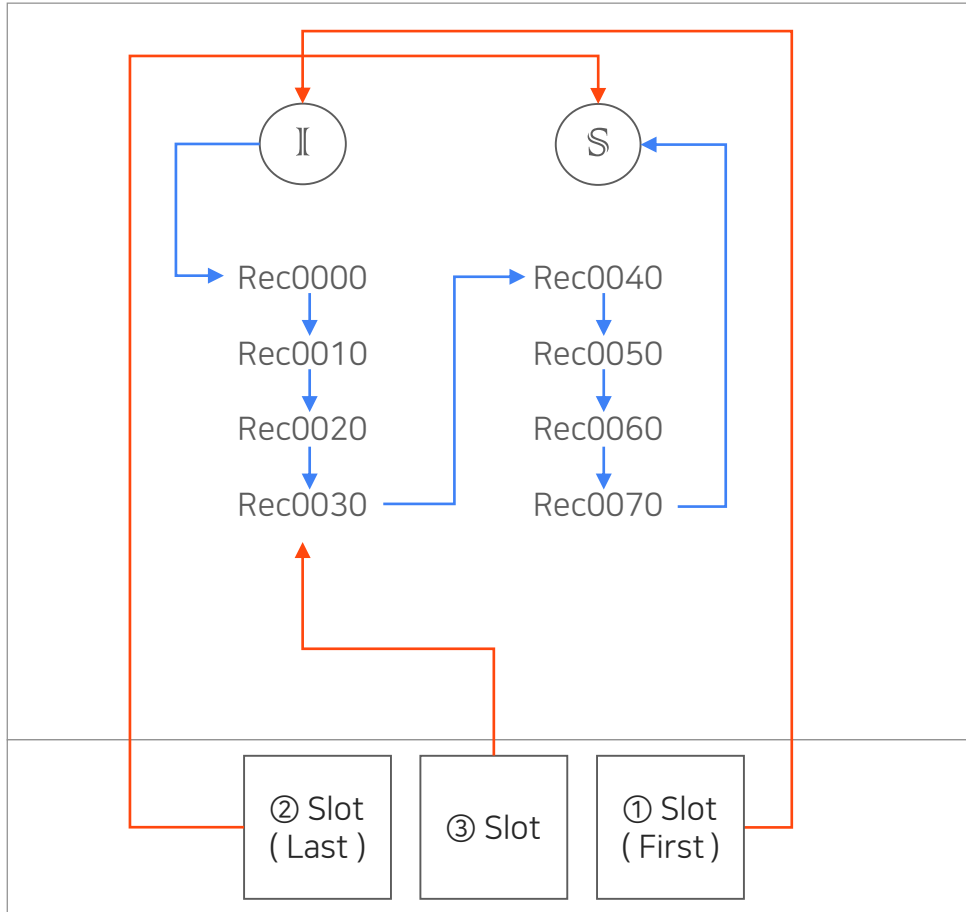
000000	04	46	ec	83	00	00	00	03	ff	ff	ff	ff	ff	ff	ff	>.F.....<
000010	00	00	00	00	0f	f2	ed	68	45	bf	00	00	00	00	00	>.....hE.....<
000020	00	00	00	00	01	68	00	02	0e	78	80	09	00	00	00	>....h...x.....<
000030	0c	82	00	02	00	06	00	07	00	00	00	00	00	00	00	>.....<
000040	00	00	00	00	00	00	00	00	01	ac	00	00	01	68	00	>.....h..<
000050	00	02	00	f2	00	00	01	68	00	00	00	02	00	32	01	>.....h....2..<
000060	02	00	1f	69	6e	66	69	6d	75	6d	00	08	00	0b	00	>...infimum.....<
000070	73	75	70	72	65	6d	75	6d	d0	81	0b	0e	00	00	00	>supremum.....<
000080	02	00	52	65	63	30	30	30	30	20	20	20	20	20	20	>..Rec0000 <
000090	00	00	00	02	0a	bf	a2	00	00	00	04	01	10	44	20	>.....D <
0000a0	20	20	20	20	20	20	20	20	31	20	20	20	20	20	20	> 1 <
0000b0	20	20	20	20	20	20	20	20	20	20	20	20	20	20	20	> <
*																
000c70	20	20	20	20	20	20	20	20	d0	81	0b	0e	00	00	00	>@<
000c80	f3	ee	52	65	63	30	30	36	30	20	20	20	20	20	20	>..Rec0060 <
000c90	00	00	00	02	0a	c9	a1	00	00	00	04	01	10	4c	20	>.....L <
000ca0	20	20	20	20	20	20	20	20	31	20	20	20	20	20	20	> 1 <
000cb0	20	20	20	20	20	20	20	20	20	20	20	20	20	20	20	> <
*																
003ff0	00	00	00	00	00	70	00	63	04	46	ec	83	0f	f2	ed	>.....p.c.F.....h<

2



Directory Slot

3. Insert into exem_i.t1 ... 1 Records



① Slot (First) 1 Row	② Slot (Last) 5 Rows	③ Slot 4 Rows
Infimum	Rec0040	Rec0000
	Rec0050	Rec0010
	Rec0060	Rec0020
	Rec0070	Rec0030
	Supremum	



Directory Slot

3. Insert into exem_i.t1 ... 1 Records

Page Header

- Slot Count : 3
- Record Count : 8

Directory Slot 2

- offset : Supremum
- n_owned : 5

Directory Slot 3

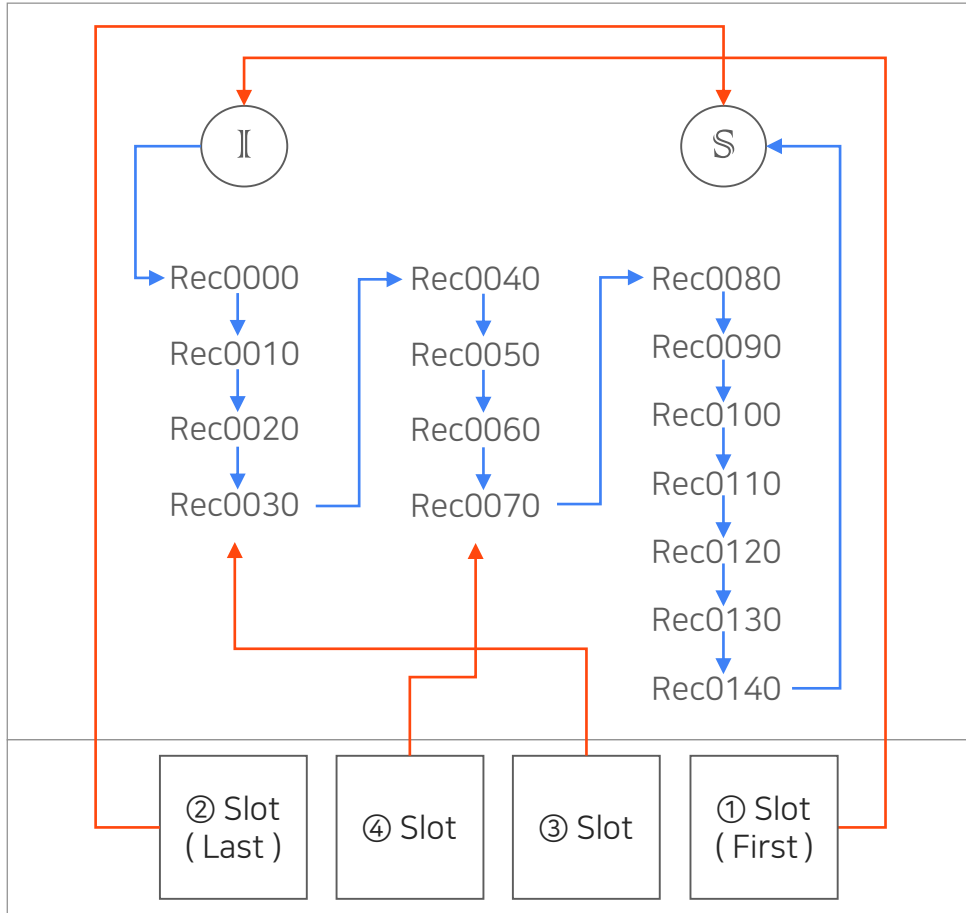
- offset : Rec0030
- n_owned : 4

000000	28	71	b4	0f	00	00	00	03	ff	ff	ff	ff	ff	ff	ff	>(q.....<
000010	00	00	00	00	0f	f2	f6	10	45	bf	00	00	00	00	00	>.....E.....<
000020	00	00	00	00	01	68	00	03	10	78	80	0a	00	00	00	>.....h...x.....<
000030	0e	82	00	02	00	07	00	08	00	00	00	00	00	00	00	>.....<
000040	00	00	00	00	00	00	00	00	01	ac	00	00	01	68	00	>.....h..<
000050	00	02	00	f2	00	00	01	68	00	00	00	02	00	32	01	>.....h....2..<
000060	02	00	1f	69	6e	66	69	6d	75	6d	00	05	00	0b	00	>...infimum.....<
000070	73	75	70	72	65	6d	75	6d	d0	81	0b	0e	00	00	10	>supremum.....<
000080	02	00	52	65	63	30	30	30	30	20	20	20	20	20	20	>..Rec0000 <
000090	00	00	00	02	0a	bf	a2	00	00	00	04	01	10	44	20	>.....D <
0000a0	20	20	20	20	20	20	20	20	31	20	20	20	20	20	20	> 1 <
*																
000670	20	20	20	20	20	20	20	20	d0	81	0b	0e	00	04	00	>.....(<
000680	02	00	52	65	63	30	30	33	30	20	20	20	20	20	20	>..Rec0030 <
000690	00	00	00	02	0a	c6	a1	00	00	00	04	01	10	57	20	>.....W <
*																
003ff0	00	00	00	00	00	70	00	63	04	46	ec	83	0f	f2	ed	>.....p.c.F.....h<
000e70	20	20	20	20	20	20	20	20	d0	81	0b	0e	00	00	00	>.....H<
000e80	f1	ee	52	65	63	30	30	37	30	20	20	20	20	20	20	>..Rec0070 <
000e90	00	00	00	02	0a	ce	a1	00	00	00	04	01	10	52	20	>.....R <
000ea0	20	20	20	20	20	20	20	20	31	20	20	20	20	20	20	> 1 <
000eb0	20	20	20	20	20	20	20	20	20	20	20	20	20	20	20	> <
*																
003ff0	00	00	00	70	06	82	00	63	28	71	b4	0f	0f	f2	f6	>...p...c(q.....<



Directory Slot

4. Insert into exem_i.t1 ... 7 Records



① Slot (First) 1 Row	② Slot (Last) 8 Rows	③ Slot 4 Rows	④ Slot 4 Rows
Infimum	Rec0080	Rec0000	Rec0040
	Rec0090	Rec0010	Rec0050
	Rec0100	Rec0020	Rec0060
	Rec0110	Rec0030	Rec0070
	Rec0120		
	Rec0130		
	Rec0140		
	Supremum		



Directory Slot

4. Insert into exem_i.t1 ... 7 Records

Page Header

- Slot Count : 4
- Record Count : 15

Directory Slot 2

- offset : Supremum
- n_owned : 8

Directory Slot 4

- offset : Rec0070
- n_owned : 4

000000	f1	5b	d8	7f	00	00	00	03	ff	ff	ff	ff	ff	ff	ff	>.[.....<
000010	00	00	00	00	0f	f3	0b	f7	45	bf	00	00	00	00	00	>.....E.....<
000020	00	00	00	00	01	68	00	04	1e	78	80	11	00	00	00	>....h...x.....<
000030	1c	82	00	02	00	0e	00	0f	00	00	00	00	00	00	00	>.....<
000040	00	00	00	00	00	00	00	00	01	ac	00	00	01	68	00	>.....h..<
000050	00	02	00	f2	00	00	01	68	00	00	00	02	00	32	01	>.....h....2..<
000060	02	00	1f	69	6e	66	69	6d	75	6d	00	08	00	0b	00	>...infimum.....<
000070	73	75	70	72	65	6d	75	6d	d0	81	0b	0e	00	00	10	>supremum.....<
000080	02	00	52	65	63	30	30	30	30	20	20	20	20	20	20	>..Rec0000 <
000090	00	00	00	02	0a	bf	a2	00	00	00	04	01	10	44	20	>.....D <
0000a0	20	20	20	20	20	20	20	20	31	20	20	20	20	20	20	> 1 <
*																
000e70	20	20	20	20	20	20	20	20	d0	81	0b	0e	00	04	00	>H<
000e80	02	00	52	65	63	30	30	37	30	20	20	20	20	20	20	>..Rec0070 <
000e90	00	00	00	02	0a	ce	a1	00	0	0	0	0	1	5	2	>.....R
000ea0	20	20	20	20	20	20	20	20	31	20	20	20	20	20	20	> 1 <
*																
003ff0	00	70	0e	82	06	82	00	63	f1	5b	d8	7f	0f	f3	0b	>.p.....c.[.....<

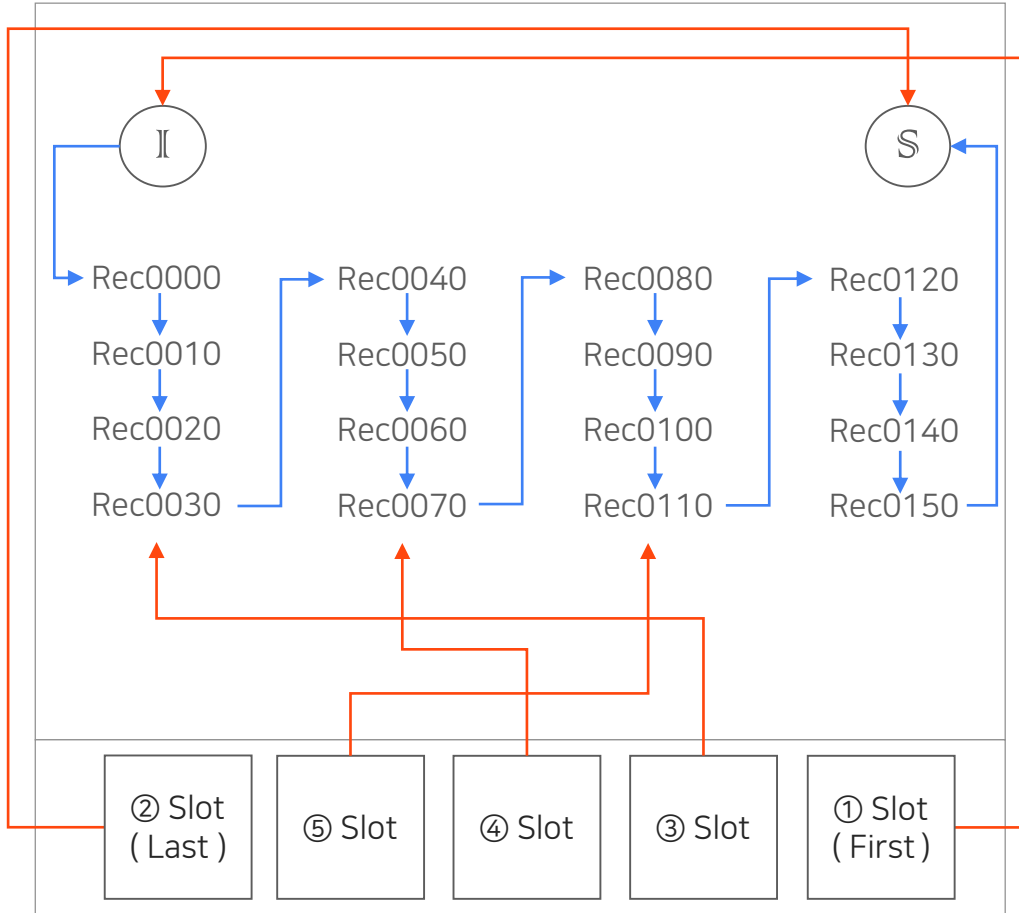
2

4



Directory Slot

5. Insert into exem_i.t1 ... 1 Records



① Slot 1 Row	② Slot 5 Rows	③ Slot 4 Rows	④ Slot 4 Rows	⑤ Slot 4 Rows
Infimum	Rec0120	Rec0000	Rec0040	Rec0080
	Rec0130	Rec0010	Rec0050	Rec0090
	Rec0140	Rec0020	Rec0060	Rec0100
	Rec0150	Rec0030	Rec0070	Rec0110
	Supremum			



Directory Slot

5. Insert into exem_i.t1 ... 1 Records

Page Header

- Slot Count : 5
- Record Count : 16

Directory Slot 2

- offset : Supremum
- n_owned : 5

Directory Slot 5

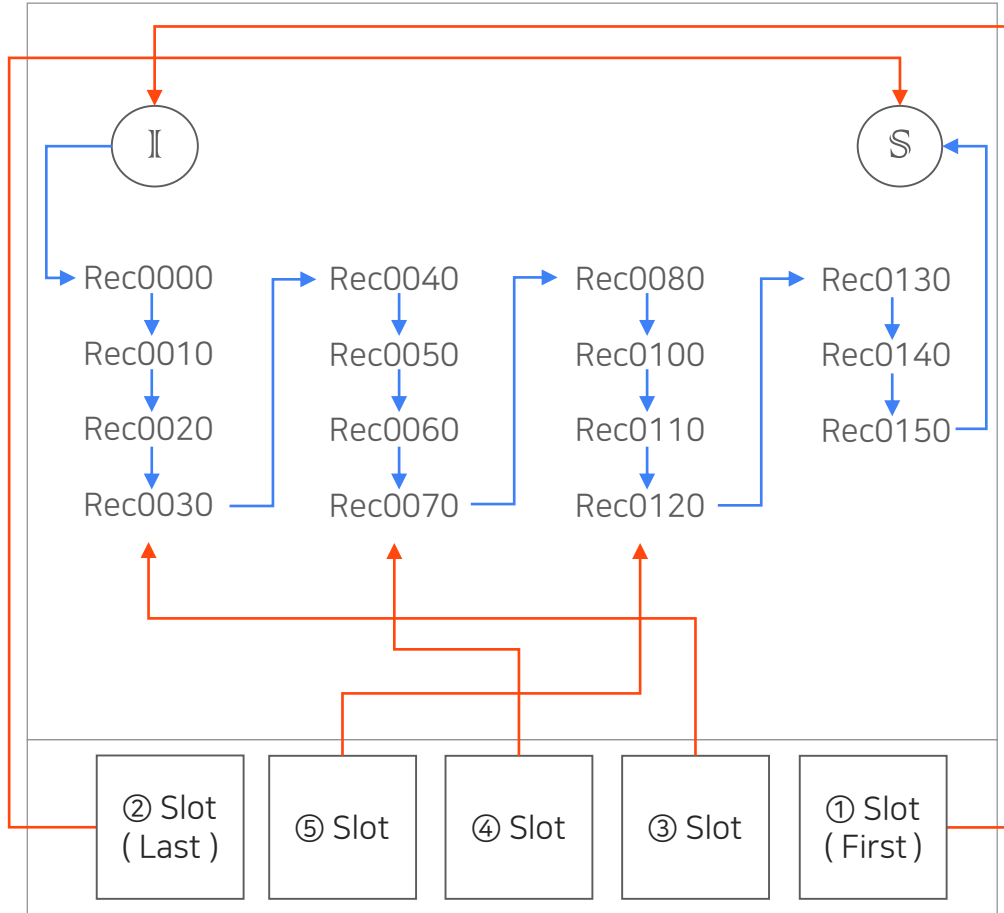
- offset : Rec0110
- n_owned : 4

000000	e9	9b	d5	07	00	00	00	03	ff	ff	ff	ff	ff	ff	ff	>.....<
000010	00	00	00	00	0f	f3	14	9f	45	bf	00	00	00	00	00	>.....E.....<
000020	00	00	00	00	01	68	00	05	20	78	80	12	00	00	00	>....h.. x.....<
000030	1e	82	00	02	00	0f	00	10	00	00	00	00	00	00	00	>.....<
000040	00	00	00	00	00	00	00	00	01	ac	00	00	01	68	00	>.....h..<
000050	00	02	00	f2	00	00	01	68	00	00	00	02	00	32	01	>.....h....2..<
000060	02	00	1f	69	6e	66	69	6d	75	6d	00	05	00	0b	00	>...infimum.....<
000070	73	75	70	72	65	6d	75	6d	d0	81	0b	0e	00	00	10	>supremum.....<
000080	02	00	52	65	63	30	30	30	30	20	20	20	20	20	20	>..Rec0000 <
000090	00	00	00	02	0a	bf	a2	00	00	00	04	01	10	44	20	>.....D <
0000a0	20	20	20	20	20	20	20	20	31	20	20	20	20	20	20	> 1 <
																*
001670	20	20	20	20	20	20	20	20	d0	81	0b	0e	00	04	00	>h<
001680	02	00	52	65	63	30	31	31	30	20	20	20	20	20	20	>..Rec0110 <
																*
003fe0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	70	>.....p <
003ff0	16	82	0e	82	06	82	00	63	e9	9b	d5	07	0f	f3	14	>.....c.....<



Directory Slot

6. Delete from exem_i.t1 Where c1 = 'Rec0090';



① Slot 1 Row	② Slot 4 Rows	③ Slot 4 Rows	④ Slot 4 Rows	⑤ Slot 4 Rows
Infimum	Rec0130	Rec0000	Rec0040	Rec0080
	Rec0140	Rec0010	Rec0050	Rec0090
	Rec0150	Rec0020	Rec0060	Rec0100
	Supremum	Rec0030	Rec0070	Rec0110
				Rec0120



Directory Slot

6. Delete from exem_i.t1 Where c1 = 'Rec0090';

Page Header

- Slot Count : 5
- Garbage Pointer : Rec0090
- Garbage Size : 0x200
- Record Count : 15

Rec0080

- Next Record : Rec0100

Rec0090

- Next Record : NULL

Directory Slot 2

- offset : Supremum
- n_owned : 4

Directory Slot 5

- offset : Rec0120
- n_owned : 4

000000	a3	7c	4c	50	00	00	00	03	ff	ff	ff	ff	ff	ff	ff	>. LP.....<
000010	00	00	00	00	0f	f3	15	bf	45	bf	00	00	00	00	00	>.....E.....<
000020	00	00	00	00	01	68	00	05	20	78	80	12	12	82	02	>....h.. x.....<
000030	00	00	00	02	00	0f	00	0f	00	00	00	00	00	00	00	>.....<
000040	00	00	00	00	00	00	00	00	01	ac	00	00	01	68	00	>.....h..<
000050	00	02	00	f2	00	00	01	68	00	00	00	02	00	32	01	>.....h....2..<
000060	02	00	1f	69	6e	66	69	6d	75	6d	00	04	00	0b	00	>...infimum.....<
000070	73	75	70	72	65	6d	75	6d	d0	81	0b	0e	00	00	10	>supremum.....<
000080	02	00	52	65	63	30	30	30	30	20	20	20	20	20	20	>..Rec0000 <
000090	00	00	00	02	0a	bf	a2	00	00	00	04	01	10	44	20	>.....D <
0000a0	20	20	20	20	20	20	20	20	31	20	20	20	20	20	20	> 1 <
*																
001070	20	20	20	20	20	20	20	20	d0	81	0b	0e	00	00	50	>P<
001080	04	00	52	65	63	30	30	38	30	20	20	20	20	20	20	>..Rec0080 <
...																
001270	20	20	20	20	20	20	20	20	d0	81	0b	0e	00	20	00	>X<
001280	00	00	52	65	63	30	30	39	30	20	20	20	20	20	20	>..Rec0090 <
...																
001470	20	20	20	20	20	20	20	20	d0	81	0b	0e	00	00	00	>`<
001480	02	00	52	65	63	30	31	30	30	20	20	20	20	20	20	>..Rec0100 <
...																
001870	20	20	20	20	20	20	20	20	d0	81	0b	0e	00	04	00	>p<
001880	02	00	52	65	63	30	31	32	30	20	20	20	20	20	20	>..Rec0120 <
...																
003fe0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	70	>p<
003ff0	18	82	0e	82	06	82	00	63	a3	7c	4c	50	0f	f3	15	>.....c. LP....<

2

5



InnoDB 특징

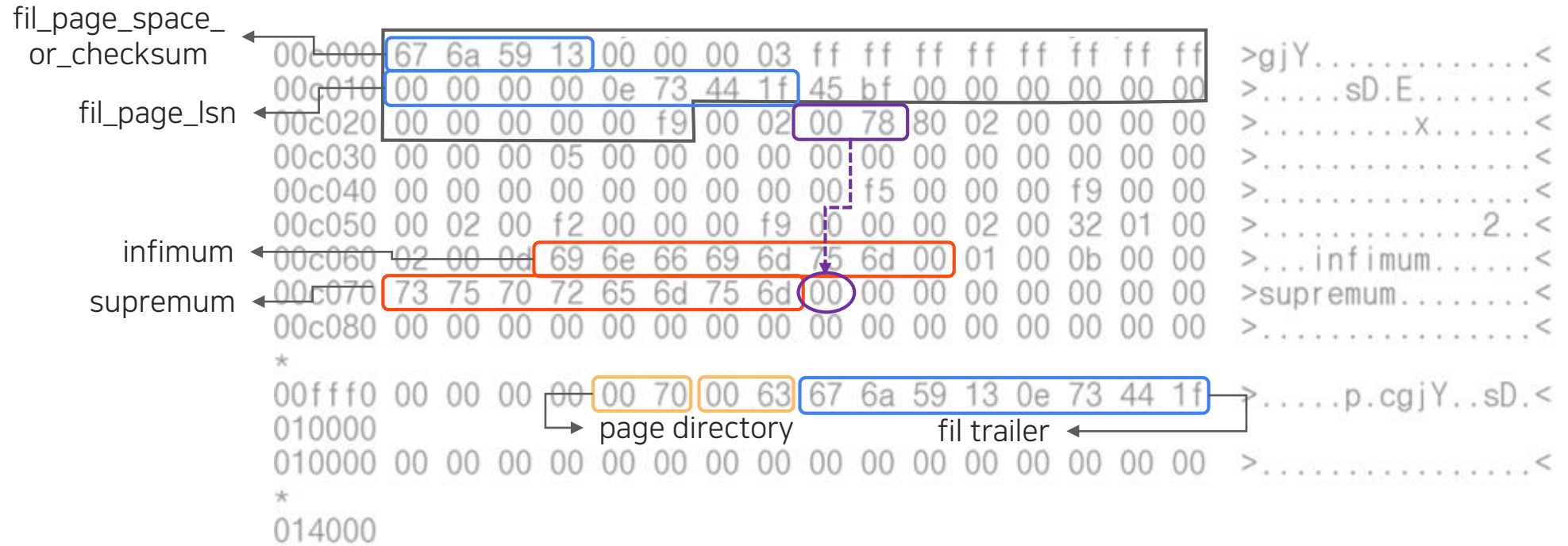
- 1. 역사
- 2. 아키텍처
- 3. 메모리
- 4. 디스크
- 5. 성능

- 4.1 Redo Log File
- 4.2 Logical Structure
 - 4.2.1 Tablespace
 - 4.2.1.1 IBD File의 구조
 - 4.2.2 Segment, Extent
 - 4.2.3 Page
 - 4.2.3.1 Page Layout
 - 4.2.3.2 Directory Slot
 - 4.2.3.3 Page 내부 동작원리
- 4.3 Undo, MVCC Architecture
- 4.4 Purge
 - 4.4.1 Purge 정의 및 필요성
 - 4.4.2 Purge Update Undo
 - 4.4.3 Purge Garbage Space



Page 내부 동작원리

테이블 생성 후, 초기 상태





Page 내부 동작원리

A Insert (insert into exem_i.t1 values ('Rec001','A',rpad(' ',@v1,' '));)

```

00c000 6a c1 da 65 00 00 00 03 ff ff ff ff ff ff ff ff >j...e.....<
00c010 00 00 00 00 0e 73 4c 67 45 bf 00 00 00 00 00 00 >.....sLgE.....<
00c020 00 00 00 00 00 f9 00 02 02 78 80 03 00 00 00 00 >.....x.....<
00c030 00 82 00 05 00 00 00 01 00 00 00 00 00 00 00 00 >.....<
00c040 00 00 00 00 00 00 00 00 00 f5 00 00 00 f9 00 00 >.....<
00c050 00 02 00 f2 00 00 00 f9 00 00 00 02 00 32 01 00 >.....2..<
00c060 02 00 1f 69 6e 66 69 6d 75 6d 00 02 00 0b 00 00 >...inifum.....<
00c070 73 75 70 72 65 6d 75 6d d0 81 0b 0e 00 00 00 10 >supremum.....<
00c080 ff ee 52 65 63 30 30 31 20 20 20 20 20 20 20 20 >..Rec001 <
00c090 00 00 00 00 77 8b a1 00 00 00 04 01 10 41 20 20 >...w.....A <
00c0a0 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 > <
*
00c270 20 20 20 20 20 20 20 20 00 00 00 00 00 00 00 00 > .....<
00c280 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 >.....<
*
00ff00 00 00 00 00 00 70 00 63 6a c1 da 65 0e 73 4c 67 >.....p.cj...e.sLg<
010000
010000 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 >.....<
*
014000

```

Transaction ID	00 00 00 00 77 8b
Roll Pointer	a1 00 00 00 04 01 10



Page 내부 동작원리

B로 Update (update exem_i.t1 set c2='B' where c1 = 'Rec001';)

```

00c000 7f e0 bd ee 00 00 00 03 ff ff ff ff ff ff ff ff >.....<
00c010 00 00 00 00 0e 73 4d 2f 45 bf 00 00 00 00 00 00 >.....sM/E.<
00c020 00 00 00 00 00 f9 00 02 02 78 80 03 00 00 00 00 >.....X.<
00c030 00 82 00 05 00 00 00 01 00 00 00 00 00 00 00 00 >.....<
00c040 00 00 00 00 00 00 00 00 00 f5 00 00 00 f9 00 00 >.....<
00c050 00 02 00 f2 00 00 00 f9 00 00 00 02 00 32 01 00 >.....2.<
00c060 02 00 1f 69 6e 66 69 6d 75 6d 00 02 00 0b 00 00 >...infinum.<
00c070 73 75 70 72 65 6d 75 6d d0 81 0b 0e 00 00 00 10 >supremum.<
00c080 ff ee 52 65 63 30 30 31 20 20 20 20 20 20 20 20 >..Rec001<
00c090 00 00 00 00 77 8c 21 00 00 00 05 06 d1 42 20 20 >...w!.....B<
00c0a0 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 >.....<
*
00c270 20 20 20 20 20 20 20 20 00 00 00 00 00 00 00 00 >.....<
00c280 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 >.....<
*
00fff0 00 00 00 00 00 70 00 63 7f e0 bd ee 0e 73 4d 2f >.....p.c.....sM/<
010000
010000 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 >.....<
*
014000

```

Update Undo

```

0006d0 00 07 04 1c 00 80 f4 00 00 00 00 77 8b e0 a1 00 >.....w....<
0006e0 00 00 04 01 10 0e 52 65 63 30 30 31 20 20 20 20 >.....Rec001<
0006f0 20 20 20 20 01 03 0b 41 20 20 20 20 20 20 20 20 >...A<
000700 20 20 06 d1 00 00 00 00 00 00 00 00 00 00 00 00 >.....<
000710 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 >.....<
*
003ff0 00 00 00 00 00 00 00 00 d0 c1 d2 52 0e 73 4c e7 >.....R.sL.<
004000

```

top_page_no	[5]
top_offset	[1745][0x6d1]

Transaction ID	00 00 00 00 77 8c
Roll Pointer	21 00 00 00 05 06 d1



Page 내부 동작원리

C로 Update (update exem_i.t1 set c2='C' where c1 = 'Rec001';)

```

00c000 58 80 a6 94 00 00 00 03 ff ff ff ff ff ff ff >X.....<
00c010 00 00 00 00 0e 73 4e 5e 45 bf 00 00 00 00 00 >.....sN^E.<
00c020 00 00 00 00 00 f9 00 02 02 78 80 03 00 00 00 >.....x.<
00c030 00 82 00 05 00 00 00 01 00 00 00 00 00 00 00 >.....<
00c040 00 00 00 00 00 00 00 00 00 f5 00 00 00 f9 00 >.....<
00c050 00 02 00 f2 00 00 00 f9 00 00 00 02 00 32 01 >.....2.<
00c060 02 00 1f 69 6e 66 69 6d 75 6d 00 02 00 0b 00 >...infimum.<
00c070 73 75 70 72 65 6d 75 6d d0 81 0b 0e 00 00 00 10 >supremum.<
00c080 ff ee 52 65 63 30 30 31 20 20 20 20 20 20 20 >..Rec001<
00c090 00 00 00 00 77 8e 21 00 00 00 05 07 be 43 20 20 >...w.!...C<
00c0a0 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 >.....<
*
00c270 20 20 20 20 20 20 20 20 00 00 00 00 00 00 00 >.....<
00c280 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 >.....<
*
00fff0 00 00 00 00 00 70 00 63 58 80 a6 94 0e 73 4e 5e >.....p.cX....sN^<
010000
010000 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 >.....<
*
014000

```

Update Undo

```

0006d0 00 07 04 1c 00 80 f4 00 00 00 00 77 8b e0 a1 00 >.....w.....<
0006e0 00 00 04 01 10 0e 52 65 63 30 30 31 20 20 20 20 >.....Rec001<
0006f0 20 20 20 20 20 01 03 0b 41 20 20 20 20 20 20 20 >...A<
000700 20 20 06 d1 00 00 00 00 00 00 77 8e 00 00 00 00 >.....w.....<
000710 00 00 00 00 00 01 07 be 00 00 00 00 00 00 00 00 >.....<
000720 00 00 00 00 06 17 00 00 00 00 00 00 00 00 00 00 >.....<
000730 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 >.....<
*
0007b0 00 00 00 00 00 00 00 00 00 00 00 00 00 07 f1 >.....<
0007c0 1c 00 80 f4 00 00 00 00 77 8c e0 21 00 00 00 05 >.....w.!.....<
0007d0 06 d1 0e 52 65 63 30 30 31 20 20 20 20 20 20 20 >...Rec001<
0007e0 20 01 03 0b 42 20 20 20 20 20 20 20 20 20 07 >...B<
0007f0 be 00 00 00 00 00 00 00 00 00 00 00 00 00 00 >.....<
000800 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 >.....<
*
003ff0 00 00 00 00 00 00 00 00 bd f7 2d 67 0e 73 4e 16 >.....-g.sN.<
004000

```

top_page_no	[5]
top_offset	[1982][0x7be]

Transaction ID	00 00 00 00 77 8e
Roll Pointer	21 00 00 00 05 07 be



Page 내부 동작원리

D로 Update (update exem_i.t1 set c2='D' where c1 = 'Rec001' ;)

```

00c000 17 04 e8 6c 00 00 00 03 ff ff ff ff ff ff ff ff >...l.....<
00c010 00 00 00 00 0e 73 4f 7d 45 bf 00 00 00 00 00 00 >.....s0}E.....<
00c020 00 00 00 00 00 f9 00 02 02 78 80 03 00 00 00 00 >.....x.....<
00c030 00 82 00 05 00 00 00 01 00 00 00 00 00 00 00 >.....<
00c040 00 00 00 00 00 00 00 00 00 f5 00 00 00 f9 00 00 >.....<
00c050 00 02 00 f2 00 00 00 f9 00 00 00 02 00 32 01 00 >.....2.<
00c060 02 00 1f 69 6e 66 69 6d 75 6d 00 02 00 0b 00 00 >...infimum.....<
00c070 73 75 70 72 65 6d 75 6d d0 81 0b 0e 00 00 00 10 >supremum.....<
00c080 ff ee 52 65 63 30 30 31 20 20 20 20 20 20 20 20 >..Rec001 <
00c090 00 00 00 00 77 90 21 00 00 00 05 08 ab 44 20 20 >...w.!.....D <
00c0a0 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 > <
*
00c270 20 20 20 20 20 20 20 20 00 00 00 00 00 00 00 00 > .....<
00c280 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 >.....<
*
00fff0 00 00 00 00 00 70 00 63 17 04 e8 6c 0e 73 4f 7d >.....p.c...l.s0}<
010000
010000 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 >.....<
*
014000

```

Transaction ID	00 00 00 00 77 90
Roll Pointer	21 00 00 00 05 08 ab

Update Undo

```

0006d0 00 07 04 1c 00 80 f4 00 00 00 00 77 8b e0 a1 00 >.....w.....<
0006e0 00 00 04 01 10 0e 52 65 63 30 30 31 20 20 20 20 >.....Rec001 <
0006f0 20 20 20 20 01 03 0b 41 20 20 20 20 20 20 20 20 >...A <
000700 20 20 06 d1 00 00 00 00 00 00 77 8e 00 00 00 00 >.....w.....<
000710 00 00 77 8f 00 00 07 be 00 00 00 00 00 00 00 00 >..w.....<
000720 00 00 07 f1 06 17 ff ff ff ff 00 00 00 00 00 05 >.....<
000730 06 39 00 00 00 00 00 00 00 00 00 00 00 00 00 >.9.....<
000740 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 >.....<
*
0007b0 00 00 00 00 00 00 00 00 00 00 00 00 00 07 f1 >.....<
0007c0 1c 00 80 f4 00 00 00 77 8c e0 21 00 00 00 05 >.....w.!.....<
0007d0 06 d1 0e 52 65 63 30 30 31 20 20 20 20 20 20 20 >...Rec001 <
0007e0 20 01 03 0b 42 20 20 20 20 20 20 20 20 20 07 >...B <
0007f0 be 00 00 00 00 00 00 77 90 00 00 00 00 00 00 >.....w.....<
000800 00 00 01 08 ab 00 00 00 00 00 00 00 00 00 00 >.....<
000810 00 07 04 00 00 00 00 00 00 00 00 00 00 00 00 >.....<
000820 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 >.....<
*
0008a0 00 00 00 00 00 00 00 00 00 00 00 08 de 1c 00 80 >.....<
0008b0 f4 00 00 00 00 77 8e e0 21 00 00 00 05 07 be 0e >.....w.!.....<
0008c0 52 65 63 30 30 31 20 20 20 20 20 20 20 01 03 >Rec001 <
0008d0 0b 43 20 20 20 20 20 20 20 20 20 08 ab 00 00 >.C <
0008e0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 >.....<
*
003ff0 00 00 00 00 00 00 00 71 5b 06 87 0e 73 4f 35 >.....q[...s05<
004000

```

top_page_no	[5]
top_offset	[2219][0x8ab]



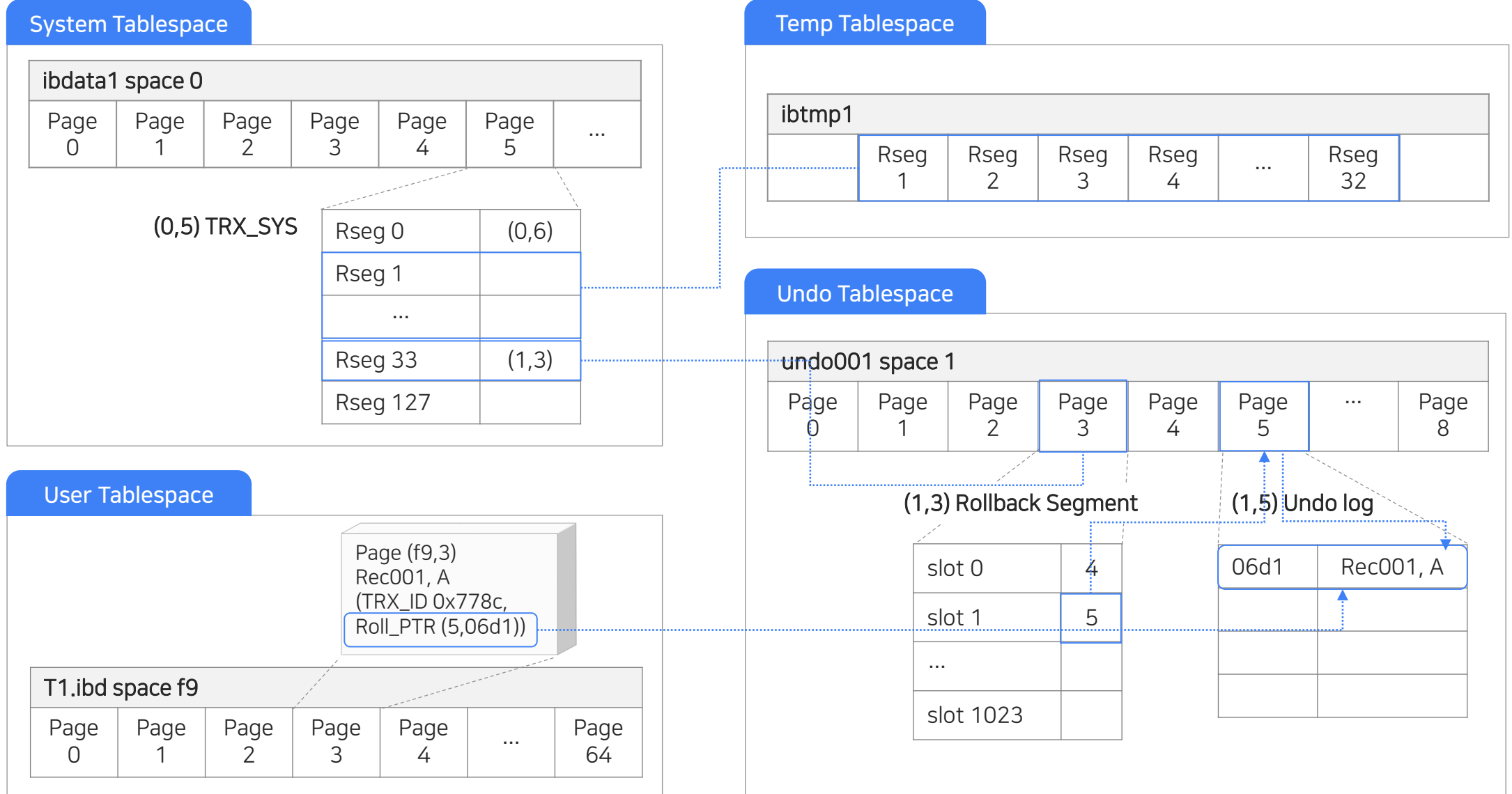
InnoDB 특징

- 1. 역사
- 2. 아키텍처
- 3. 메모리
- 4. 디스크
- 5. 성능

- 4.1 Redo Log File
- 4.2 Logical Structure
 - 4.2.1 Tablespace
 - 4.2.1.1 IBD File의 구조
 - 4.2.2 Segment, Extent
 - 4.2.3 Page
 - 4.2.3.1 Page Layout
 - 4.2.3.2 Directory Slot
 - 4.2.3.3 Page 내부 동작원리
- 4.3 Undo, MVCC Architecture
- 4.4 Purge
 - 4.4.1 Purge 정의 및 필요성
 - 4.4.2 Purge Update Undo
 - 4.4.3 Purge Garbage Space



Undo Architecture





Undo Architecture

Insert A 후, Select * from t1;

TRX_SYS
max_trx_id = 0x778c
rw_trx_list->count = 0
mvcc->m_views->count = 1

Buffer Pool

0x7f0208608000
Data Page (f9,3)
Rec001, A
(TRX_ID 0x778b,
Roll_PTR (4,0110))

User Tablespace

Page (f9,3)
Rec001, A
(TRX_ID 0x778b,
Roll_PTR (4,0110))

T1.ibd space f9

Page 0	Page 1	Page 2	Page 3	Page 4	...	Page 64
--------	--------	--------	--------	--------	-----	---------

System Tablespace

ibdata1 space 0

Page 0	Page 1	...	Page 5	...
--------	--------	-----	--------	-----

(0,5) TRX_SYS

Rseg 0	(0,6)
Rseg 1	
...	
Rseg 33	(1,3)
Rseg 127	

Undo Tablespace

undo001 space 1

Page 0	Page 1	Page 2	Page 3	Page 4	Page 5	...	Page 8
--------	--------	--------	--------	--------	--------	-----	--------

(1,3) Rollback Segment

(1,4) Undo log

slot 0	4
slot 1	
...	
slot 1023	

0110	Rec001, A



Undo Architecture

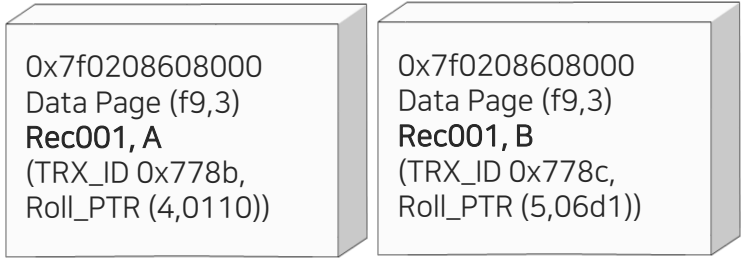
A를 B로 update (Update t1 Set c2 = B;)

TRX_SYS
max_trx_id = 0x778d
rw_trx_list->count = 0
mvcc->m_views->count = 1

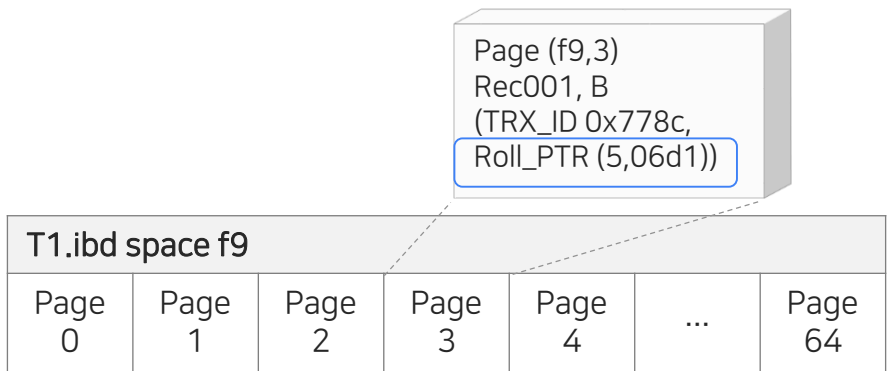


TRX_ID : 0x778c
addr=0x1b74a750
page_no=5
offset=0x6d1

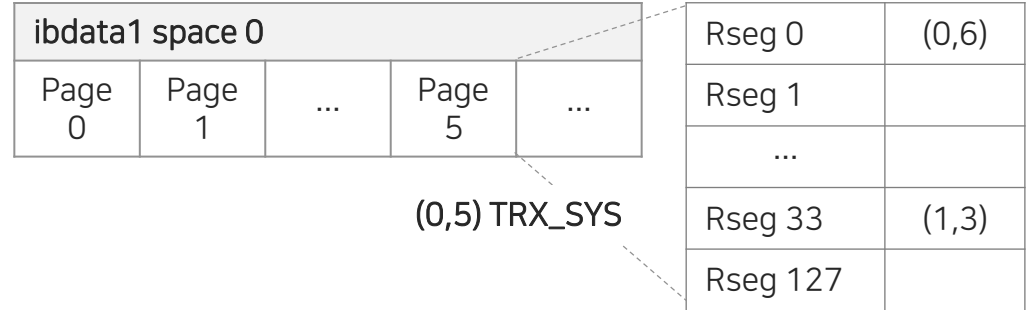
Buffer Pool



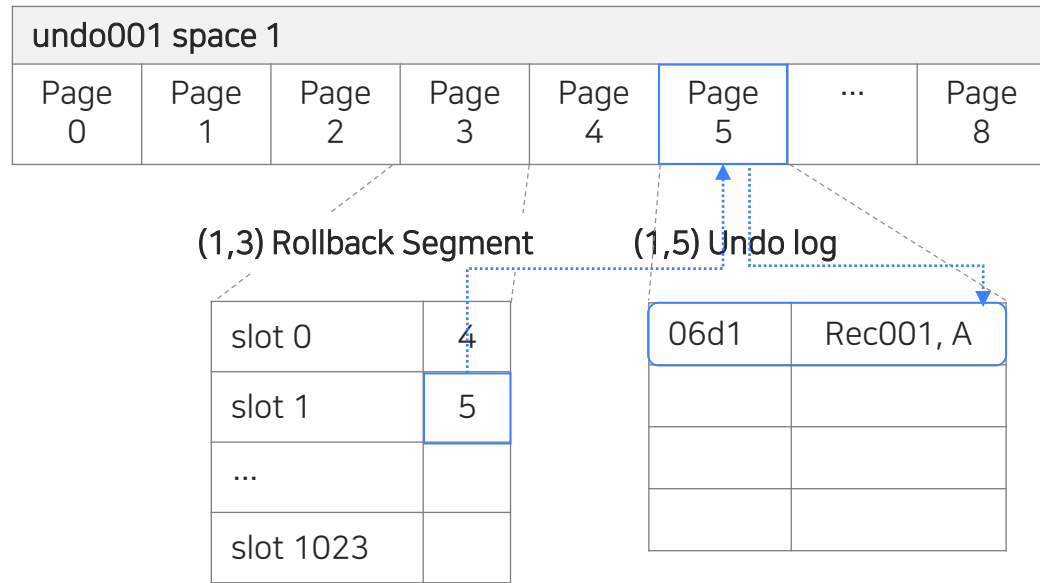
User Tablespace



System Tablespace

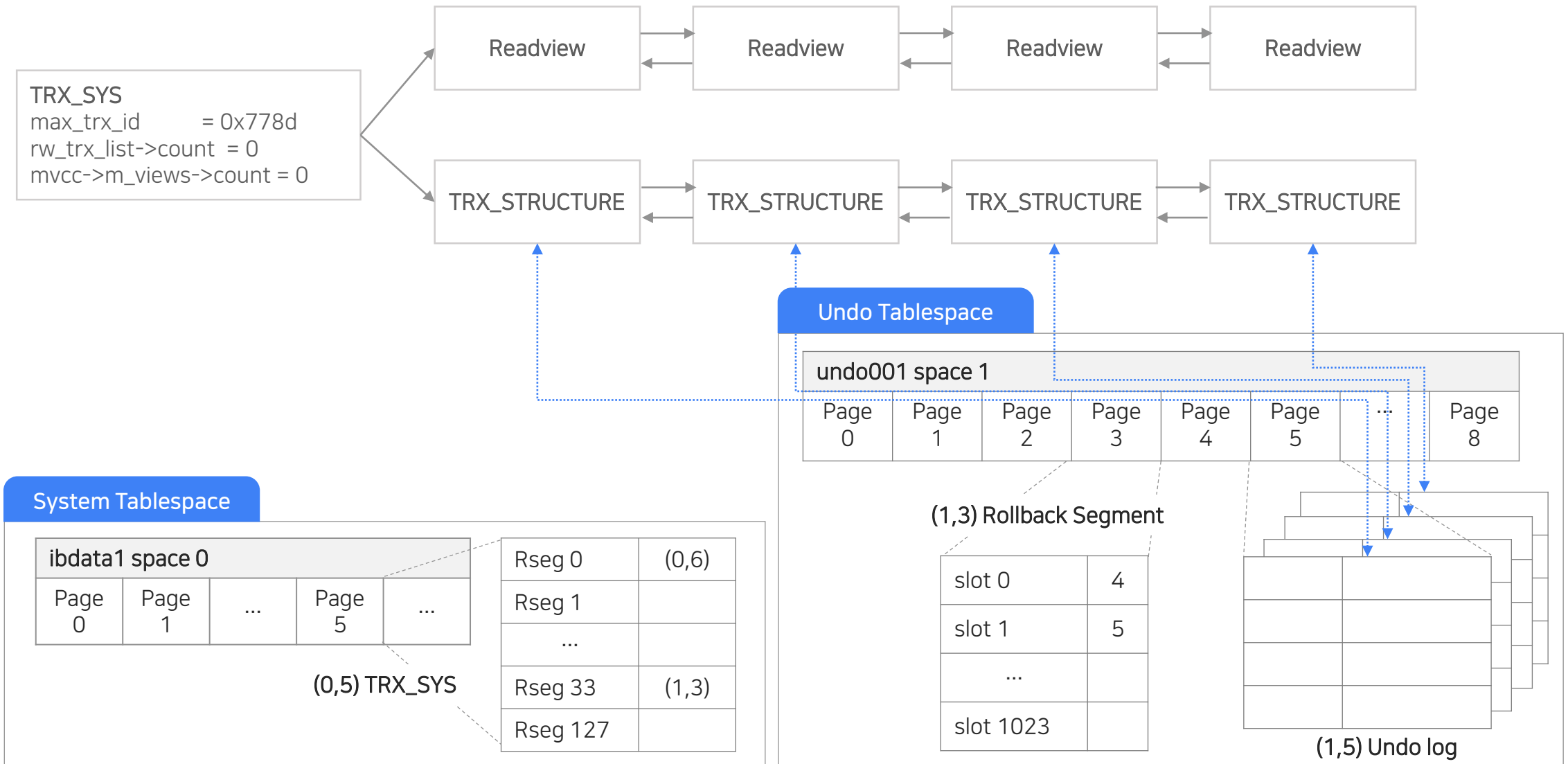


Undo Tablespace





MVCC Architecture





MVCC Architecture

1. 초기상태



Session 0

Insert into t1
values('Rec001','A');
Commit ;

TRX_SYS
max_trx_id = 0x778c
rw_trx_list->count = 0
mvcc->m_views->count = 0

Buffer Pool

System Tablespace

ibdata1 space 0

Page 0	Page 1	...	Page 5	...
--------	--------	-----	--------	-----

Rseg 0	(0,6)
Rseg 1	
...	
Rseg 33	(1,3)
Rseg 127	

User Tablespace

Undo Tablespace

Page (f9,3)
Rec001, A
(TRX_ID 0x778b,
Roll_PTR (4,110))

T1.ibd space f9

Page 0	...	Page 3	Page 4	...	Page 64
--------	-----	--------	--------	-----	---------

(1,3) Rollback Segment

slot 0	4
slot 1	
...	
slot 1023	

(1,4) Undo log

Undo Page (1,4)
Rec001, A
(TRX_ID 0x778b)



MVCC Architecture

2. 세션1 Select

Session 0

TRX_SYS
 max_trx_id = 0x778c
 rw_trx_list->count = 0
 mvcc->m_views->count = 1

Readview : 0x35dd108
 m_low_limit_no = 0x778c
 m_low_limit_id = 0x778c
 m_up_limit_id = 0x778c
 m_ids->m_size = 0

Session 1

Select * from t1;
 Rec001, A
 Readview :
 0x35dd108

Buffer Pool

0x7f0208608000
 Data Page (f9,3)
 Rec001, A
 (TRX_ID 0x778b,
 ROLL_PTR (4,110))

System Tablespace

ibdata1 space 0

Page 0	Page 1	...	Page 5	...
--------	--------	-----	--------	-----

Rseg 0	(0,6)
Rseg 1	
...	
Rseg 33	(1,3)
Rseg 127	

User Tablespace

Page (f9,3)
 Rec001, A
 (TRX_ID 0x778b,
 Roll_PTR (4,110))

T1.ibd space f9

Page 0	...	Page 3	Page 4	...	Page 64
--------	-----	--------	--------	-----	---------

Undo Tablespace

(1,3) Rollback Segment

slot 0	4
slot 1	
...	
slot 1023	



MVCC Architecture

3. 세션0 Update

Session 0
Update t1
Set c2 = B;

Session 1
Select * from t1;
Rec001, A
Readview :
0x35dd108

TRX_SYS
max_trx_id = 0x778d
rw_trx_list->count = 0
mvcc->m_views->count = 1

TRX ID 0x778c
addr=0x1b74a750

Readview : 0x35dd108
m_low_limit_no = 0x778c
m_low_limit_id = 0x778c
m_up_limit_id = 0x778c
m_ids->m_size = 0

Buffer Pool

0x7f02083b4000
Undo Page (1,5)
Rec001, A
(TRX_ID 0x778b)

0x7f0208608000
Data Page (f9,3)
Rec001, B
(TRX_ID 0x778c,
ROLL_PTR (5,6d1))

System Tablespace

ibdata1 space 0

Page 0	Page 1	...	Page 5	...
--------	--------	-----	--------	-----

Rseg 0	(0,6)
Rseg 1	
...	
Rseg 33	(1,3)
Rseg 127	

User Tablespace

Page (f9,3)
Rec001, B
(TRX_ID 0x778c)

T1.ibd space f9

Page 0	...	Page 3	Page 4	...	Page 64
--------	-----	--------	--------	-----	---------

Undo Tablespace

(1,3) Rollback Segment

slot 0	4
slot 1	5
...	
slot 1023	

(1,5) Undo log

Undo Page (1,5)
(0x6d1)
Rec001, A
(TRX_ID 0x778b)



MVCC Architecture

4. 세션2 Select

Session 0

TRX_SYS
 max_trx_id = 0x778e
 rw_trx_list->count = 0
 mvcc->m_views->count = 2
 m_views->start = 0x35dd068

Readview : 0x35dd068
 m_low_limit_no = 0x778e
 m_low_limit_id = 0x778e
 m_up_limit_id = 0x778e
 m_ids->m_size = 0

Readview : 0x35dd108
 m_low_limit_no = 0x778c
 m_low_limit_id = 0x778c
 m_up_limit_id = 0x778c
 m_ids->m_size = 0

Session 1

Select * from t1;
 Rec001, A
 Readview :
 0x35dd108

Buffer Pool

0x7f02083b4000
 Undo Page (1,5)
 Rec001, A
 (TRX_ID 0x778b)

0x7f0208608000
 Data Page (f9,3)
 Rec001, B
 (TRX_ID 0x778c,
 ROLL_PTR (5,6d1))

System Tablespace

ibdata1 space 0

Page 0	Page 1	...	Page 5	...
--------	--------	-----	--------	-----

Rseg 0	(0,6)
Rseg 1	
...	
Rseg 33	(1,3)
Rseg 127	

Session 2

Select * from t1;
 Rec001, B
 Readview :
 0x35dd068

User Tablespace

Page (f9,3)
 Rec001, B
 (TRX_ID 0x778c)

T1.ibd space f9

Page 0	...	Page 3	Page 4	...	Page 64
--------	-----	--------	--------	-----	---------

Undo Tablespace

(1,3) Rollback Segment

slot 0	4
slot 1	5
...	
slot 1023	

(1,5) Undo log

Undo Page (1,5)
 (0x6d1)
 Rec001, A
 (TRX_ID 0x778b)



MVCC Architecture

5. 세션0 Update

Session 0
Update t1
set c2 = C;

TRX_SYS
max_trx_id = 0x778e
rw_trx_list->count = 0
mvcc->m_views->count = 2
m_views->start = 0x35dd068

TRX ID 0x778e
addr=0x1b74a750

Readview : 0x35dd068
m_low_limit_no = 0x778e
m_low_limit_id = 0x778e
m_up_limit_id = 0x778e
m_ids->m_size = 0

Readview : 0x35dd108
m_low_limit_no = 0x778c
m_low_limit_id = 0x778c
m_up_limit_id = 0x778c
m_ids->m_size = 0

Session 1
Select * from t1;
Rec001, A
Readview :
0x35dd108

Buffer Pool

0x7f02083b4000 Undo Page (1,5) (0x6d1) Rec001, A (TRX_ID 0x778b)	0x7f02083b4000 Undo Page (1,5) (0x7be) Rec001, B (TRX_ID 0x778c ROLL_PTR (5,6d1))	0x7f0208608000 Data Page (f9,3) Rec001, B (TRX_ID 0x778c, ROLL_PTR (5,6d1))
---	---	--

System Tablespace

ibdata1 space 0

Page 0	Page 1	...	Page 5	...
--------	--------	-----	--------	-----

Rseg 0	(0,6)
Rseg 1	
...	
Rseg 33	(1,3)
Rseg 127	

Session 2
Select * from t1;
Rec001, B
Readview :
0x35dd068

User Tablespace

Page (f9,3)
Rec001, C
(TRX_ID 0x778e)

T1.ibd space f9

Page 0	...	Page 3	Page 4	...	Page 64
--------	-----	--------	--------	-----	---------

Undo Tablespace

(1,3) Rollback Segment

slot 0	4
slot 1	5
...	
slot 1023	

(1,5) Undo log

Undo Page (1,5)
(0x7be)
Rec001, B
(TRX_ID 0x778c
ROLL_PTR (5,6d1))



MVCC Architecture

6. 세션3 Select

Session 0

Session 1

Select * from t1;
 Rec001, A
 Readview :
 0x35dd108

Session 2

Select * from t1;
 Rec001, B
 Readview :
 0x35dd068

Session 2

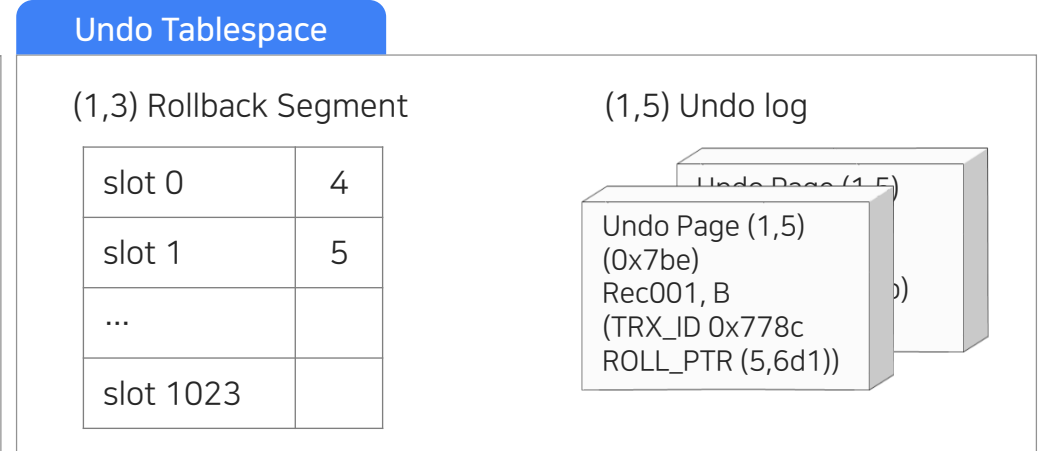
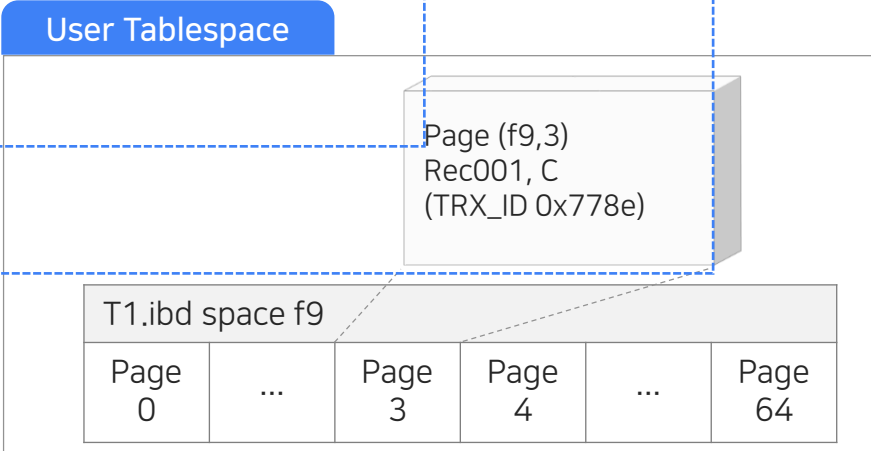
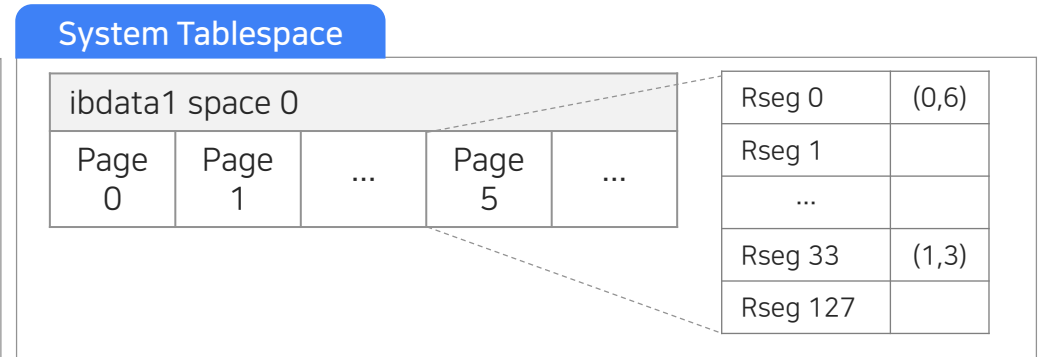
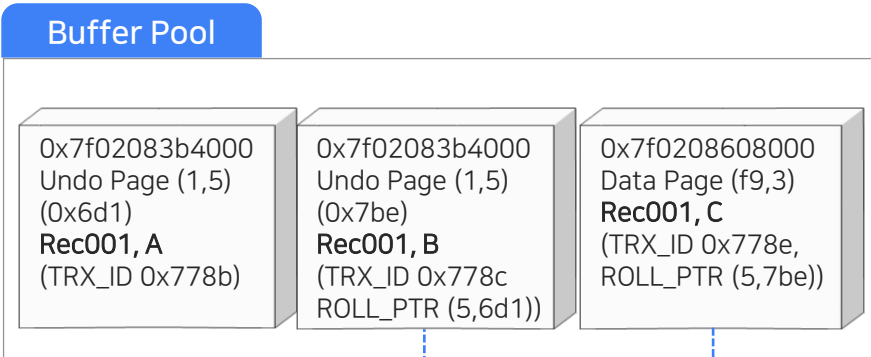
Select * from t1;
 Rec001, C
 Readview :
 0x35ddfc8

TRX_SYS
 max_trx_id = 0x7790
 rw_trx_list->count = 0
 mvcc->m_views->count = 3
 m_views->start = 0x35ddfc8

Readview : 0x35dcfc8
 m_low_limit_no = 0x7790
 m_low_limit_id = 0x7790
 m_up_limit_id = 0x7790
 m_ids->m_size = 0

Readview : 0x35dd068
 m_low_limit_no = 0x778e
 m_low_limit_id = 0x778e
 m_up_limit_id = 0x778e
 m_ids->m_size = 0

Readview : 0x35dd108
 m_low_limit_no = 0x778c
 m_low_limit_id = 0x778c
 m_up_limit_id = 0x778c
 m_ids->m_size = 0





Readview

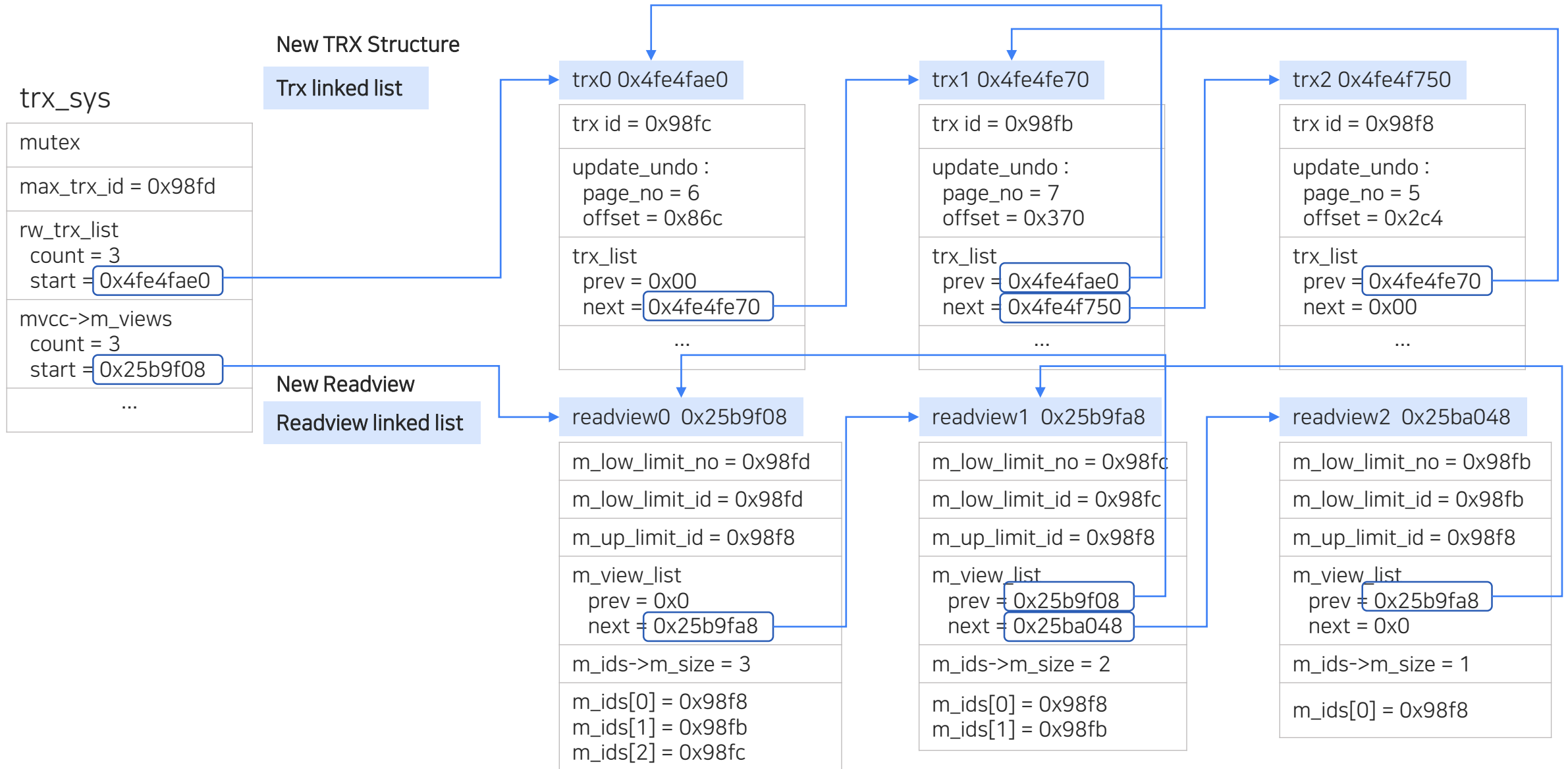
정의 및 목적

- 트랜잭션이 발생하면, COMMIT 시점이 아닌 트랜잭션 시작 시점의 TRX_ID 가 저장됨
- 쿼리 시작시점의 액티브 트랜잭션들의 집합 (COMMIT/ROLLBACK 하지 않은 상태)
- MVCC 구현에 반드시 필요

Component	Description
low_limit_no	- 해당 값을 Purge thread가 참조하여 불필요한 Undo log들을 처리하는 purge 작업에 사용된다. - low_limit_no = Readview 생성시 trx_sys의 max_trx_id값
low_limit_id	- low_limit_id보다 크거나 같은 TRX_ID ($trx_id \geq low_limit_id$) 는 해당 readview에서 조회 안 됨 - low_limit_id = Readview 생성시 trx_sys의 max_trx_id값
up_limit_id	- up_limit_id($trx_id < up_limit_id$) 보다 작은 트랜잭션 TRX_ID 는 해당 readview에서 모두 조회됨 - up_limit_id = readview 생성시, active 트랜잭션 중 가장 작은 TRX_ID
trx_ids[]	- 모든 액티브 트랜잭션 id로 구성된 그룹
view_list	- Readview list (doubly linked list)



Readview





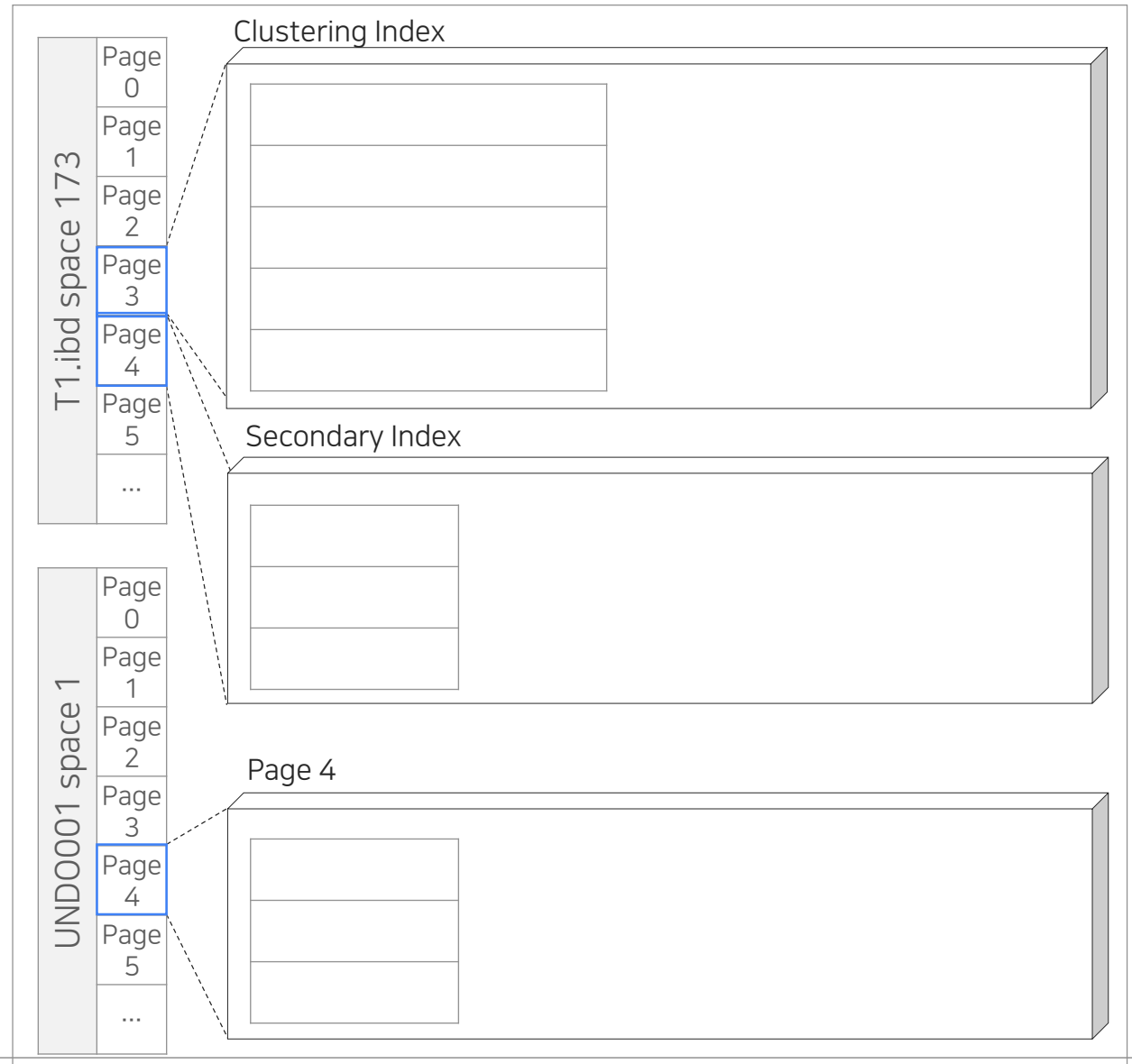
Readview

1. t1 Table 생성

create table t1 (id int primary key, comment char(50), filler char(2)) engine = innodb;

2. Secondary index 생성

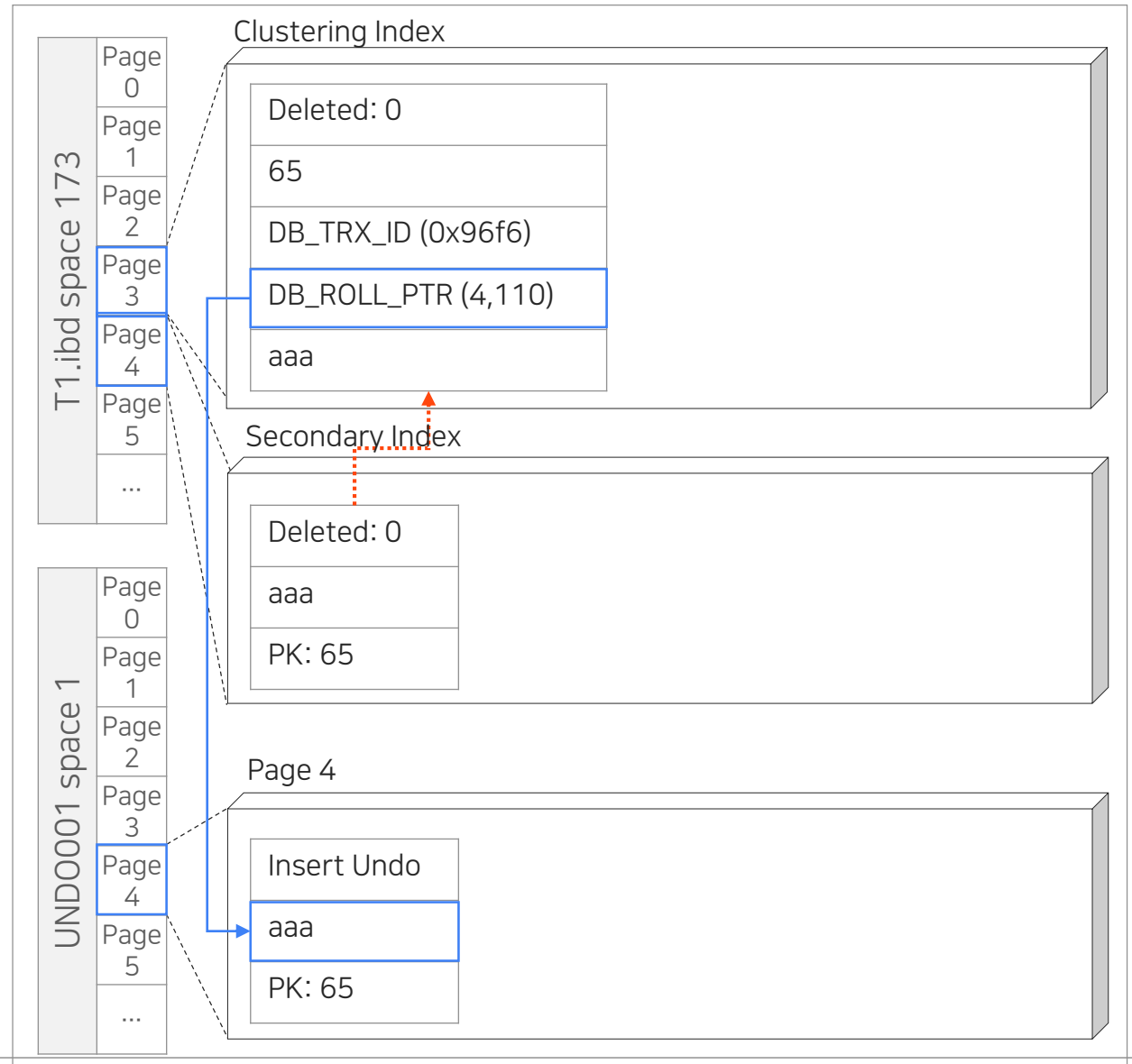
create index test_idx on test(comment);





Readview

3. TRX_ID가 0x96f6 일때, t1에 1건 Insert
insert into t.test values (65, 'aaa',' ');



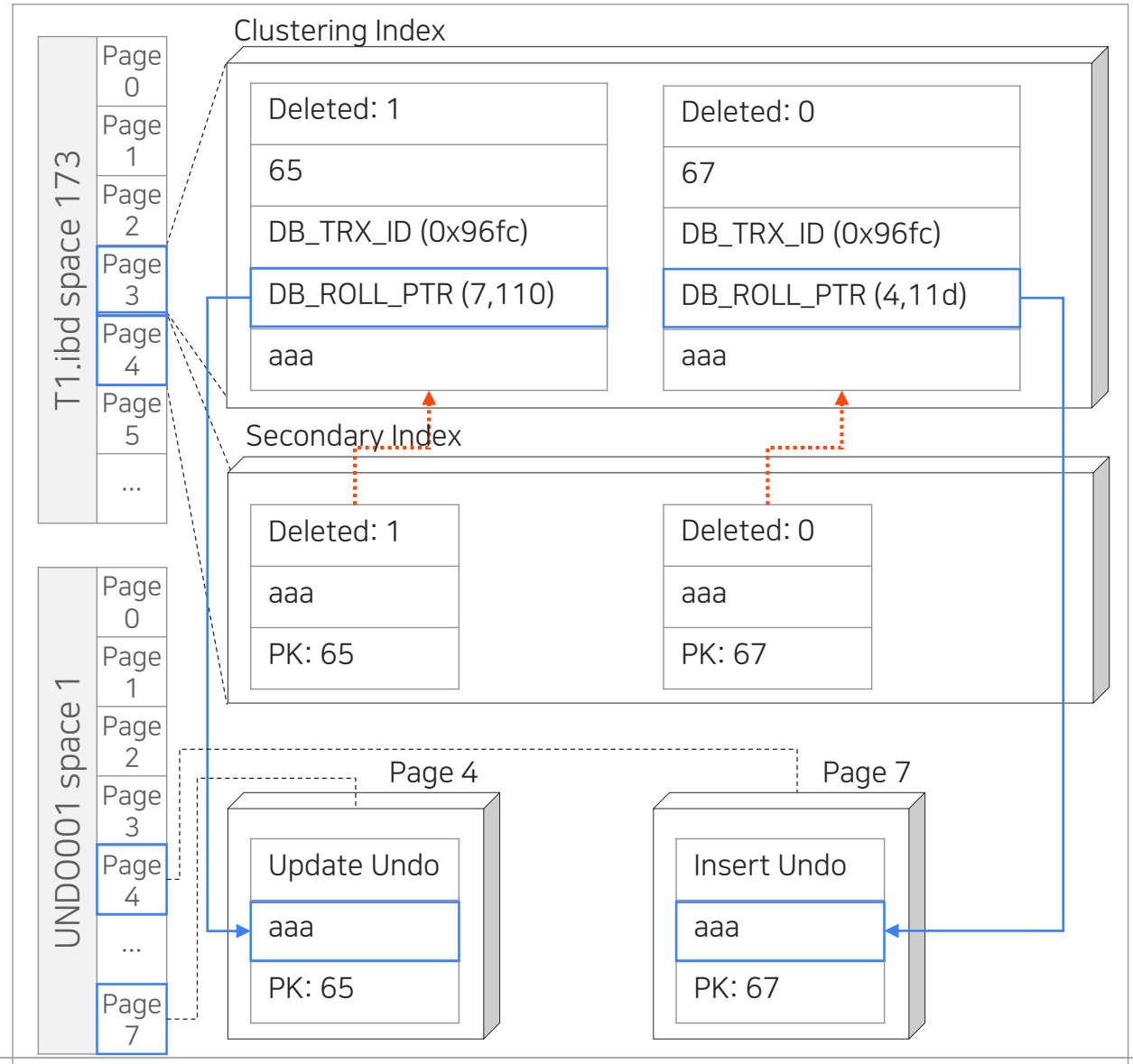


Readview

4. Primary key 업데이트

update t.test set id = 67 where id = 65 ;

- 새로운 레코드 (67, aaa)가 추가됨
(Insert 발생하는 것처럼 Insert Undo에 저장)
- 이전 레코드 (65, aaa)는 삭제 표시됨 (삭제 flag, trx_id, roll_ptr를 설정하고, before-image를 Update Undo에 저장)
- Secondary Index 페이지에서도 레코드 (aaa, 65) 는 삭제 설정하고 레코드 (aaa,67)가 추가됨



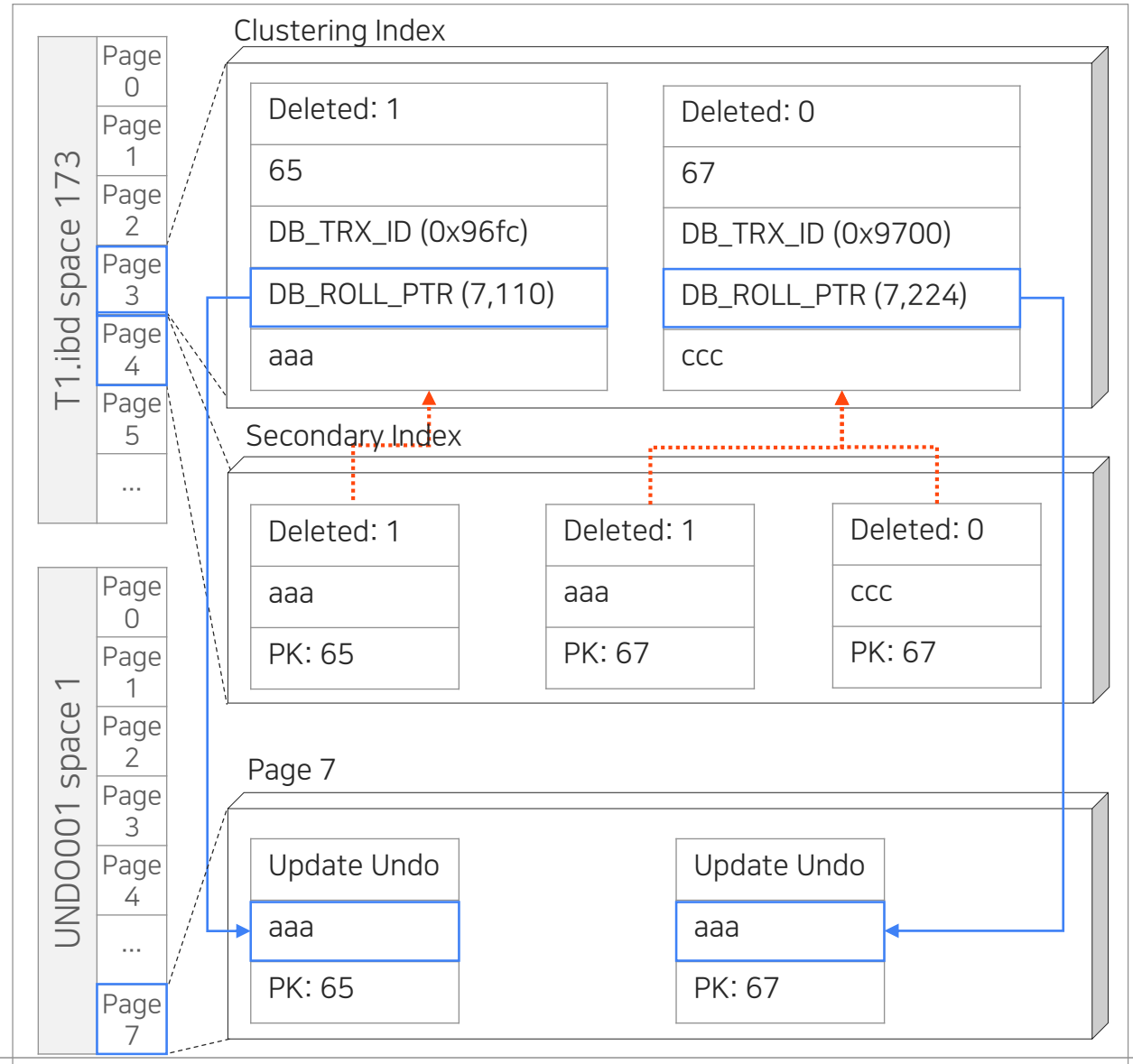


Readview

5. Secondary key 업데이트

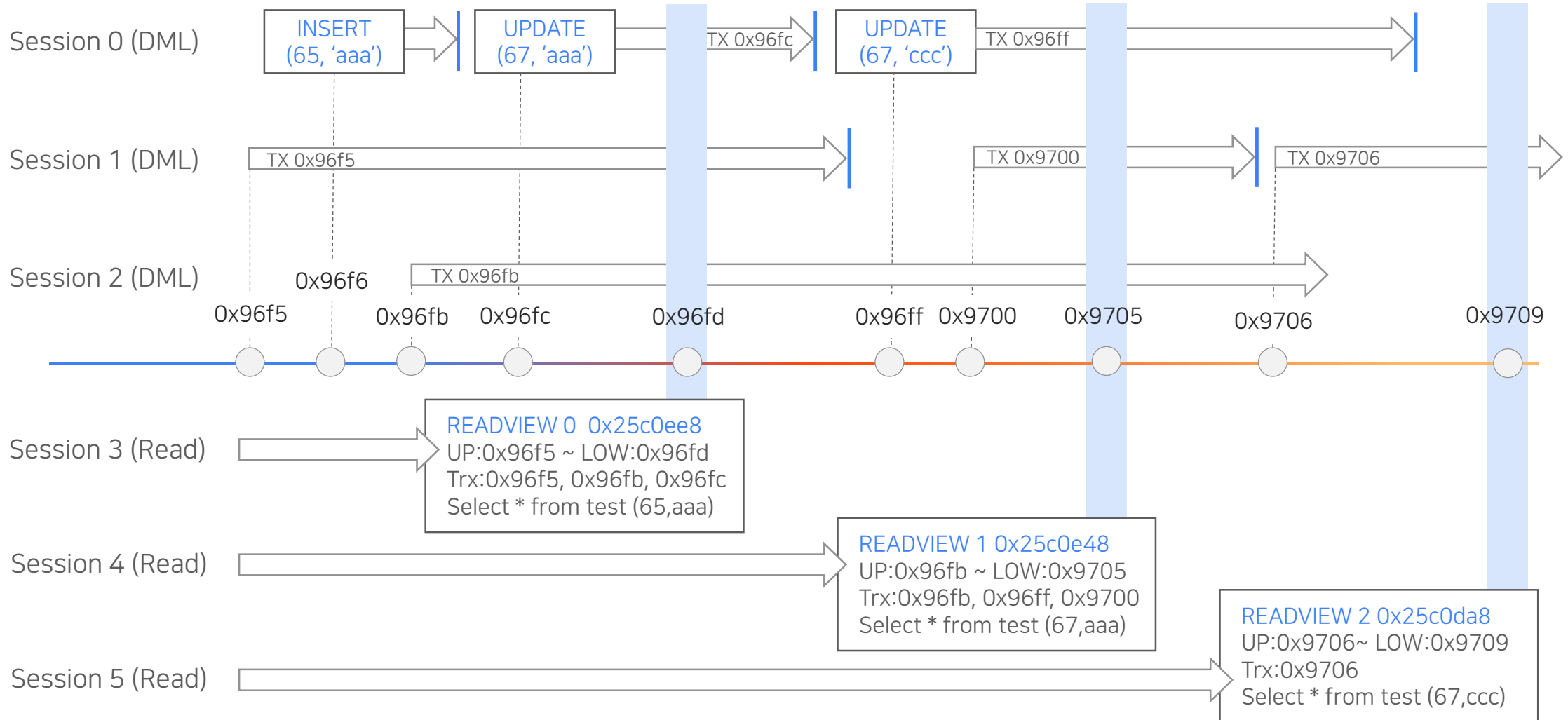
update t.test set comment = 'ccc' where id = 67 ;

- 레코드 (67,aaa)를 레코드 (67,ccc)로 변경하고, before-image를 Update Undo에 저장
- Secondary Index 페이지에서도 레코드 (aaa,67)는 삭제 설정되고, 레코드 (ccc,67)가 추가됨.





Readview





Readview

select * from t1;

	case	up_limit_id	low_limit_id	trx_ids []	query result
Readview0	Primary Key 변경	0x96f5	0x96fd	0x96f5, 0x96fb, 0x96fc	(65, aaa)
Readview1	Secondary Key 변경	0x96fb	0x9705	0x96fb, 0x96ff, 0x9700	(67, aaa)
Readview2	해당 테이블에 트랜잭션이 없을 경우	0x9706	0x9709	0x9706	(67, ccc)



InnoDB 특징

- 1. 역사
- 2. 아키텍처
- 3. 메모리
- 4. 디스크
- 5. 성능

- 4.1 Redo Log File
- 4.2 Logical Structure
 - 4.2.1 Tablespace
 - 4.2.1.1 IBD File의 구조
 - 4.2.2 Segment, Extent
 - 4.2.3 Page
 - 4.2.3.1 Page Layout
 - 4.2.3.2 Directory Slot
 - 4.2.3.3 Page 내부 동작원리
- 4.3 Undo, MVCC Architecture
- 4.4 Purge
 - 4.4.1 Purge 정의 및 필요성
 - 4.4.2 Purge Update Undo
 - 4.4.3 Purge Garbage Space



Purge 정의 및 필요성

1. Purge의 정의

데이터의 변경이 발생하였을 때, 다른 트랜잭션의 읽기 일관성(MVCC)을 위해 변경 전의 데이터를 롤백 세그먼트에 기록해둔다. 이 데이터가 더 이상 필요하지 않게 되면 해당 **세그먼트를 재사용하기 위해 청소**하는 것을 의미한다.

2. Purge의 필요성

- History list가 길어져 before-image가 필요한 쿼리의 성능 저하를 방지하기 위해서
- 변경 및 삭제된 자료들이 차지하고 있는 **디스크 공간을 확보**하기 위해



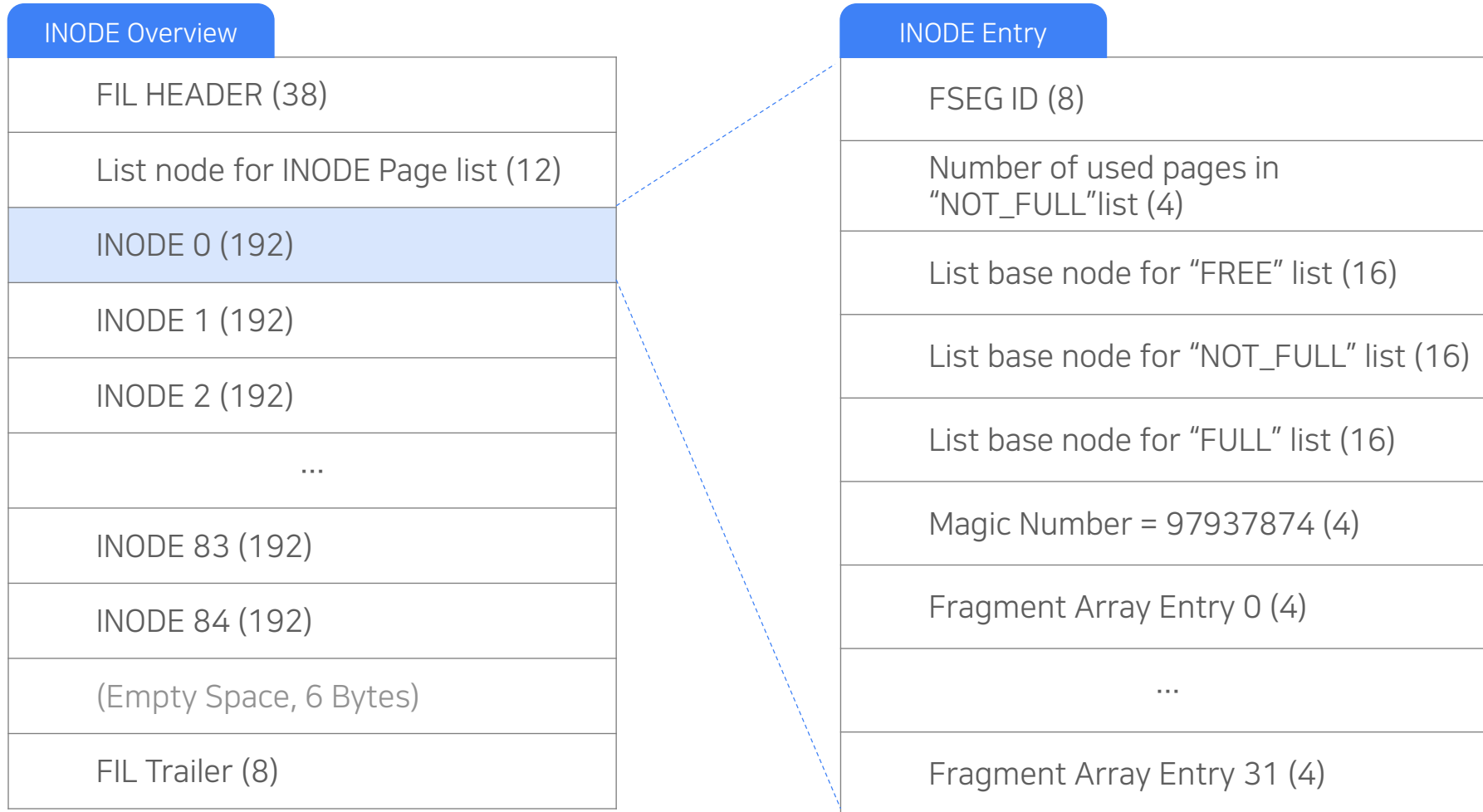
InnoDB 특징

- 1. 역사
- 2. 아키텍처
- 3. 메모리
- 4. 디스크
- 5. 성능

- 4.1 Redo Log File
- 4.2 Logical Structure
 - 4.2.1 Tablespace
 - 4.2.1.1 IBD File의 구조
 - 4.2.2 Segment, Extent
 - 4.2.3 Page
 - 4.2.3.1 Page Layout
 - 4.2.3.2 Directory Slot
 - 4.2.3.3 Page 내부 동작원리
- 4.3 Undo, MVCC Architecture
- 4.4 Purge
 - 4.4.1 Purge 정의 및 필요성
 - 4.4.2 Purge Update Undo
 - 4.4.3 Purge Garbage Space



Purge Update Undo





Purge Update Undo

- INODE Entry

테이블스페이스 내의 세그먼트 정보를 얻기 위해 사용

Element	Description
File Segment ID	- 파일 세그먼트의 ID값 - 값이 0 이면, entry를 사용하지 않았다는 의미
Number of used pages in the NOT_NULL list	- File Space Header의 FREE_FRAG 리스트와 비슷함 (*FREE_FRAG : fragments에 사용된다고 할당된 extent에서 아직 사용하지 않은 페이지들을 가진 extent) - NOT_FULL 리스트에서 사용하는 page number를 기록
FREE	- 파일 세그먼트에 할당되었지만, 한번도 사용하지 않은 extent를 가진 리스트
NOT_FULL	- 파일 세그먼트에 할당된 것 중 적어도 하나는 사용한 extent를 가진 리스트 - 마지막에 남은 free page가 사용되면, 이 extent는 FULL 리스트로 넘어감
FULL	- File segment에 할당된 페이지를 전부 사용하고 있는 extent를 가진 리스트 - 페이지 하나라도 free 상태로 변경된다면, 이 extent는 NOT_FULL 리스트로 넘어감
Magic Number	- Marker로서 97937874라는 값이 저장되는데, 이것은 file segment INODE의 초기화를 의미
Fragment Array	- FREE_FRAG 또는 FULL_FRAG에 있는 extent로 부터 개별적으로 할당된 페이지들의 array (*FULL_FRAG : 남아있는 free page가 없는 경우에 위치)



Purge Update Undo



show engine innodb status;
Trx id counter : 21430
History list length : 0

User Tablespace

T1.ibd space 8f					
Page 0	Page 1	Page 2	Page 3	...	Page 64

Page (8f,3)
Rec0000, A, 199
Rec0010, A, 199
Rec0020, A, 199
Rec0030, A, 199
Rec0040, A, 199

Undo Tablespace

undo001 space 1							
Page 0	Page 1	Page 2	Page 3	Page 4	Page 5	...	Page 8

(1,2) INODE


FSEG_ID	PAGE_NO
00 00 00 01	00 00 00 03
00 00 00 04	00 00 00 04

(1,3) Rollback Segment

HISTORY_SIZE = 0	
First Page No/Offset = FF/0	
Last Page No/Offset = FF/0	
slot 0	00 00 00 04
slot 1	
...	
slot 1023	



Purge Update Undo

 show engine innodb status;
Trx id counter : 21430
History list length : 0

Begin;
Update t1 set c3=c3+1;

User Tablespace

T1.ibd space 8f					
Page 0	Page 1	Page 2	Page 3	...	Page 64

Page (8f,3)
Rec0000, A, 200
Rec0010, A, 200
Rec0020, A, 200
Rec0030, A, 200
Rec0040, A, 200

Undo Tablespace

undo001 space 1							
Page 0	Page 1	Page 2	Page 3	Page 4	Page 5	...	Page 8

(1,2) INODE

FSEG_ID	PAGE_NO
00 00 00 01	00 00 00 03
00 00 00 04	00 00 00 04
00 00 1a d2	00 00 00 05


(1,3) Rollback Segment

HISTORY_SIZE = 0	
First Page No/Offset = FF/0	
Last Page No/Offset = FF/0	
slot 0	00 00 00 04
slot 1	00 00 00 05
...	
slot 1023	

Page (1,5)
Trx_id : 0x53b6
Trx_no : 0x00
Rec0000, A, 199



Purge Update Undo



show engine innodb status;
Trx id counter : 21432
History list length : 1

Commit;

User Tablespace

T1.ibd space 8f					
Page 0	Page 1	Page 2	Page 3	...	Page 64

Page (8f,3)
Rec0000, A, 200
Rec0010, A, 200
Rec0020, A, 200
Rec0030, A, 200
Rec0040, A, 200

Undo Tablespace

undo001 space 1							
Page 0	Page 1	Page 2	Page 3	Page 4	Page 5	...	Page 8

(1,2) INODE

FSEG_ID	PAGE_NO
00 00 00 01	00 00 00 03
00 00 00 04	00 00 00 04
00 00 1a d2	00 00 00 05

(1,3) Rollback Segment

HISTORY_SIZE = 1	
First Page No/Offset = 05/78	
Last Page No/Offset = 05/78	
slot 0	00 00 00 04
slot 1	
...	
slot 1023	

Page (1,5)
Trx_id : 0x53b6
Trx_no : 0x53b7
Rec0000, A, 199



Purge Update Undo



show engine innodb status;
Trx id counter : 21432
History list length : 1

Begin;
Update t1 set c3=c3+1;

User Tablespace

T1.ibd space 8f					
Page 0	Page 1	Page 2	Page 3	...	Page 64

Page (8f,3)
 Rec0000, A, 201
 Rec0010, A, 201
 Rec0020, A, 201
 Rec0030, A, 201
 Rec0040, A, 201

Undo Tablespace

undo001 space 1							
Page 0	Page 1	Page 2	Page 3	Page 4	Page 5	Page 6	...

(1,2) INODE

FSEG_ID	PAGE_NO
00 00 00 01	00 00 00 03
00 00 00 04	00 00 00 04
00 00 1a d2	00 00 00 05
00 00 1a d3	00 00 00 06


(1,3) Rollback Segment

HISTORY_SIZE = 1	
First Page No/Offset = 05/78	
Last Page No/Offset = 05/78	
slot 0	00 00 00 04
slot 1	00 00 00 06
...	
slot 1023	

Page (1,6)
 Trx_id : 0x53b8
 Trx_no : 0x00
 Rec0000, A, 200



Purge Update Undo



show engine innodb status;
Trx id counter : 21434
History list length : 2

Commit;

User Tablespace

T1.ibd space 8f					
Page 0	Page 1	Page 2	Page 3	...	Page 64
<div data-bbox="468 968 820 1253" style="border: 1px solid gray; padding: 5px; width: fit-content; margin: auto;"> Page (8f,3) Rec0000, A, 201 Rec0010, A, 201 Rec0020, A, 201 Rec0030, A, 201 Rec0040, A, 201 </div>					

Undo Tablespace

undo001 space 1																									
Page 0	Page 1	Page 2	Page 3	Page 4	Page 5	Page 6	...																		
<div style="display: flex; justify-content: space-around;"> <div data-bbox="1085 582 1513 982" style="border: 1px solid gray; padding: 5px;"> <h3>(1,2) INODE</h3> <table border="1"> <thead> <tr> <th>FSEG_ID</th> <th>PAGE_NO</th> </tr> </thead> <tbody> <tr><td>00 00 00 01</td><td>00 00 00 03</td></tr> <tr><td>00 00 00 04</td><td>00 00 00 04</td></tr> <tr><td>00 00 1a d2</td><td>00 00 00 05</td></tr> <tr><td>00 00 1a d3</td><td>00 00 00 06</td></tr> </tbody> </table> </div> <div data-bbox="1544 582 2007 1053" style="border: 1px solid gray; padding: 5px;"> <h3>(1,3) Rollback Segment</h3> <p>HISTORY_SIZE = 2 First Page No/Offset = 06/78 Last Page No/Offset = 05/78</p> <table border="1"> <tr><td>slot 0</td><td>00 00 00 04</td></tr> <tr><td>slot 1</td><td></td></tr> <tr><td>...</td><td></td></tr> <tr><td>slot 1023</td><td></td></tr> </table> </div> <div data-bbox="2023 642 2339 865" style="border: 1px solid gray; padding: 5px; width: fit-content;"> <p>Page (1,6) Trx_id : 0x53b8 Trx_no : 0x53b9 Rec0000, A, 200</p> </div> </div>								FSEG_ID	PAGE_NO	00 00 00 01	00 00 00 03	00 00 00 04	00 00 00 04	00 00 1a d2	00 00 00 05	00 00 1a d3	00 00 00 06	slot 0	00 00 00 04	slot 1		...		slot 1023	
FSEG_ID	PAGE_NO																								
00 00 00 01	00 00 00 03																								
00 00 00 04	00 00 00 04																								
00 00 1a d2	00 00 00 05																								
00 00 1a d3	00 00 00 06																								
slot 0	00 00 00 04																								
slot 1																									
...																									
slot 1023																									



Purge Update Undo



show engine innodb status;
Trx id counter : 21434
History list length : 2

Begin;
Update t1 set c3=c3+1;

User Tablespace

T1.ibd space 8f					
Page 0	Page 1	Page 2	Page 3	...	Page 64

Page (8f,3)
 Rec0000, A, 202
 Rec0010, A, 202
 Rec0020, A, 202
 Rec0030, A, 202
 Rec0040, A, 202

Undo Tablespace

undo001 space 1							
Page 0	Page 1	Page 2	Page 3	Page 4	Page 5	...	Page 7

(1,2) INODE

FSEG_ID	PAGE_NO
00 00 00 01	00 00 00 03
00 00 00 04	00 00 00 04
00 00 1a d2	00 00 00 05
00 00 1a d3	00 00 00 06
00 00 1a d4	00 00 00 07


(1,3) Rollback Segment

HISTORY_SIZE = 2 First Page No/Offset = 06/78 Last Page No/Offset = 05/78	
slot 0	00 00 00 04
slot 1	00 00 00 07
...	
slot 1023	

Page (1,7)
 Trx_id : 0x53ba
 Trx_no : 0x00
 Rec0000, A, 201



Purge Update Undo



show engine innodb status;
Trx id counter : 21436
History list length : 3

Commit;

User Tablespace

T1.ibd space 8f					
Page 0	Page 1	Page 2	Page 3	...	Page 64

Page (8f,3)
Rec0000, A, 202
Rec0010, A, 202
Rec0020, A, 202
Rec0030, A, 202
Rec0040, A, 202

Undo Tablespace

undo001 space 1							
Page 0	Page 1	Page 2	Page 3	Page 4	Page 5	...	Page 7

(1,2) INODE

FSEG_ID	PAGE_NO
00 00 00 01	00 00 00 03
00 00 00 04	00 00 00 04
00 00 1a d2	00 00 00 05
00 00 1a d3	00 00 00 06
00 00 1a d4	00 00 00 07


(1,3) Rollback Segment

HISTORY_SIZE = 3	
First Page No/Offset = 07/78	
Last Page No/Offset = 05/78	
slot 0	00 00 00 04
slot 1	
...	
slot 1023	

Page (1,7)
Trx_id : 0x53ba
Trx_no : 0x53bb
Rec0000, A, 201



Purge Update Undo



show engine innodb status;
Trx id counter : 21520
History list length : 45

Begin;
Update t1 set c3=c3+1;

User Tablespace

T1.ibd space 8f					
Page 0	Page 1	Page 2	Page 3	...	Page 64
<div style="border: 1px solid black; padding: 5px; width: fit-content; margin: auto;"> Page (8f,3) Rec0000, A, 245 Rec0010, A, 245 Rec0020, A, 245 Rec0030, A, 245 Rec0040, A, 245 </div>					


Undo Tablespace

undo001 space 1																																	
Page 0	Page 1	Page 2	Page 3	Page 4	Page 5	...	Page 32																										
<div style="display: flex; justify-content: space-around;"> <div style="text-align: center;"> <p>(1,2) INODE</p> <table border="1"> <thead> <tr> <th>FSEG_ID</th> <th>PAGE_NO</th> </tr> </thead> <tbody> <tr><td>00 00 00 01</td><td>00 00 00 03</td></tr> <tr><td>00 00 00 04</td><td>00 00 00 04</td></tr> <tr><td>00 00 1a d2</td><td>00 00 00 05</td></tr> <tr><td>...</td><td>...</td></tr> <tr><td>00 00 1a ff</td><td>00 00 00 32</td></tr> </tbody> </table> </div> <div style="text-align: center;"> <p>(1,3) Rollback Segment</p> <table border="1"> <thead> <tr> <th colspan="2">HISTORY_SIZE = 2d</th> </tr> </thead> <tbody> <tr> <td colspan="2">First Page No/Offset = 31/78</td> </tr> <tr> <td colspan="2">Last Page No/Offset = 05/78</td> </tr> <tr> <td>slot 0</td> <td>00 00 00 04</td> </tr> <tr> <td>slot 1</td> <td>00 00 00 32</td> </tr> <tr> <td>...</td> <td></td> </tr> <tr> <td>slot 1023</td> <td></td> </tr> </tbody> </table> </div> </div>								FSEG_ID	PAGE_NO	00 00 00 01	00 00 00 03	00 00 00 04	00 00 00 04	00 00 1a d2	00 00 00 05	00 00 1a ff	00 00 00 32	HISTORY_SIZE = 2d		First Page No/Offset = 31/78		Last Page No/Offset = 05/78		slot 0	00 00 00 04	slot 1	00 00 00 32	...		slot 1023	
FSEG_ID	PAGE_NO																																
00 00 00 01	00 00 00 03																																
00 00 00 04	00 00 00 04																																
00 00 1a d2	00 00 00 05																																
...	...																																
00 00 1a ff	00 00 00 32																																
HISTORY_SIZE = 2d																																	
First Page No/Offset = 31/78																																	
Last Page No/Offset = 05/78																																	
slot 0	00 00 00 04																																
slot 1	00 00 00 32																																
...																																	
slot 1023																																	

Page (1,32)
 Trx_id : 0x5410
 Trx_no : 0x00
 Rec0000, A, 244



Purge Update Undo

 show engine innodb status;
Trx id counter : 21522
History list length : 0

Commit;

User Tablespace

T1.ibd space 8f

Page 0	Page 1	Page 2	Page 3	...	Page 64
--------	--------	--------	--------	-----	---------

Page (8f,3)
Rec0000, A, 245
Rec0010, A, 245
Rec0020, A, 245
Rec0030, A, 245
Rec0040, A, 245

Undo Tablespace

undo001 space 1

Page 0	Page 1	Page 2	Page 3	Page 4	Page 5	...	Page 32
--------	--------	--------	--------	--------	--------	-----	---------

(1,2) INODE

FSEG_ID	PAGE_NO
00 00 00 01	00 00 00 03
00 00 00 04	00 00 00 04

(1,3) Rollback Segment

HISTORY_SIZE = 0	
First Page No/Offset = FF/0	
Last Page No/Offset = FF/0	
slot 0	00 00 00 04
slot 1	
...	
slot 1023	

Page (1,32)
Trx_id : 0x5410
Trx_no : 0x5411
Rec0000, A, 244

- Update Undo가 Purge 됨
- History list length : 2d → 0
- First/Last Page 정보 초기화
- INODE : 05 ~ 32 페이지 정보 초기화



InnoDB 특징

- 1. 역사
- 2. 아키텍처
- 3. 메모리
- 4. 디스크
- 5. 성능

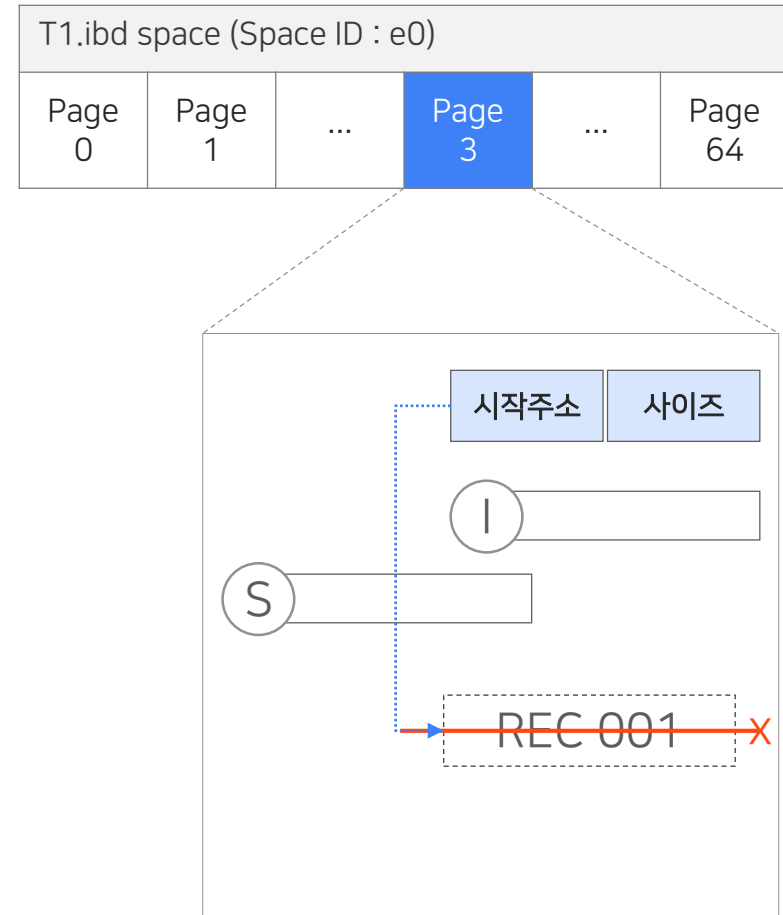
- 4.1 Redo Log File
- 4.2 Logical Structure
 - 4.2.1 Tablespace
 - 4.2.1.1 IBD File의 구조
 - 4.2.2 Segment, Extent
 - 4.2.3 Page
 - 4.2.3.1 Page Layout
 - 4.2.3.2 Directory Slot
 - 4.2.3.3 Page 내부 동작원리
- 4.3 Undo, MVCC Architecture
- 4.4 Purge
 - 4.4.1 Purge 정의 및 필요성
 - 4.4.2 Purge Update Undo
 - 4.4.3 Purge Garbage Space



Purge Garbage Space

정의 및 목적

- 레코드를 삭제할 경우, 바로 삭제되지 않고 해당 레코드 헤더에 위치한 'Info_flags' (4 bit)의 bit를 2로 설정
- Purge Thread가 COMMIT이 적용된 페이지 헤더에 위치한 **Garbage Offset**을 설정함으로써 해당 공간을 재사용
(* Garbage Offset = start of page free record list pointer)
- 2건 이상 DELETE, COMMIT될 경우, Garbage Offset은 **가장 마지막에 삭제된 레코드**의 포인터를 가지며, 가장 마지막에 삭제된 행은 그 이전에 삭제된 행의 포인터를 가짐으로써 삭제된 공간의 재사용은 마지막부터 시작





Purge Garbage Space

1. Insert into exem_i.t1 values ('Rec0000','A','1 ') ... ('Rec0040','A','1');

FIL Header

- Checksum : 0x9732acfa
- LSN : 0x0f7f2cbe

Page Header

- PAGE_N_RECS : 5
- Last Insert : Rec0040

Rec0000

- Next Record : Rec0010

Rec0040

- Next Record : supremum

000000	97	32	ac	fa	00	00	00	03	ff	ff	ff	ff	ff	ff	ff	>.2.....<
000010	00	00	00	00	0f	7f	2c	be	45	bf	00	00	00	00	00	>.....,E.....<
000020	00	00	00	00	00	e0	00	02	0a	78	80	07	00	00	00	>.....x.....<
000030	08	82	00	02	00	04	00	05	00	00	00	00	00	00	00	>.....<
000040	00	00	00	00	00	00	00	00	da	00	00	00	e0	00	00	>.....<
000050	00	02	00	f2	00	00	00	e0	00	00	00	02	00	32	01	>.....2..<
000060	02	00	1f	69	6e	66	69	6d	75	6d	00	06	00	0b	00	>...infimum.....<
000070	73	75	70	72	65	6d	75	6d	d0	81	0b	0e	00	00	00	>supremum.....<
000080	02	00	52	65	63	30	30	30	30	20	20	20	20	20	20	>..Rec0000 <
000090	00	00	00	01	86	18	a1	00	00	00	04	01	10	41	20	>.....A <
0000a0	20	20	20	20	20	20	20	20	31	20	20	20	20	20	20	> 1 <
0000b0	20	20	20	20	20	20	20	20	20	20	20	20	20	20	20	> <
*																
000270	20	20	20	20	20	20	20	20	d0	81	0b	0e	00	00	00	><
000280	02	00	52	65	63	30	30	31	30	20	20	20	20	20	20	>..Rec0010 <
000290	00	00	00	01	86	18	a1	00	00	00	04	01	27	41	20	>.....'A <
0002a0	20	20	20	20	20	20	20	20	31	20	20	20	20	20	20	> 1 <
0002b0	20	20	20	20	20	20	20	20	20	20	20	20	20	20	20	> <
*																
000470	20	20	20	20	20	20	20	20	d0	81	0b	0e	00	00	00	><
000480	02	00	52	65	63	30	30	32	30	20	20	20	20	20	20	>..Rec0020 <
000490	00	00	00	01	86	18	a1	00	00	00	04	01	3e	41	20	>.....>A <
0004a0	20	20	20	20	20	20	20	20	31	20	20	20	20	20	20	> 1 <
0004b0	20	20	20	20	20	20	20	20	20	20	20	20	20	20	20	> <
*																
000670	20	20	20	20	20	20	20	20	d0	81	0b	0e	00	00	00	>(<
000680	02	00	52	65	63	30	30	33	30	20	20	20	20	20	20	>..Rec0030 <
000690	00	00	00	01	86	18	a1	00	00	00	04	01	55	41	20	>.....UA <
0006a0	20	20	20	20	20	20	20	20	31	20	20	20	20	20	20	> 1 <
0006b0	20	20	20	20	20	20	20	20	20	20	20	20	20	20	20	> <
*																
000870	20	20	20	20	20	20	20	20	d0	81	0b	0e	00	00	00	>0<
000880	f7	ee	52	65	63	30	30	34	30	20	20	20	20	20	20	>..Rec0040 <
000890	00	00	00	01	86	18	a1	00	00	00	04	01	6c	41	20	>.....IA <



Purge Garbage Space

2. Begin;

Delete from exem_i.t1 where c1 = 'Rec0010' ;

Rec0010

- Info Flags 0x20UL (Deleted Bit)
- Transaction ID(6 Bytes)
- Roll Pointer(7 Bytes)

000000	97	32	ac	fa	00	00	00	03	ff	ff	ff	ff	ff	ff	ff	ff	>.2.....<
000010	00	00	00	00	0f	7f	2c	be	45	bf	00	00	00	00	00	00	>.....,E.....<
000020	00	00	00	00	00	e0	00	02	0a	78	80	07	00	00	00	00	>.....x.....<
000030	08	82	00	02	00	04	00	05	00	00	00	00	00	00	00	00	>.....<
000040	00	00	00	00	00	00	00	00	da	00	00	00	e0	00	00	00	>.....<
000050	00	02	00	f2	00	00	00	e0	00	00	00	02	00	32	01	00	>.....2..<
000060	02	00	1f	69	6e	66	69	6d	75	6d	00	06	00	0b	00	00	>...infimum.....<
000070	73	75	70	72	65	6d	75	6d	d0	81	0b	0e	00	00	00	10	>supremum.....<
000080	02	00	52	65	63	30	30	30	30	20	20	20	20	20	20	20	>..Rec0000 <
000090	00	00	00	01	86	18	a1	00	00	00	04	01	10	41	20	20	>.....A <
0000a0	20	20	20	20	20	20	20	20	31	20	20	20	20	20	20	20	> 1 <
0000b0	20	20	20	20	20	20	20	20	20	20	20	20	20	20	20	20	> <
*																	
000270	20	20	20	20	20	20	20	20	d0	81	0b	0e	00	20	00	18	><
000280	02	00	52	65	63	30	30	31	30	20	20	20	20	20	20	20	>..Rec0010 <
000290	00	00	00	01	86	21	22	00	00	00	08	1b	99	41	20	20	>.....!".....A <
0002a0	20	20	20	20	20	20	20	20	31	20	20	20	20	20	20	20	> 1 <
0002b0	20	20	20	20	20	20	20	20	20	20	20	20	20	20	20	20	> <



Purge Garbage Space

3. Commit ;

FIL Header

- Checksum : 0x9aa7a5de
- LSN : 0x7f3742

Page Header

- PAGE_N_RECS : 5

Rec0000

- Next Record : Rec0010

FIL Trailer

- Checksum : 0x9aa7a5de
- LSN : 0x7f3742

000000	9a	a7	a5	de	00	00	00	03	ff	ff	ff	ff	ff	ff	ff	>.....<
000010	00	00	00	00	0f	7f	37	42	45	bf	00	00	00	00	00	>.....7BE.....<
000020	00	00	00	00	e0	00	02	0a	78	80	07	00	00	00	00	>.....x.....<
000030	08	82	00	02	00	04	00	05	00	00	00	00	00	00	00	>.....<
000040	00	00	00	00	00	00	00	00	da	00	00	00	e0	00	00	>.....<
000050	00	02	00	f2	00	00	00	e0	00	00	00	02	00	32	01	>.....2..<
000060	02	00	1f	69	6e	66	69	6d	75	6d	00	06	00	0b	00	>...infimum.....<
000070	73	75	70	72	65	6d	75	6d	d0	81	0b	0e	00	00	10	>supremum.....<
000080	02	00	52	65	63	30	30	30	30	20	20	20	20	20	20	>..Rec0000 <
000090	00	00	00	01	86	18	a1	00	00	00	04	01	10	41	20	>.....A <
0000a0	20	20	20	20	20	20	20	20	31	20	20	20	20	20	20	> 1 <
0000b0	20	20	20	20	20	20	20	20	20	20	20	20	20	20	20	> <
*																
000270	20	20	20	20	20	20	20	20	d0	81	0b	0e	00	20	00	> <
000280	02	00	52	65	63	30	30	31	30	20	20	20	20	20	20	>..Rec0010 <
000290	00	00	00	01	86	21	22	00	00	00	08	1b	99	41	20	>.....!".....A <
0002a0	20	20	20	20	20	20	20	20	31	20	20	20	20	20	20	> 1 <
0002b0	20	20	20	20	20	20	20	20	20	20	20	20	20	20	20	> <
*																
000a70	20	20	20	20	20	20	20	20	00	00	00	00	00	00	00	><
000a80	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	>.....<
*																
003ff0	00	00	00	00	00	70	00	63	9a	a7	a5	de	0f	7f	37	>.....p.c.....7B<



Purge Garbage Space

4. Purge Garbage Space

Page Header

- Garbage Pointer => Rec0010
- Garbage Size : 0x200
- Last Insert : 0
- PAGE_N_RECS : 4

Rec0000

- Next Record : Rec0020

Rec0010

- Next Record : Null
- Info Flags 0x20UL (Deleted Bit)

000000	9a	a7	a5	de	00	00	00	03	ff	ff	ff	ff	ff	ff	ff	>.....<
000010	00	00	00	00	0f	7f	37	42	45	bf	00	00	00	00	00	>.....7BE.....<
000020	00	00	00	00	e0	00	02	0a	78	80	07	02	82	02	00	>.....x.....<
000030	00	00	00	02	00	04	00	04	00	00	00	00	00	00	00	>.....<
000040	00	00	00	00	00	00	00	00	00	da	00	00	00	e0	00	>.....<
000050	00	02	00	f2	00	00	00	e0	00	00	00	02	00	32	01	>.....2..<
000060	02	00	1f	69	6e	66	69	6d	75	6d	00	05	00	0b	00	>...infimum.....<
000070	73	75	70	72	65	6d	75	6d	d0	81	0b	0e	00	00	10	>supremum.....<
000080	04	00	52	65	63	30	30	30	30	20	20	20	20	20	20	>..Rec0000 <
000090	00	00	00	01	86	18	a1	00	00	00	04	01	10	41	20	>.....A <
0000a0	20	20	20	20	20	20	20	20	31	20	20	20	20	20	20	> 1 <
0000b0	20	20	20	20	20	20	20	20	20	20	20	20	20	20	20	> <
*																
000270	20	20	20	20	20	20	20	20	d0	81	0b	0e	00	20	00	> <
000280	00	00	52	65	63	30	30	31	30	20	20	20	20	20	20	>..Rec0010 <
000290	00	00	00	01	86	21	22	00	00	00	08	1b	99	41	20	>.....!".....A <
0002a0	20	20	20	20	20	20	20	20	31	20	20	20	20	20	20	> 1 <
0002b0	20	20	20	20	20	20	20	20	20	20	20	20	20	20	20	> <
*																
000470	20	20	20	20	20	20	20	20	d0	81	0b	0e	00	00	00	> <
000480	02	00	52	65	63	30	30	32	30	20	20	20	20	20	20	>..Rec0020 <
000490	00	00	00	01	86	18	a1	00	00	00	04	01	3e	41	20	>.....>A <
0004a0	20	20	20	20	20	20	20	20	31	20	20	20	20	20	20	> 1 <
0004b0	20	20	20	20	20	20	20	20	20	20	20	20	20	20	20	> <



Purge Garbage Space

5. Delete from exem_i.t1 where c1 = 'Rec0020' ;

FIL Header

- Checksum:0x89dd891a
- LSN : 0xf7f37ec

Rec0002

- Info Flags 0x20UL (Deleted Bit)

FIL Trailer

- Checksum : 0x89dd891a
- LSN : 0xf7f37ec

000000	89	dd	89	1a	00	00	00	03	ff	ff	ff	ff	ff	ff	ff	>.....<
000010	00	00	00	00	0f	7f	37	ec	45	bf	00	00	00	00	00	>.....7.E.....<
000020	00	00	00	00	e0	00	02	0a	78	80	07	02	82	02	00	>.....x.....<
000030	00	00	00	02	00	04	00	04	00	00	00	00	00	00	00	>.....<
000040	00	00	00	00	00	00	00	00	da	00	00	00	e0	00	00	>.....<
000050	00	02	00	f2	00	00	00	e0	00	00	00	02	00	32	01	>.....2..<
000060	02	00	1f	69	6e	66	69	6d	75	6d	00	05	00	0b	00	>...infimum.....<
000070	73	75	70	72	65	6d	75	6d	d0	81	0b	0e	00	00	00	>supremum.....<
000080	04	00	52	65	63	30	30	30	30	20	20	20	20	20	20	>..Rec0000 <
000090	00	00	00	01	86	18	a1	00	00	00	04	01	10	41	20	>.....A <
0000a0	20	20	20	20	20	20	20	20	31	20	20	20	20	20	20	> 1 <
0000b0	20	20	20	20	20	20	20	20	20	20	20	20	20	20	20	> <
*																
000270	20	20	20	20	20	20	20	20	d0	81	0b	0e	00	20	00	> <
000280	00	00	52	65	63	30	30	31	30	20	20	20	20	20	20	>..Rec0010 <
000290	00	00	00	01	86	21	22	00	00	00	08	1b	99	41	20	>.....!".....A <
0002a0	20	20	20	20	20	20	20	20	31	20	20	20	20	20	20	> 1 <
0002b0	20	20	20	20	20	20	20	20	20	20	20	20	20	20	20	> <
*																
000470	20	20	20	20	20	20	20	20	d0	81	0b	0e	00	20	00	> <
000480	02	00	52	65	63	30	30	32	30	20	20	20	20	20	20	>..Rec0020 <
000490	00	00	00	01	86	23	21	00	00	00	0a	29	92	41	20	>.....#!.....).A <
0004a0	20	20	20	20	20	20	20	20	31	20	20	20	20	20	20	> 1 <
003ff0	00	00	00	00	00	70	00	63	89	dd	89	1a	0f	7f	37	>.....p.c.....7.<



Purge Garbage Space

6. Commit ;

FIL Header

- Checksum :0xba9a2140
- LSN : 0xf7f38b4

Page Header

- PAGE_N_RECS : 4

Rec0000

- Next Record : Rec0020

FIL Trailer

- Checksum :0xba9a2140
- LSN : 0xf7f38b4

000000	ba	9a	21	40	00	00	00	03	ff	ff	ff	ff	ff	ff	ff	>..!@.....<
000010	00	00	00	00	0f	7f	38	b4	45	bf	00	00	00	00	00	>.....8.E.....<
000020	00	00	00	00	e0	00	02	0a	78	80	07	02	82	02	00	>.....x.....<
000030	00	00	00	02	00	04	00	04	00	00	00	00	00	00	00	>.....<
000040	00	00	00	00	00	00	00	00	da	00	00	00	e0	00	00	>.....<
000050	00	02	00	f2	00	00	00	e0	00	00	00	02	00	32	01	>.....2...<
000060	02	00	1f	69	6e	66	69	6d	75	6d	00	05	00	0b	00	>...infimum.....<
000070	73	75	70	72	65	6d	75	6d	d0	81	0b	0e	00	00	00	>supremum.....<
000080	04	00	52	65	63	30	30	30	30	20	20	20	20	20	20	>..Rec0000 <
000090	00	00	00	01	86	18	a1	00	00	00	04	01	10	41	20	>.....A <
0000a0	20	20	20	20	20	20	20	20	31	20	20	20	20	20	20	> 1 <
0000b0	20	20	20	20	20	20	20	20	20	20	20	20	20	20	20	> <
*																
000270	20	20	20	20	20	20	20	20	d0	81	0b	0e	00	20	00	><
000280	00	00	52	65	63	30	30	31	30	20	20	20	20	20	20	>..Rec0010 <
000290	00	00	00	01	86	21	22	00	00	00	08	1b	99	41	20	>.....!".....A <
0002a0	20	20	20	20	20	20	20	20	31	20	20	20	20	20	20	> 1 <
0002b0	20	20	20	20	20	20	20	20	20	20	20	20	20	20	20	> <
*																
000470	20	20	20	20	20	20	20	20	d0	81	0b	0e	00	20	00	><
000480	02	00	52	65	63	30	30	32	30	20	20	20	20	20	20	>..Rec0020 <
000490	00	00	00	01	86	23	21	00	00	00	0a	29	92	41	20	>....#!....).A <
0004a0	20	20	20	20	20	20	20	20	31	20	20	20	20	20	20	> 1 <
0004b0	20	20	20	20	20	20	20	20	20	20	20	20	20	20	20	> <
*																
003ff0	00	00	00	00	00	70	00	63	ba	9a	21	40	0f	7f	38	>....p.c..!@..8.<



Purge Garbage Space

7. Purge Garbage Space

Page Header

- PAGE_N_RECS : 3
- Garbage Offset : Rec0020
- Garbage Size : 0x400

Rec0000

- Next Record : Rec0030

Rec0020

- Next Record : Rec0010

000000	ba	9a	21	40	00	00	00	03	ff	ff	ff	ff	ff	ff	ff	>..!@.....<
000010	00	00	00	00	0f	7f	38	b4	45	bf	00	00	00	00	00	>.....8.E.....<
000020	00	00	00	00	e0	00	02	0a	78	80	07	04	82	04	00	>.....x.....<
000030	00	00	00	02	00	04	00	03	00	00	00	00	00	00	00	>.....<
000040	00	00	00	00	00	00	00	00	da	00	00	00	e0	00	00	>.....<
000050	00	02	00	f2	00	00	00	e0	00	00	00	02	00	32	01	>.....2..<
000060	02	00	1f	69	6e	66	69	6d	75	6d	00	04	00	0b	00	>...infimum.....<
000070	73	75	70	72	65	6d	75	6d	d0	81	0b	0e	00	00	00	>supremum.....<
000080	06	00	52	65	63	30	30	30	30	20	20	20	20	20	20	>..Rec0000 <
000090	00	00	00	01	86	18	a1	00	00	00	04	01	10	41	20	>.....A <
0000a0	20	20	20	20	20	20	20	20	31	20	20	20	20	20	20	> 1 <
0000b0	20	20	20	20	20	20	20	20	20	20	20	20	20	20	20	> <
*																
000270	20	20	20	20	20	20	20	20	d0	81	0b	0e	00	20	00	> <
000280	00	00	52	65	63	30	30	31	30	20	20	20	20	20	20	>..Rec0010 <
000290	00	00	00	01	86	21	22	00	00	08	1b	99	41	20	20	>.....!".....A <
0002a0	20	20	20	20	20	20	20	20	31	20	20	20	20	20	20	> 1 <
0002b0	20	20	20	20	20	20	20	20	20	20	20	20	20	20	20	> <
*																
000470	20	20	20	20	20	20	20	20	d0	81	0b	0e	00	20	00	> <
000480	fe	00	52	65	63	30	30	32	30	20	20	20	20	20	20	>..Rec0020 <
000490	00	00	00	01	86	23	21	00	00	00	0a	29	92	41	20	>.....#!.....).A <
0004a0	20	20	20	20	20	20	20	20	31	20	20	20	20	20	20	> 1 <
0004b0	20	20	20	20	20	20	20	20	20	20	20	20	20	20	20	> <
*																
000670	20	20	20	20	20	20	20	20	d0	81	0b	0e	00	00	00	>(<
000680	02	00	52	65	63	30	30	33	30	20	20	20	20	20	20	>..Rec0030 <
000690	00	00	00	01	86	18	a1	00	00	00	04	01	55	41	20	>.....UA <
0006a0	20	20	20	20	20	20	20	20	31	20	20	20	20	20	20	> 1 <
0006b0	20	20	20	20	20	20	20	20	20	20	20	20	20	20	20	> <



InnoDB 특징

- 1. 역사
- 2. 아키텍처
- 3. 메모리
- 4. 디스크
- 5. 성능

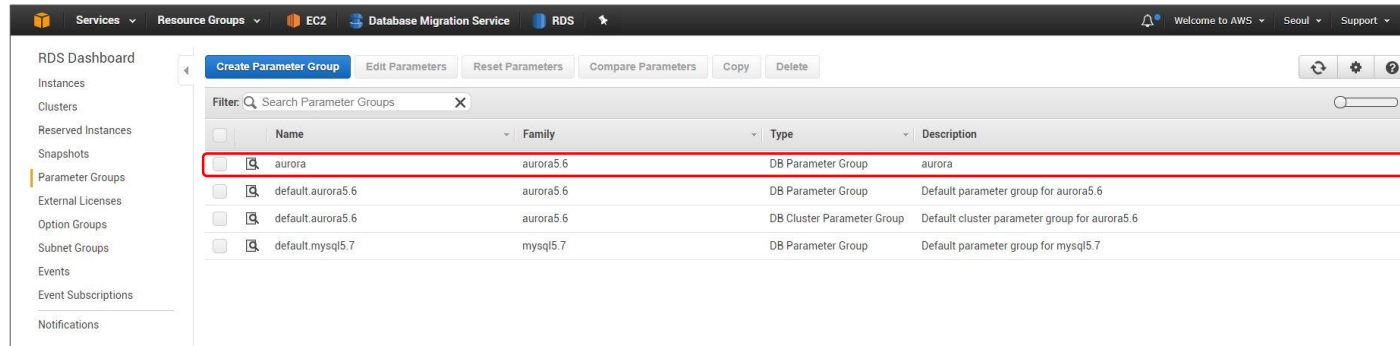
- 5.1 Performance_schema 설정
- 5.2 Performance_schema 설정에 따른 벤치마크



Performance_schema 활성화 (1/3)

- RDS Dashboard - Parameter Groups - Edit Parameters

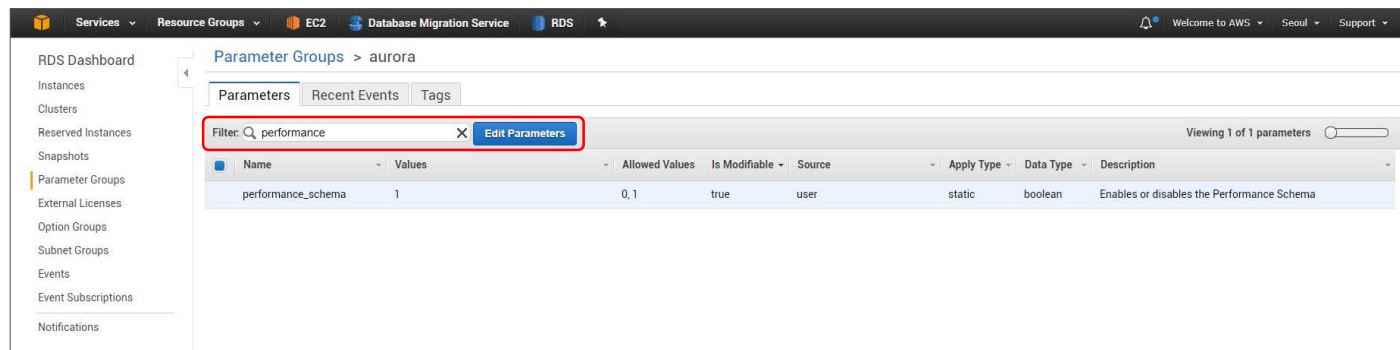
1) default 외에 새로 파라미터 그룹 생성 - Create Parameter Group



The screenshot shows the AWS RDS console's 'Parameter Groups' page. The 'aurora' parameter group is highlighted with a red box. The table below shows the list of parameter groups:

Name	Family	Type	Description
aurora	aurora5.6	DB Parameter Group	aurora
default.aurora5.6	aurora5.6	DB Parameter Group	Default parameter group for aurora5.6
default.aurora5.6	aurora5.6	DB Cluster Parameter Group	Default cluster parameter group for aurora5.6
default.mysql5.7	mysql5.7	DB Parameter Group	Default parameter group for mysql5.7

2) 새로 생성한 "aurora" 파라미터를 선택하고 Filter에 "performance_schema" 검색 후, Edit Parameters



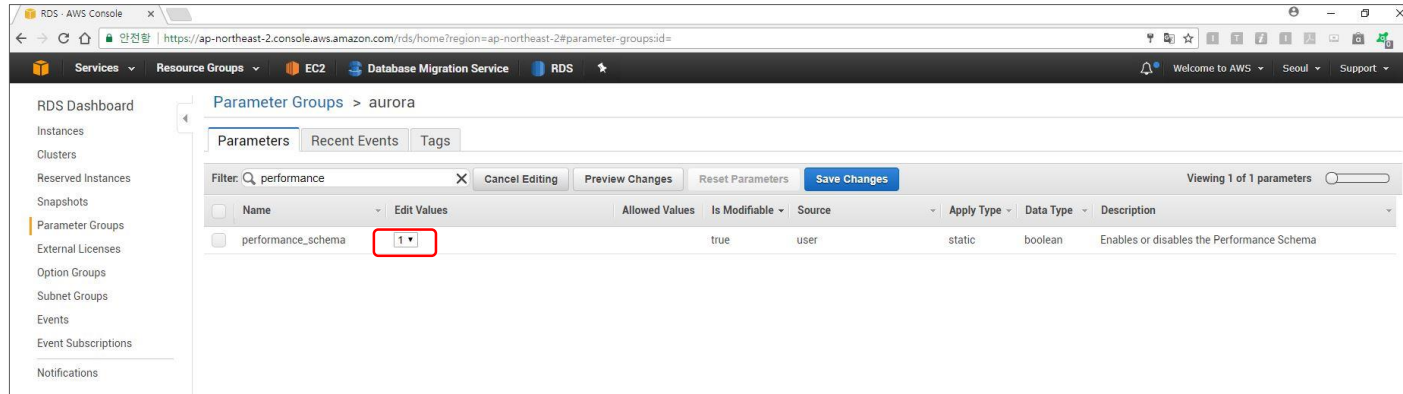
The screenshot shows the 'aurora' parameter group details page. The 'performance_schema' parameter is highlighted with a red box. The table below shows the details of the parameter:

Name	Values	Allowed Values	Is Modifiable	Source	Apply Type	Data Type	Description
performance_schema	1	0,1	true	user	static	boolean	Enables or disables the Performance Schema

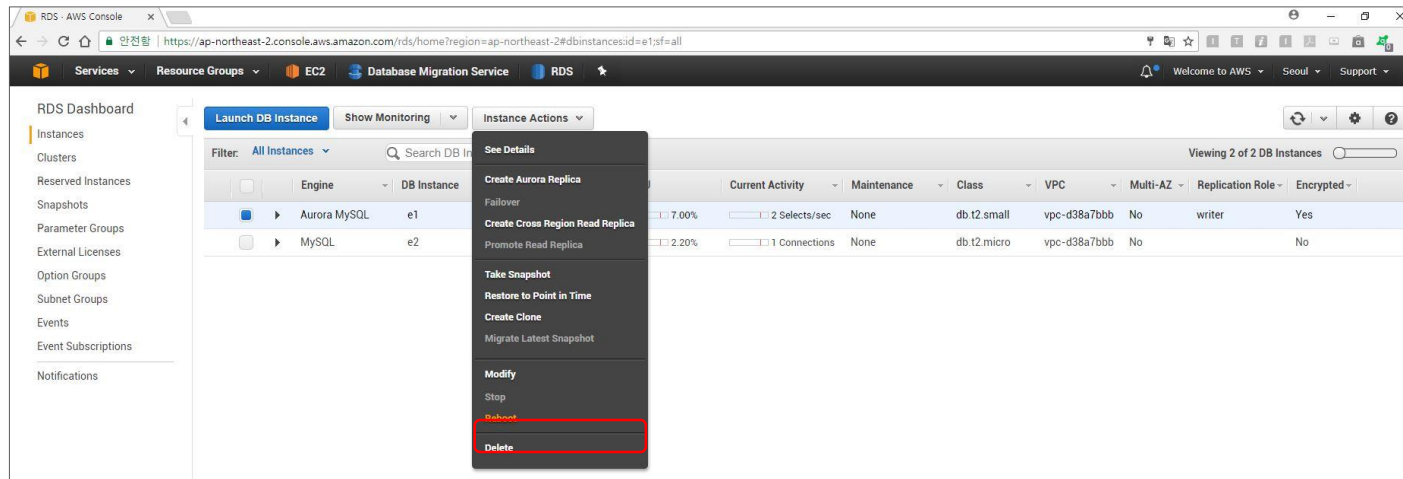


Performance_schema 활성화 (2/3)

- Save Changes - Reboot



3) 0에서 1로 변경 후, Save Changes



4) Reboot



Performance_schema 활성화 (3/3)

- performance_schema = ON

```
mysql> show variables like 'performance_schema';
+-----+-----+
| Variable_name | Value |
+-----+-----+
| performance_schema | ON    |
+-----+-----+
1 row in set (0.01 sec)
```




Performance_schema 설정

Performance_schema 설정 테이블 목록

필요한 부분만 설정하여 오버헤드를 최소화

```
mysql> show tables from performance_schema like 'setup%';
+-----+
| Tables_in_performance_schema (setup%) |
+-----+
| setup_actors                          |
| setup_consumers                       |
| setup_instruments                     |
| setup_objects                         |
| setup_timers                          |
+-----+
5 rows in set (0.01 sec)
```

setup_actors	모니터링 대상 유저 설정
setup_consumers	모니터링 데이터 테이블 설정
setup_instruments	모니터링 지표 설정
setup_objects	모니터링 대상 오브젝트 설정
setup_timers	모니터링 타이머 값 설정



Performance_schema 설정

setup_consumer 기본 설정 값 확인

```
mysql> select * from performance_schema.setup_consumers;
+-----+-----+
| NAME                                | ENABLED |
+-----+-----+
| events_stages_current               | NO      |
| events_stages_history               | NO      |
| events_stages_history_long          | NO      |
| events_statements_current           | YES     |
| events_statements_history           | NO      |
| events_statements_history_long      | NO      |
| events_waits_current                | NO      |
| events_waits_history                | NO      |
| events_waits_history_long           | NO      |
| global_instrumentation              | YES     |
| thread_instrumentation              | YES     |
| statements_digest                   | YES     |
+-----+-----+
12 rows in set (0.02 sec)
```



Performance_schema 설정

setup_consumer 설정

```
mysql> update performance_schema.setup_consumers set enabled='no' where name not in
('global_instrumentation','thread_instrumentation');
Query OK, 2 rows affected (0.00 sec)
Rows matched: 10  Changed: 2  Warnings: 0
```

```
mysql> select * from performance_schema.setup_consumers;
```

NAME	ENABLED
events_stages_current	NO
events_stages_history	NO
events_stages_history_long	NO
events_statements_current	NO
events_statements_history	NO
events_statements_history_long	NO
events_waits_current	NO
events_waits_history	NO
events_waits_history_long	NO
global_instrumentation	YES
thread_instrumentation	YES
statements_digest	NO

```
12 rows in set (0.00 sec)
```



Performance_schema 설정

setup_instruments 기본 설정 값 확인

```
mysql> select substr(name,1,instr(concat(name,'/'),'/')) instrument_type, enabled,  
timed, count(*)  
-> from performance_schema.setup_instruments  
-> group by 1,2,3;
```

instrument_type	enabled	timed	count(*)
idle	YES	YES	1
stage/	NO	NO	115
statement/	YES	YES	173
wait/	YES	YES	51
wait/	NO	NO	240

5 rows in set (0.02 sec)



Performance_schema 설정

setup_instruments 초기화 후, 필요한 지표만 설정

```
mysql> update performance_schema.setup_instruments set enabled='no', timed='no';  
Query OK, 225 rows affected (0.02 sec)  
Rows matched: 580  Changed: 225  Warnings: 0
```

```
mysql> update performance_schema.setup_instruments set enabled='yes', timed='yes'  
where name like 'wait%';  
Query OK, 291 rows affected (0.03 sec)  
Rows matched: 291  Changed: 291  Warnings: 0
```



Performance_schema 설정

PS 기본 설정 값 확인

```
mysql> select substr(name,1,instr(concat(name,'/'),'/')) instrument_type, enabled,
timed, count(*)
  -> from performance_schema.setup_instruments group by 1,2,3;
+-----+-----+-----+-----+
| instrument_type | enabled | timed | count(*) |
+-----+-----+-----+-----+
| idle           | NO     | NO    | 1        |
| stage/        | NO     | NO    | 115     |
| statement/    | NO     | NO    | 173     |
| wait/         | YES    | YES   | 291     |
+-----+-----+-----+-----+
4 rows in set (0.01 sec)
```



InnoDB 특징

- 1. 역사
- 2. 아키텍처
- 3. 메모리
- 4. 디스크
- 5. 성능

- 5.1 Performance_schema 설정
- 5.2 Performance_schema 설정에 따른 벤치마크





Performance_schema 벤치마크

[테스트 환경]

PERFORMANCE_SCHEMA = OFF

[테스트 결과]

OLTP test statistics:

queries performed	read	2520742		
	write	720061		
	other	360053		
	total	3600856		
	transactions	180000 (2555.84 per sec.)		
	read/write requests	3240803 (46016.47 per sec.)		
	other operations	360053 (5112.43 per sec.)		
	ignored errors	53 (0.75 per sec.)		
	reconnects	0 (0.00 per sec.)		
General statistics	total time	70.4270s		
	total number of events	180000		
	total time taken by event execution	7037.4301s		
	response time	min	3.85ms	
		avg	39.10ms	
		max	1018.18ms	
approx. 95 percentile		71.99ms		
Threads fairness	events (avg/stddev)	1800.0000/32.00		
	execution time (avg/stddev)	70.3743/0.01		



Performance_schema 벤치마크

[테스트 환경]

PERFORMANCE_SCHEMA = ON DEFAULT

[테스트 결과]

OLTP test statistics:

queries performed	read	2520812		
	write	720070		
	other	360058		
	total	3600940		
	transactions	180000 (2443.41 per sec.)		
	read/write requests	3240882 (43993.41 per sec.)		
	other operations	360058 (4887.61 per sec.)		
	ignored errors	58 (0.79 per sec.)		
	reconnects	0 (0.00 per sec.)		
General statistics	total time	73.6674s		
	total number of events	180000		
	total time taken by event execution	7363.8586s		
	response time	min	5.23ms	
		avg	40.91ms	
		max	708.95ms	
approx. 95 percentile		74.71ms		
Threads fairness	events (avg/stddev)	1800.0000/29.71		
	execution time (avg/stddev)	73.6386/0.01		



Performance_schema 벤치마크

[테스트 환경]

PERFORMANCE_SCHEMA = ON #필요한 부분만 설정
WAIT INSTRUMENTS

[테스트 결과]

OLTP test statistics:

queries performed	read	2246062		
	write	641634		
	other	320831		
	total	3208527		
	transactions	160398 (2673.05 per sec.)		
	read/write requests	2887696 (48123.69 per sec.)		
	other operations	320831 (5346.67 per sec.)		
	ignored errors	35 (0.58 per sec.)		
	reconnects	0 (0.00 per sec.)		
General statistics	total time	60.0057s		
	total number of events	160398		
	total time taken by event execution	5999.8410s		
	response time	min	4.55ms	
		avg	37.41ms	
		max	763.07ms	
approx. 95 percentile		65.22ms		
Threads fairness	events (avg/stddev)	1603.9800/25.70		
	execution time (avg/stddev)	59.9984/0.00		



Performance_schema 벤치마크

[테스트 환경]

PERFORMANCE_SCHEMA = ON #모든 지표 설정 ON
ALL INSTRUMENTS = 'YES'

[테스트 결과]

OLTP test statistics:

queries performed	read	2013396		
	write	575139		
	other	287585		
	total	2876120		
	transactions	143771 (2395.72 per sec.)		
	read/write requests	2588535 (43133.83 per sec.)		
	other operations	287585 (4792.15 per sec.)		
	ignored errors	43 (0.72 per sec.)		
	reconnects	0 (0.00 per sec.)		
General statistics	total time	60.0117s		
	total number of events	143771		
	total time taken by event execution	5999.3113s		
	response time	min	4.33ms	
		avg	41.73ms	
		max	1215.04ms	
approx. 95 percentile		73.32ms		
Threads fairness	events (avg/stddev)	1437.7100/26.33		
	execution time (avg/stddev)	59.9931/0.01		



Performance_schema 벤치마크

[테스트 환경]

PERFORMANCE_SCHEMA = ON #모든 지표 설정 OFF
ALL INSTRUMENTS = 'NO'

[테스트 결과]

OLTP test statistics:

queries performed	read	2181942		
	write	623274		
	other	311657		
	total	3116873		
	transactions	155804 (2594.42 per sec.)		
	read/write requests	2805216 (46712.02 per sec.)		
	other operations	311657 (5189.66 per sec.)		
	ignored errors	49 (0.82 per sec.)		
	reconnects	0 (0.00 per sec.)		
General statistics	total time	60.0534s		
	total number of events	155804		
	total time taken by event execution	6000.0118s		
	response time	min		3.38ms
		avg		38.51ms
		max		889.88ms
		approx. 95 percentile		70.67ms
Threads fairness	events (avg/stddev)	1558.0400/31.69		
	execution time (avg/stddev)	60.0001/0.02		



Performance_schema 벤치마크

Performance_schema	Setup_Instruments	TRANSACTIONS PER SECOND	RESPONSE TIME AVG (ms)	R/W REQUESTS PER SECOND
OFF	-	2711	36.88	48809
ON	Defaults	2544 (- 6%)	39.25 (+2.4)	45810
ON (MFM 환경)	Wait events	2673 (- 1.41%)	37.41 (+0.53)	48123
ON	ALL	2395 (- 11.65%)	41.73 (+4.85)	43133
ON	NO	2594 (- 4.31%)	36.65 (+0.23)	49123

- 테스트 환경 -

Benchmark Tool : sysbench (OLTP)

Total time : 60s

Threads : 100



Performance_schema 벤치마크

Performance_schema	Setup	TPM			
		1 차	2차	3차	평균
OFF	-	463,752	460,944	459,622	461,439(100%)
ON	Default	455,609	449,237	454,342	453,063(98.2%)
ON	ALL-enable	437,462	441,968	432,719	437,383(94.8%)
ON	All-disable	466,526	466,564	442,941	458,677(99.4%)

- 테스트 환경 -

Benchmark Tool : HammerDB(TPC-C for MySQL)

Total time : Rampup (2m), Timing test (5m)

Virtual User : 8

Aurora : **db.t2.small** class



Performance_schema 성능테스트



테스트결과

[테스트 환경]

PERFORMANCE_SCHEMA = OFF

[테스트 결과]

Log Output temp file .

1차 테스트	Start	Hammerdb Log @ Fri Nov 17 14:38:44 KST 2017
	Rampup	2 minutes
	Timing test	5 minutes
	TEST RESULT	463752 MySQL TPM at 7090 NOPM
2차 테스트	Start	Hammerdb Log @ Fri Nov 17 14:52:49 KST 2017
	Rampup	2 minutes
	Timing test	5 minutes
	TEST RESULT	460944 MySQL TPM at 7077 NOPM
3차 테스트	Start	Hammerdb Log @ Fri Nov 17 15:07:07 KST 2017
	Rampup	2 minutes
	Timing test	5 minutes
	TEST RESULT	459622 MySQL TPM at 7021 NOPM



Performance_schema 성능테스트



테스트결과

[테스트 환경]

PERFORMANCE_SCHEMA = ON

setup_consumers : default

setup_instruments : default

[테스트 결과]

Log Output temp file .

1차 테스트	Start	Hammerdb Log @ Fri Nov 17 13:53:02 KST 2017
	Rampup	2 minutes
	Timing test	5 minutes
	TEST RESULT	455609 MySQL TPM at 6947 NOPM
2차 테스트	Start	Hammerdb Log @ Fri Nov 17 14:02:00 KST 2017
	Rampup	2 minutes
	Timing test	5 minutes
	TEST RESULT	449237 MySQL TPM at 6888 NOPM
3차 테스트	Start	Hammerdb Log @ Fri Nov 17 16:16:09 KST 2017
	Rampup	2 minutes
	Timing test	5 minutes
	TEST RESULT	454342 MySQL TPM at 6983 NOPM



Performance_schema 성능테스트

[테스트 환경]

PERFORMANCE_SCHEMA = ON

setup_consumers : all enable

→ update performance_schema.setup_consumers
set enabled='yes' ;

setup_instruments : all enable

→ update performance_schema.setup_instruments
set enabled='yes' ;

[테스트 결과]

Log Output temp file .

1차 테스트	Start	Hammerdb Log @ Fri Nov 17 17:22:27 KST 2017
	Rampup	2 minutes
	Timing test	5 minutes
	TEST RESULT	437462 MySQL TPM at 6685 NOPM
2차 테스트	Start	Hammerdb Log @ Fri Nov 17 17:34:46 KST 2017
	Rampup	2 minutes
	Timing test	5 minutes
	TEST RESULT	441968 MySQL TPM at 6754 NOPM
3차 테스트	Start	Hammerdb Log @ Fri Nov 17 17:42:59 KST 2017
	Rampup	2 minutes
	Timing test	5 minutes
	TEST RESULT	432719 MySQL TPM at 6588 NOPM



Performance_schema 성능테스트

[테스트 환경]

PERFORMANCE_SCHEMA = ON

setup_consumers : all disable

→ update performance_schema.setup_consumers
set enabled='no' ;

setup_instruments : all disable

→ update performance_schema.setup_instruments
set enabled='no' ;

[테스트 결과]

Log Output temp file .

1차 테스트	Start	Hammerdb Log @ Fri Nov 17 16:50:35 KST 2017
	Rampup	2 minutes
	Timing test	5 minutes
	TEST RESULT	466526 MySQL TPM at 7135 NOPM
2차 테스트	Start	Hammerdb Log @ Fri Nov 17 17:04:47 KST 2017
	Rampup	2 minutes
	Timing test	5 minutes
	TEST RESULT	466564 MySQL TPM at 7137 NOPM
3차 테스트	Start	Hammerdb Log @ Fri Nov 17 17:13:05 KST 2017
	Rampup	2 minutes
	Timing test	5 minutes
	TEST RESULT	442941 MySQL TPM at 6762 NOPM



Monitoring





Amazon CloudWatch



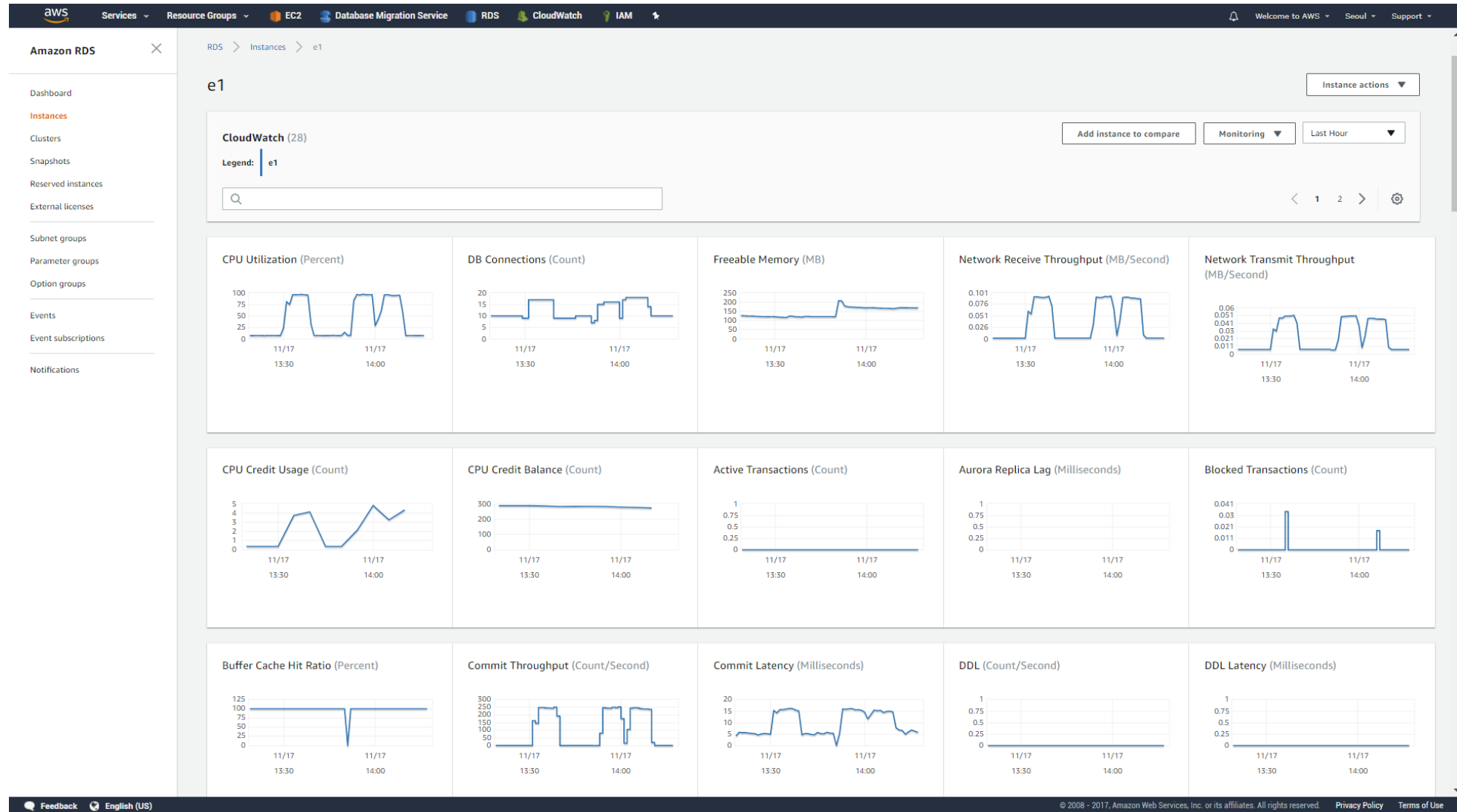
AWS 클라우드 리소스와 AWS에서 실행되는 애플리케이션을 위한 모니터링 서비스

- 지표를 수집 및 추적
- 로그 파일을 수집 및 모니터링
- 경고 설정
- AWS 리소스 변경에 자동 대응
- 시스템 전반의 리소스 사용률, 애플리케이션 성능, 운영 상태를 파악

- Amazon EC2 인스턴스 기본 모니터링: 사전 선택한 지표 7개를 5분 간격으로, 상태 확인 지표 3개를 1분 간격으로 모니터링 (추가 비용 X)
- Amazon EC2 인스턴스 세부 모니터링: 기본 모니터링에서 사용 가능한 모든 지표를 1분 간격으로 확인 (추가 비용 0)
- Amazon RDS DB 인스턴스: 사전 선택한 14개를 1분 간격으로 모니터링 (추가 비용 X)



Amazon CloudWatch (Amazon RDS 콘솔)





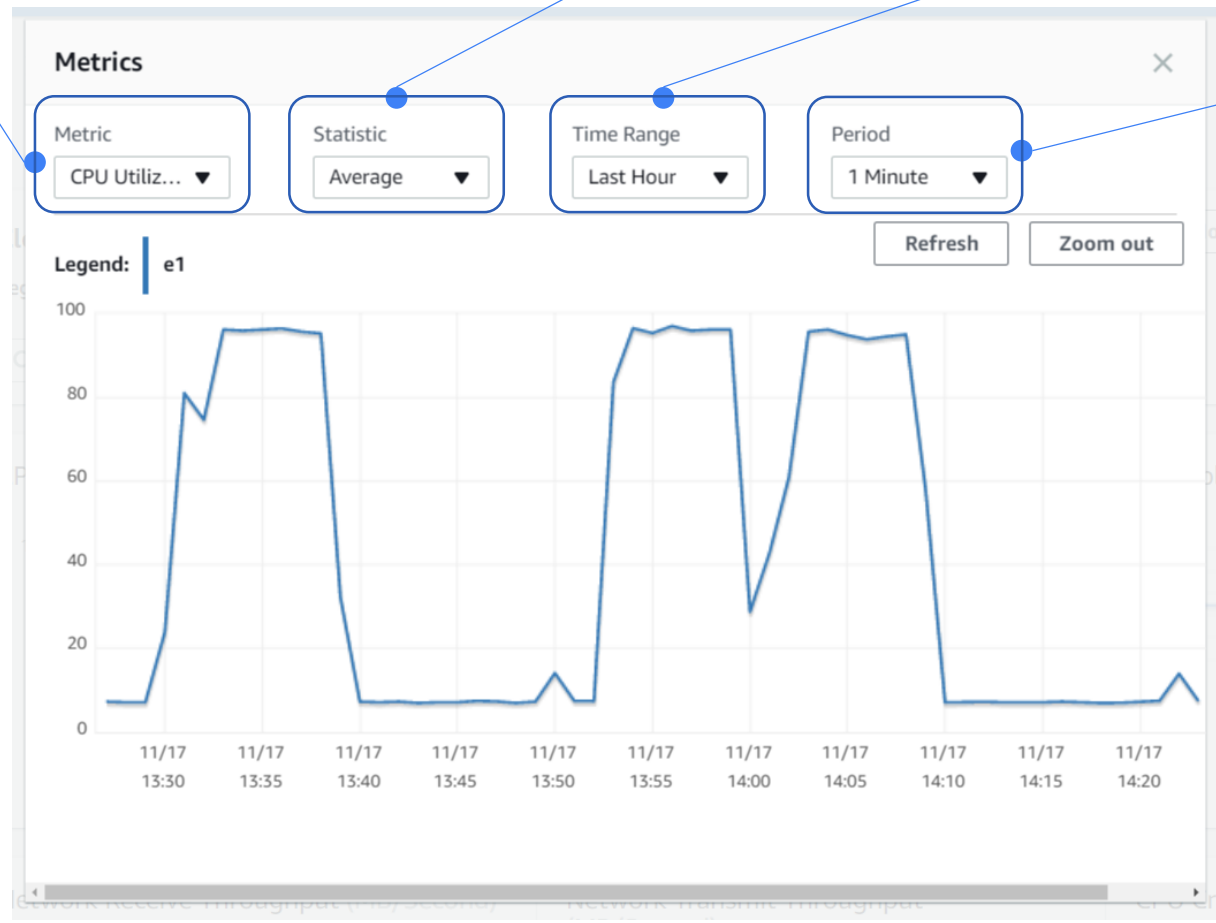
Amazon CloudWatch (Amazon RDS 콘솔)

Average
Minimum
Maximum
Sum
Sample Count

1 / 3 / 6 / 12 / 24 시간
3일 / 1주 / 2주 전

Amazon RDS 지표 및 차원
(총 28 개)

1 / 5 / 15 분
1 / 6 시간
1일





	설명	예시
네임스페이스	<ul style="list-style-type: none">CloudWatch 지표용 컨테이너다른 네임 스페이스의 지표는 격리되어 있어서 다른 애플리케이션 지표가 집계 되는 일은 없음	<code>--namespace AWS/service</code> Amazon EC2: AWS/EC2 Amazon RDS: AWS/RDS
지표	<ul style="list-style-type: none">CloudWatch에 게시된 시간 순서별 데이터 요소 세트이름, 네임스페이스 및 하나 이상의 차원으로 고유하게 정의됨	<code>--metric-name CPUUtilization</code>
통계	<ul style="list-style-type: none">지정한 기간에 걸친 지표 데이터 집계Minimum, Maximum, Sum, Average, SampleCount, pNN.NN	<code>--statistics "Maximum" "Average"</code> <code>"SampleCount"</code>
기간	<ul style="list-style-type: none">지정한 기간에 대해 수집된 지표 데이터의 집계초 단위로 정의되며 유효한 기간 값은 1, 5, 10, 30, 60의 배수	<code>--period 60</code>
타임스탬프	<ul style="list-style-type: none">시간, 분, 초와 더불어 완전한 날짜가 있는 dateTime 개체Cloudwatch에서 통계를 검색할 때 모든 시간은 UTC를 따름 (필수요건은 아님)cf) 서울 타임존 TZ = ASIA/Seoul	<code>--start-time 2017-09-29T00:30:00Z</code> <code>--end-time 2017-09-29T23:00:00Z</code>



Amazon CloudWatch (CLI)

IAM(Identity and Access Management) – Users – Add user

The screenshot shows the AWS IAM console interface. At the top, there is a navigation bar with the AWS logo and various service icons (EC2, Database Migration Service, RDS). Below this is a search bar labeled 'Search IAM' and two buttons: 'Add user' (highlighted with a red box) and 'Delete user'. The main content area features a search input field with the placeholder text 'Find users by username or access key'. Below the search field is a table with the following columns: 'User name', 'Groups', 'Access key age', 'Password age', 'Last activity', 'MFA', and 'Access key'. The table is currently empty, and a message below it states 'There are no IAM users. [Learn more](#)'. The left sidebar contains a navigation menu with options: Dashboard, Groups, Users (highlighted), Roles, Policies, Identity providers, Account settings, Credential report, and Encryption keys. At the bottom of the console, there is a footer with 'Feedback', 'English (US)', and copyright information: '© 2008 - 2017, Amazon Web Services, Inc. or its affiliates. All rights reserved. Privacy Policy Terms of Use'.



Amazon CloudWatch (CLI)

User name, Access type 선택

Set user details

You can add multiple users at once with the same access type and permissions. [Learn more](#)

User name*

[+ Add another user](#)

Select AWS access type

Select how these users will access AWS. Access keys and autogenerated passwords are provided in the last step. [Learn more](#)

Access type*

- Programmatic access**
Enables an **access key ID** and **secret access key** for the AWS API, CLI, SDK, and other development tools.
- AWS Management Console access**
Enables a **password** that allows users to sign-in to the AWS Management Console.

* Required

[Cancel](#) [Next: Permissions](#)

[Feedback](#) [English \(US\)](#) © 2008 - 2017, Amazon Web Services, Inc. or its affiliates. All rights reserved. [Privacy Policy](#) [Terms of Use](#)



Amazon CloudWatch (CLI)

Add user to group – Create group

aws Services Resource Groups EC2 Database Migration Service RDS Welcome to AWS Global Support

Set permissions for iam001

- Add user to group
- Copy permissions from existing user
- Attach existing policies directly

Add user to an existing group or create a new one. Using groups is a best-practice way to manage user's permissions by job functions. [Learn more](#)

Create group Refresh

Search Showing 1 result

Group	Attached policies
-------	-------------------

Feedback English (US) © 2008 - 2017, Amazon Web Services, Inc. or its affiliates. All rights reserved. Privacy Policy Terms of Use



Amazon CloudWatch (CLI)

Group name, Filter에 "rds", "cloudwatch" 검색, 선택 후 Create group

Create group

Create a group and select the policies to be attached to the group. Using groups is a best-practice way to manage users' permissions by job functions, AWS service access, or your custom permissions. [Learn more](#)

Group name

Filter: Policy type Showing 6 results

	Policy name	Type	Attachments	Description
<input checked="" type="checkbox"/>	AmazonRDSDirectoryServic...	AWS managed	1	Allow RDS to access Directory Service Managed AD on behalf of t...
<input type="checkbox"/>	AmazonRDSEnhancedMoni...	AWS managed	1	Provides access to Cloudwatch for RDS Enhanced Monitoring
<input type="checkbox"/>	AmazonRDSFullAccess	AWS managed	1	Provides full access to Amazon RDS via the AWS Management C...



Amazon CloudWatch (CLI)

Review 확인 후 , Create user

The screenshot shows the AWS IAM console interface for adding a user. The top navigation bar includes the AWS logo and various service icons. The main heading is 'Add user'. A progress indicator at the top right shows four steps: 1. Details, 2. Permissions, 3. Review (current step), and 4. Complete. Below the heading, there is a 'Review' section with the instruction: 'Review your choices. After you create the user, you can view and download the autogenerated password and access key.' The 'User details' section shows: 'User name: iam001' and 'AWS access type: Programmatic access - with an access key'. The 'Permissions summary' section states: 'The user shown above will be added to the following groups.' Below this is a table with one entry: 'Group: mygroup1'. At the bottom right, there are three buttons: 'Cancel', 'Previous', and 'Create user' (which is highlighted with a red border).

Type	Name
Group	mygroup1



Amazon CloudWatch (CLI)

Access key ID, Secret access key 확인

The screenshot shows the AWS IAM console 'Add user' page. At the top, there is a navigation bar with the AWS logo and various service links. Below the navigation bar, the page title 'Add user' is displayed. A progress indicator shows four steps: 1. Details, 2. Permissions, 3. Review, and 4. Complete (highlighted in blue). A green success message states: 'Success. You successfully created the users shown below. You can view and download user security credentials. You can also email users instructions for signing in to the AWS Management Console. This is the last time these credentials will be available to download. However, you can create new credentials at any time. Users with AWS Management Console access can sign-in at: <https://638505169563.signin.aws.amazon.com/console>'. Below the message is a 'Download .csv' button. A table lists the user 'iam001' with a checkmark in the status column. The table has two columns: 'Access key ID' and 'Secret access key'. The 'Access key ID' column shows 'A...' and the 'Secret access key' column shows '***** Show'. A red box highlights these two columns, and a red line points from a larger red box below to the 'Secret access key' column. The larger red box contains the text 'Access key ID' and 'Secret access key'. A 'Close' button is located at the bottom right of the table area.

User	Access key ID	Secret access key
iam001	A.....	***** Show



Amazon CloudWatch (CLI)

```
root@opml:~# cd .aws
root@opml:~/aws# aws configure
AWS Access Key ID [*****HJ2Q]:
AWS Secret Access Key [*****1hbG]:
Default region name [ap-northeast-2]:
Default output format [table]:
```

Access Key: 앞 슬라이드 참고

Secret Access Key: 앞 슬라이드 참고

region name: ap-northeast-2 형태로 작성

output format: [JSON | TEXT | TABLE]



Amazon CloudWatch (CLI)

```
root@opml:~# cat cwatch.sh
FR=$(TZ=UTC+1 date +%Y-%m-%dT%H:00:00)
TO=$(TZ=UTC-1 date +%Y-%m-%dT%H:00:00)

MET=DatabaseConnections
MET=CPUUtilization
MET=FreeableMemory
DBID=e1

banner $DBID

for i in CPUUtilization
do
aws cloudwatch get-metric-statistics --namespace AWS/RDS --metric-name $i \
  --statistics "Maximum" "Average" "SampleCount" \
  --dimensions Name=DBInstanceIdentifier,Value=$DBID \
  --period 600 \
  --start-time $FR \
  --end-time $TO
done
```

Amazon RDS 지표 및 차원



Amazon CloudWatch (CLI)

테이블 형태로 출력된 CPUUtilization 지표

```
root@opml:~# cwatch.sh
```

```
-----
```

GetMetricStatistics					
Label	CPUUtilization				

Datapoints					
Average	Maximum	SampleCount	Timestamp	Unit	
6.8990000000000001	7.87	10.0	2017-11-16T03:00:00Z	Percent	
6.8340000000000005	7.0	10.0	2017-11-16T03:40:00Z	Percent	
6.8510000000000001	7.17	10.0	2017-11-16T04:20:00Z	Percent	
6.839	6.95	10.0	2017-11-16T03:10:00Z	Percent	
6.886	7.63	10.0	2017-11-16T03:50:00Z	Percent	
7.433	8.67	10.0	2017-11-16T04:30:00Z	Percent	
6.833	7.05	10.0	2017-11-16T03:20:00Z	Percent	
6.806	7.33	10.0	2017-11-16T04:00:00Z	Percent	
6.7690000000000001	7.67	10.0	2017-11-16T04:40:00Z	Percent	
6.981	7.97	10.0	2017-11-16T03:30:00Z	Percent	
6.819	7.0	10.0	2017-11-16T04:10:00Z	Percent	

```
-----
```




Enhanced Monitoring

- DB 인스턴스가 실행되는 운영체제에 대한 측정치를 실시간으로 제공
- Amazon CloudWatch Logs이 제공하는 프리 티어를 초과하는 것에 대해서만 비용 청구
- 모니터링 간격이 작을수록 OS 측정치가 더 자주 보고되고 모니터링 비용이 증가
- 모니터링하는 DB 인스턴스의 수가 많을수록 더 많은 비용이 청구

CloudWatch	Enhanced Monitoring
DB 인스턴스의 하이퍼바이저에서 CPU 사용률에 대한 측정치 수집 (하이퍼바이저 계층에서는 소량의 작업만 수행)	인스턴스의 에이전트에서 측정치 수집
클래스가 적을수록 단일 물리적 인스턴스에서 하이퍼바이저 계층에 의해 관리되는 가상 머신의 수가 더 많아지면 차이가 커짐	DB 인스턴스의 여러 프로세스 또는 스레드에서 CPU를 사용하는 방법을 확인하려는 경우에 유용



Enhanced Monitoring

The screenshot shows the AWS Management Console for Amazon RDS. The main content area displays the 'Enhanced monitoring (50)' page for instance 'e1'. At the top, there is a search bar labeled '키워드 검색' (Keyword search). To the right, there are controls for time selection: '11/17/2017', '04', ': 50', and a 'Go' button. Further right is a 'Monitoring' dropdown menu and a time interval dropdown set to '5 minutes'. A settings gear icon is also visible. Below these controls is a grid of charts for various memory metrics: Free Memory, Total Memory, Writeback Memory, Used Memory, Active Memory, Slab Memory, Huge Pages Surp, Huge Pages Size, and Huge Pages Total. A callout box points to a data point on the 'Free Memory' chart, showing a value of 132.38 MB at 2:04:27 PM UTC+9 on November 17, 2017.

지정한 시간으로 이동

CloudWatch
Enhanced Monitoring
OS process list

키워드 검색

5 minutes
15 minutes
30 minutes
1 hour

page size
6 / 12 / 24
한 화면에 보이는
개수 지정

Free Memory

141.6 MB

Value: 132.38 MB
Time: November 17, 2017 at 2:04:27 PM UTC+9

136.0 MB

131.84 MB

126.95 MB

122.07 MB

11/17 05:04 11/17 05:06

5분으로 설정시 5분간의 지표가 보이며,
포인팅을 하면 각 시점의 값 확인 가능



Amazon RDS 지표 및 차원

Amazon Relational Database Service는 각 활성 데이터베이스 인스턴스에서 1분마다 CloudWatch로 지표를 전송합니다. 세부 모니터링 기능이 기본적으로 활성화되어 있습니다. 자세한 내용은 Amazon Relational Database Service 사용 설명서의 DB 인스턴스 모니터링을 참조하십시오.

Amazon RDS 지표

AWS/RDS 네임스페이스에는 다음 지표가 포함되어 있습니다.

표	설명
BinLogDiskUsage	마스터에서 이진 로그가 차지하는 디스크 공간 크기. MySQL 읽기 전용 복제본에 적용. 단위: 바이트
BurstBalance	사용할 수 있는 범용 SSD(gp2) 버스트-버킷 I/O 크레딧 비율 단위: 백분율
CPUUtilization	CPU 사용 백분율. 단위: 백분율
CPUCreditUsage	[T2 인스턴스] 인스턴스가 소비한 CPU 크레딧 수입입니다. CPU 크레딧 하나는 1분 동안 100%의 사용률로 실행되는 vCPU 1개 또는 이와 동등한 vCPU, 사용률 및 시간의 조합과 동일합니다(예를 들어 2분 동안 50%의 사용률로 실행되는 vCPU 1개 또는 2분 동안 25%의 사용률로 실행되는 vCPU 2개). CPU 크레딧 지표는 5분 간격으로만 제공됩니다. 5분 이상의 시간을 지정할 경우 Average 통계 대신 Sum 통계를 사용하십시오. 단위: 수
CPUCreditBalance	[T2 인스턴스] 인스턴스에 대해 기본 CPU 사용률 이상으로 버스트가 가능한 CPU 크레딧 수입입니다. 크레딧은 축적 이후에는 크레딧 잔고에 보관되고, 만료 이후에는 크레딧 잔고에서 소멸됩니다. 크레딧은 축적 이후 24시간이 지나면 만료됩니다. CPU 크레딧 지표는 5분 간격으로만 제공됩니다. 단위: 수
DatabaseConnections	사용 중인 데이터베이스 연결 수. 단위: 수



Amazon RDS 지표 및 차원

표	설명
DiskQueueDepth	디스크 액세스를 대기 중인 I/O(읽기/쓰기 요청) 수. 단위: 수
FreeableMemory	사용 가능한 RAM 크기. 단위: 바이트
FreeStorageSpace	사용 가능한 스토리지 공간 크기. 단위: 바이트
ReplicaLag	소스 DB 인스턴스를 기준으로 읽기 전용 복제본 DB 인스턴스의 지연 시간. MySQL, MariaDB 및 PostgreSQL 읽기 전용 복제본에 적용. 단위: 초
SwapUsage	DB 인스턴스에서 사용된 스왑 공간 크기. 단위: 바이트
ReadIOPS	초당 평균 디스크 I/O 연산 수. 단위: 수/초
WriteIOPS	초당 평균 디스크 I/O 연산 수. 단위: 수/초
ReadLatency	디스크 I/O 연산당 평균 처리 시간. 단위: 초
WriteLatency	디스크 I/O 연산당 평균 처리 시간. 단위: 초
ReadThroughput	초당 디스크에서 읽은 평균 바이트 수. 단위: 바이트/초
WriteThroughput	초당 디스크에 쓴 평균 바이트 수. 단위: 바이트/초
NetworkReceiveThroughput	DB 인스턴스 수신 네트워크 트래픽(고객 데이터베이스 트래픽과 모니터링 및 복제에 사용된 Amazon RDS 트래픽을 모두 포함). 단위: 바이트/초
NetworkTransmitThroughput	DB 인스턴스 송신 네트워크 트래픽(고객 데이터베이스 트래픽과 모니터링 및 복제에 사용된 Amazon RDS 트래픽을 모두 포함). 단위: 바이트/초



Amazon RDS 지표 및 차원

Amazon RDS 차원

다음 표의 차원을 사용하여 Amazon RDS 지표 데이터를 필터링할 수 있습니다.

차원	설명
DBInstanceIdentifier	이 차원은 특정 DB 인스턴스에 대해 요청하는 데이터를 필터링합니다.
DBClusterIdentifier	이 차원은 특정 Amazon Aurora DB 클러스터에 대해 요청하는 데이터를 필터링합니다.
DBClusterIdentifier, Role	이 차원은 인스턴스 역할(WRITER/READER)별로 지표를 집계하여 특정 Amazon Aurora DB 클러스터에 대해 요청하는 데이터를 필터링합니다. 예를 들어 클러스터에 속하는 모든 READER 인스턴스에 대한 지표를 집계할 수 있습니다.
DatabaseClass	이 차원은 특정 데이터베이스 클래스의 모든 인스턴스에 대해 요청하는 데이터를 필터링합니다. 예를 들어 데이터베이스 클래스 db.m1.small에 속하는 모든 인스턴스에 대한 지표를 집계할 수 있습니다.
EngineName	이 차원은 식별된 엔진 이름에 대해 요청하는 데이터만 필터링합니다. 예를 들어 엔진 이름이 mysql인 모든 인스턴스에 대한 지표를 집계할 수 있습니다.



MAXGAUGE REALTIME MONITOR

Instance Name Business Name

OVERALL	100%	100%
ORA102	100%	100%
JAPAN	100%	100%
CHINA	100%	100%
ORACLE	100%	100%

MAXGAUGE for Amazon Aurora **On AWS**

PRODUCT DOCUMENTATION





- **MAXGAUGE**

OVERVIEW

ARCHITECTURE

FEATURE

- **FEATURE**

REAL-TIME MONITOR

ADMIN

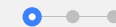
PERFORMANCE ANALYZER

- **PERFORMANCE ANALYZER**

ALERT & MONITORING

EXEM DASHBOARD





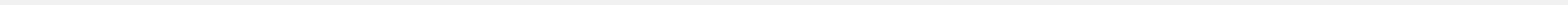
MAXGAUGE

for Amazon Aurora **on AWS**

OVERVIEW

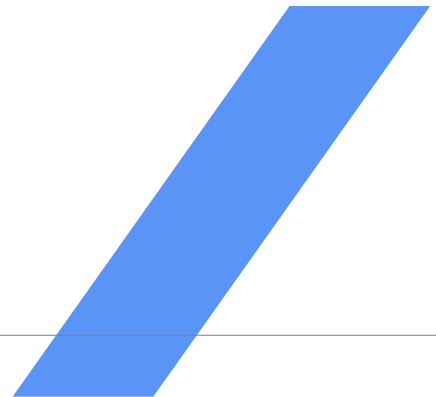
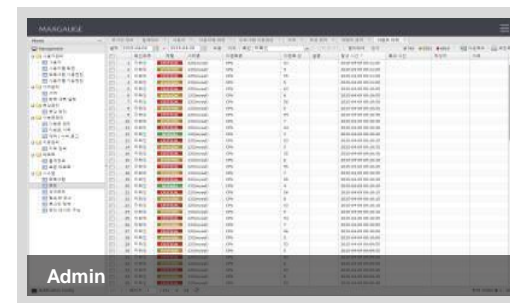
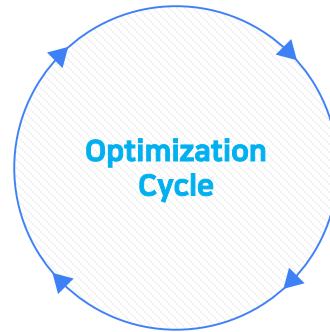
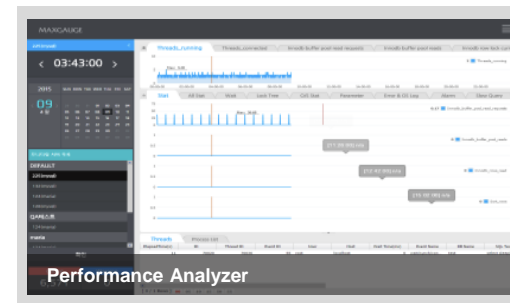
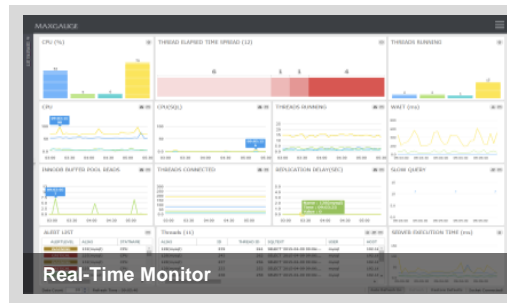
ARCHITECTURE

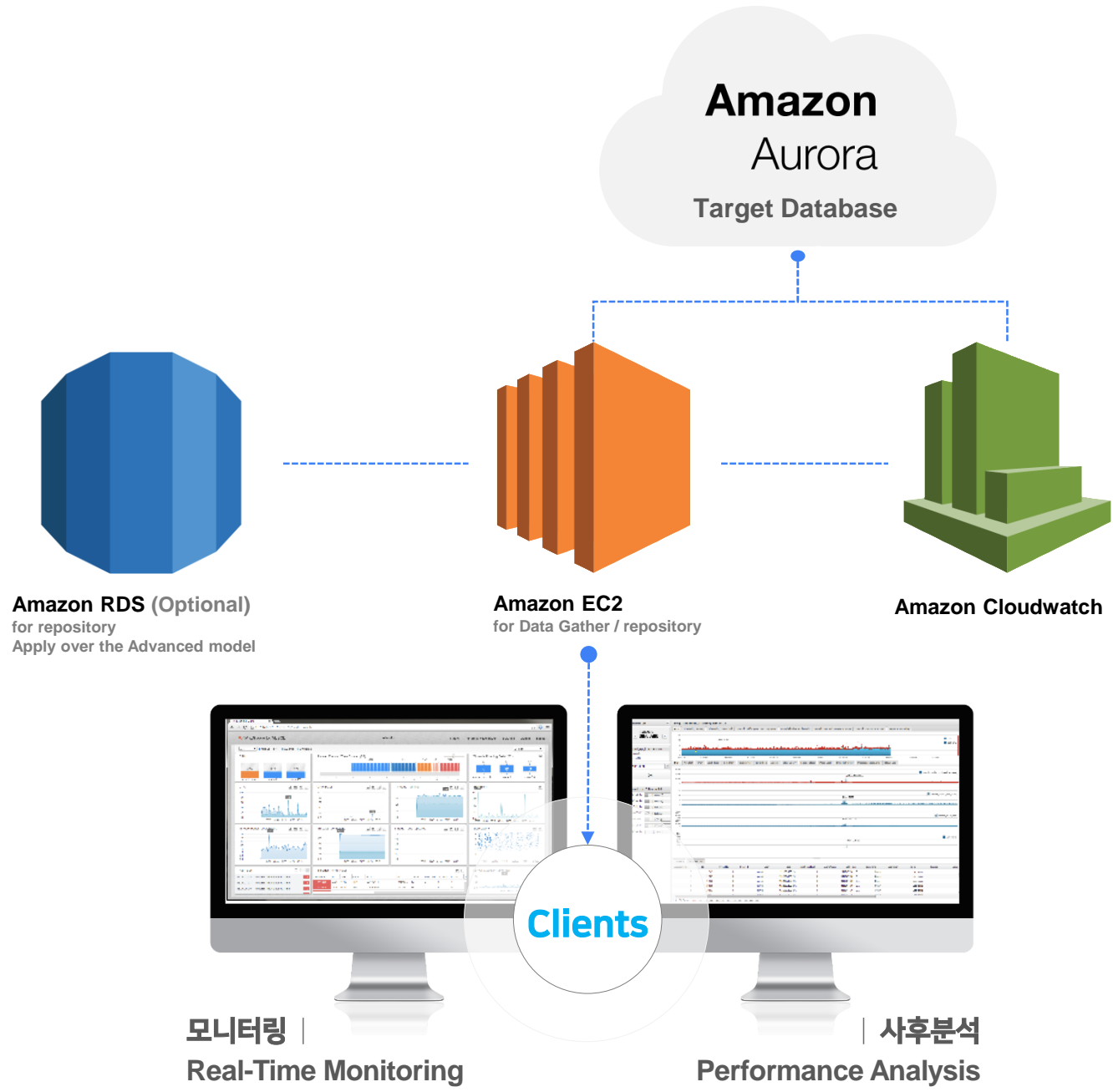
FEATURE



데이터베이스 성능 최적화를 위한 전문 소프트웨어 솔루션

맥스게이지는 데이터베이스 성능 관리를 위한 전문 소프트웨어 솔루션으로써 실시간 감시 및 가동정보의 수집, 정밀 진단 및 분석, SQL튜닝 등 다양한 기능을 제공하여 데이터베이스 시스템의 가용성 및 성능을 효율적으로 관리합니다. Pro-Active한 모니터링 기법을 제공함으로써 문제 상황을 빠르게 인지하고, 애플리케이션의 병목 현상을 효과적으로 분석하며, 다양한 성능데이터를 저장하여 투자 대비 효과에 대한 명확한 분석 자료를 제공합니다.

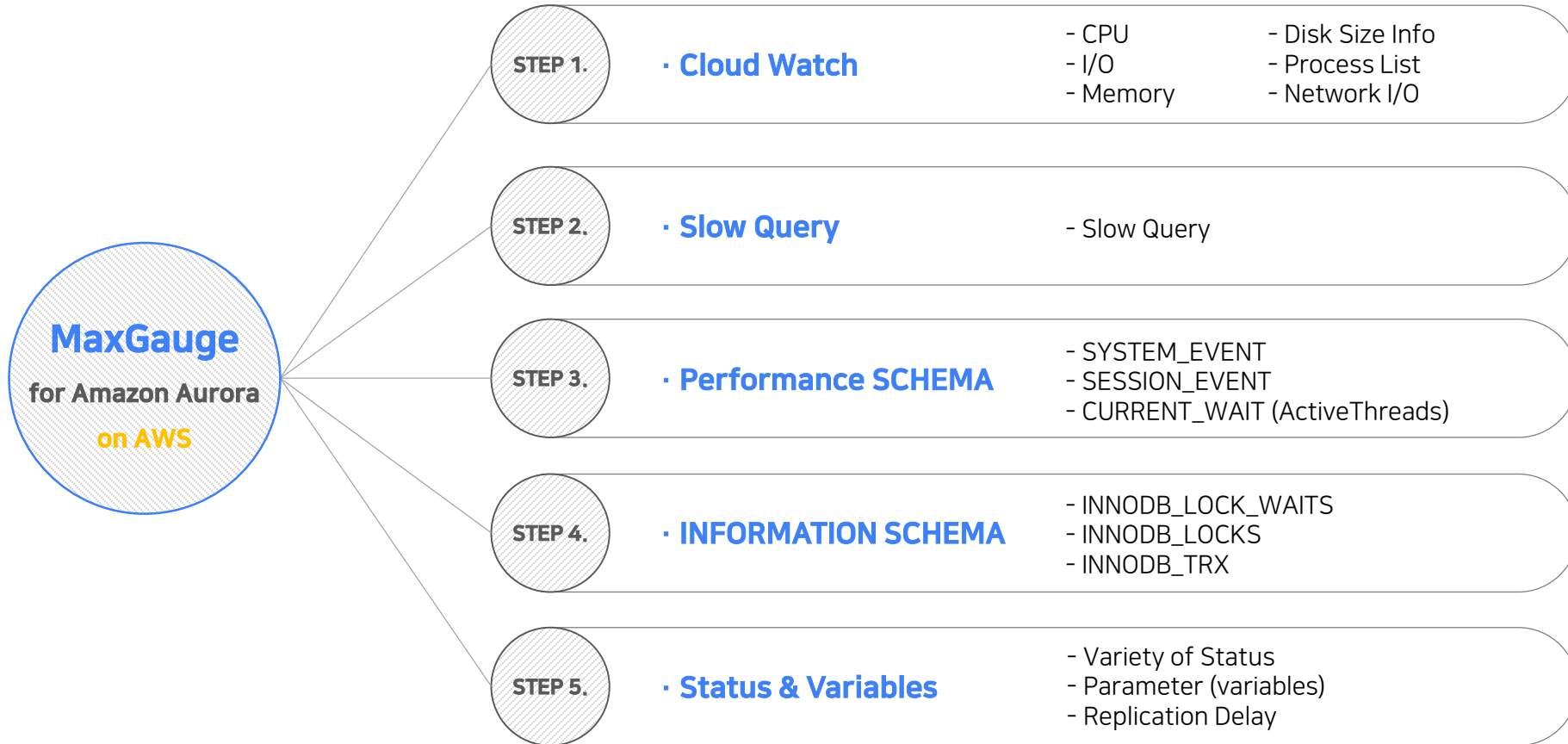




모니터링 |
Real-Time Monitoring

사후분석 |
Performance Analysis

성능 수집 데이터 종류





FEATURE

REAL-TIME MONITOR
ADMIN
PERFORMANCE ANALYZER



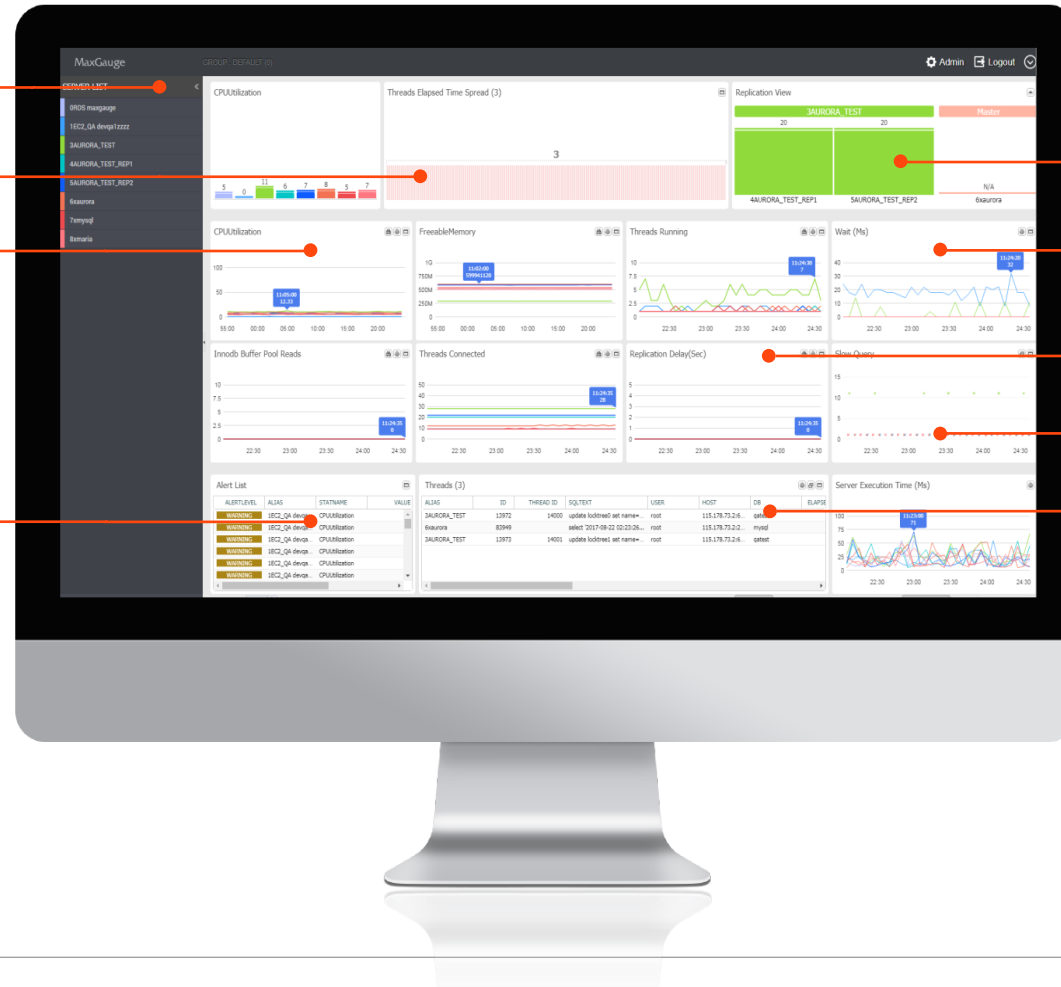
직관적이고 쉬운 사용자 인터페이스

서비스 그룹별 View

Thread 지표 시각화 영역

Multi Node 성능지표 통합 모니터링 영역

실시간 Event 이력



Replication View

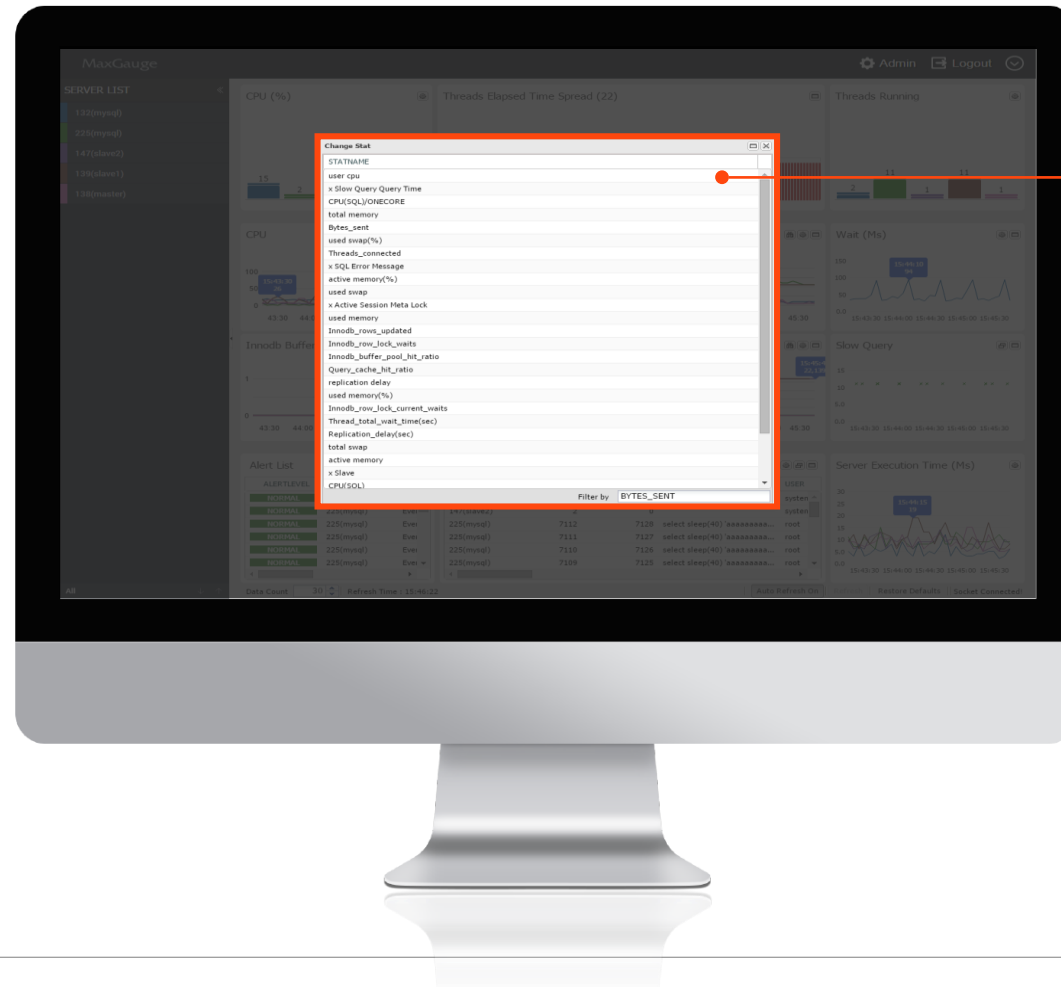
Wait 모니터링 영역

Replication 모니터링 영역

Slow Query 모니터링 영역

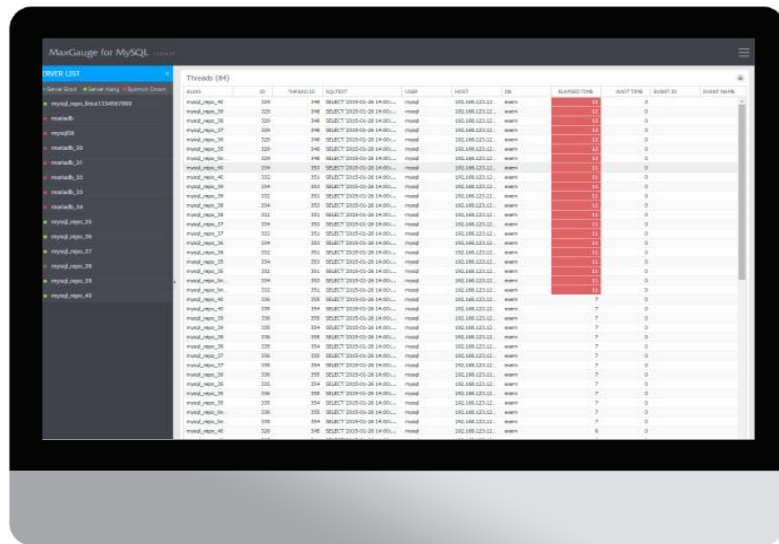
Active Thread 및 수행 Query 정보

운영 상황에 맞는 손쉬운 성능지표 배열

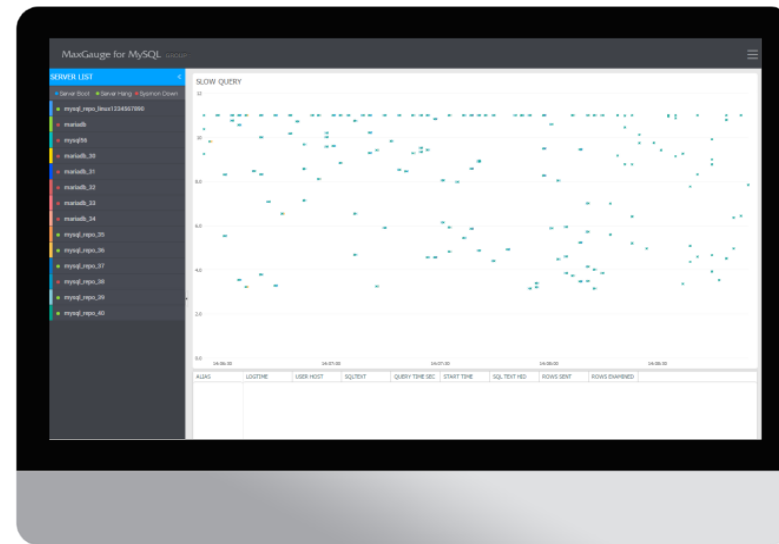


모니터링 성능지표 변경 가능

Threads, Slow Query 집중 모니터링 화면

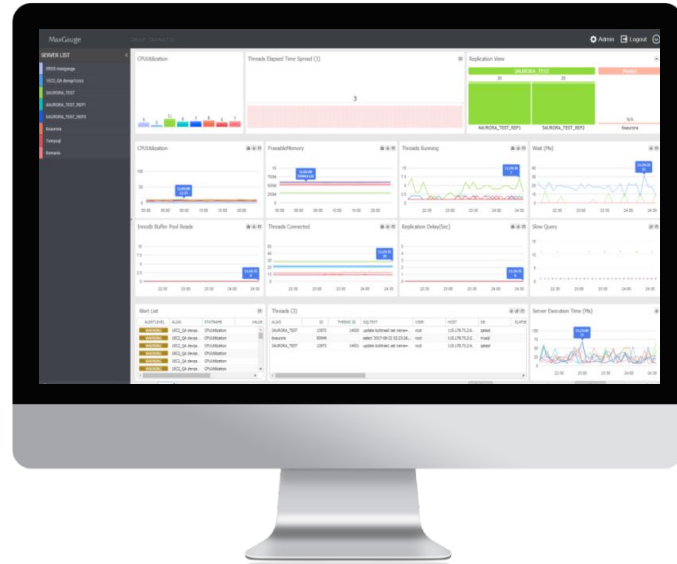


Threads View



Slow Query View

간편한 TOP-DOWN 접근방식을 통한 데이터베이스 모니터링



ALIAS	ID	THREAD ID	SQLTEXT	USER	HOST	DB	ELAPSE
testRDS	128	147	UPDATE warehouse SET w_y...	awsuser	ec2-52-197-173...	tpcc	
testEC2	1180	1205	COMMIT	root	52.197.173.96:5...	tpcc	
testRDS	130	149	CALL PAYMENT(1,2,1,2,@p_c...	awsuser	ec2-52-197-173...	tpcc	
testRDS	129	148	COMMIT	awsuser	ec2-52-197-173...	tpcc	
testRDS	127	146	COMMIT	awsuser	ec2-52-197-173...	tpcc	
testRDS	126	145	UPDATE warehouse SET w_y...	awsuser	ec2-52-197-173...	tpcc	



Thread Detail (132:mysql)Session ID: 226409 Thread ID: 226428) - 132(mysql) - Chrome

192.168.123.132:8070/realtime/thread_detail/index.html?server_no=1&session_id=226409&thread_id=226428&alias=132(mysql)

[ID = 226409] - 132(mysql) (update time : 15:05:02)

OS Stat

cpu : 5
user cpu : 1.53
used memory(%) : 97.17
used swap(%) : 0.73

Delta Info

Name	Value/Sec	Diff Value	Sigma Val
WAIT/IO/FILE/SQ/QUERY LOG	0	0	0
WAIT/SYNCH/COND/SQ/LC LOG MMAP::COND POOL	0	0	0
WAIT/SYNCH/MUTEX/SQ/LOCK OPEN	0	0	0
WAIT/SYNCH/MUTEX/MYSAHMRG/MYRG INFO::MUTEX	0	0	0
WAIT/SYNCH/MUTEX/SQ/PAGE::LOCK	0	0	0
WAIT/SYNCH/MUTEX/SQ/LOCK TABLE SHARE	0	0	0
WAIT/SYNCH/RWLOCK/SQ/MDL CONTEXT::LOCK WAITIN...	0	0	0
WAIT/IO/FILE/SQ/INT	0	0	0
WAIT/SYNCH/MUTEX/SQ/VERSION LOCK	0	0	0
WAIT/SYNCH/COND/SQ/DELAYED INSE...COND CLIENT	0	0	0
WAIT/SYNCH/MUTEX/SQ/RELAY LOG INFO::LOG SPACE L...	0	0	0
WAIT/SYNCH/MUTEX/SQ/LC LOG MMAP::LOCK SYNC	0	0	0

Name	Value
USER	root
HOST	localhost
DB	test
ELAPSED TIME(S)	30
COMMAND	Query
STATE	updating
SQLTEXT	delete from lock_test
EVENT NAME	
SOURCE	
WAIT TIME(S)	
SPINS	
OBJECT NAME	

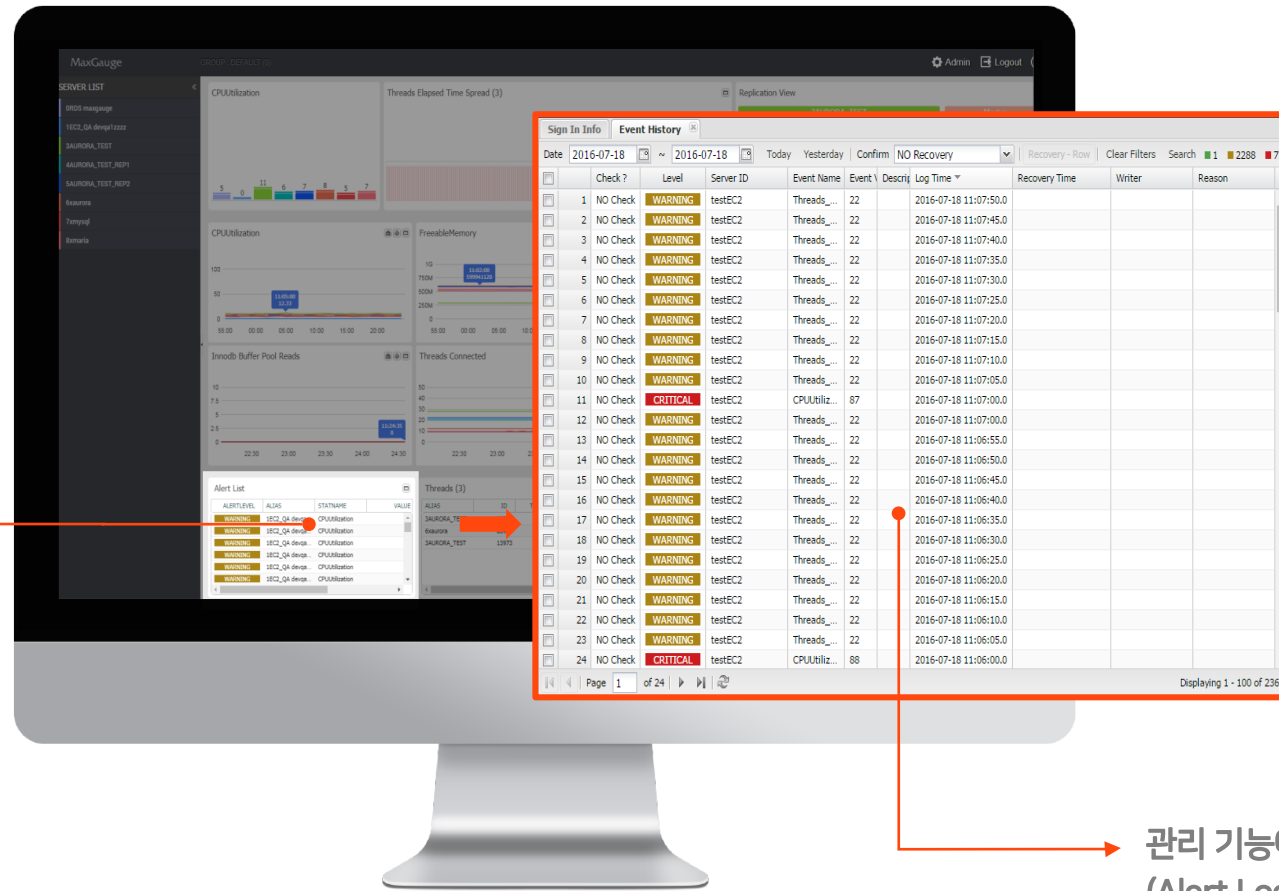
SQL Used

```
delete from lock_test
```

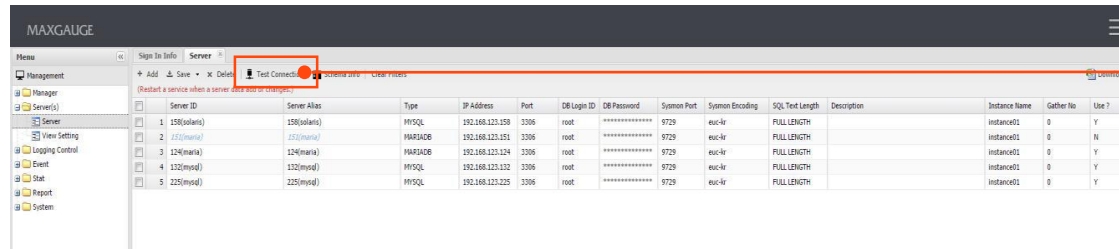
장시간 쿼리를 수행하는 Thread에 대하여 마우스
더블 클릭으로 Thread Detail 화면으로 연계

Current 대기 이벤트, DB지표와
현재 수행하는 SQL 정보 제공

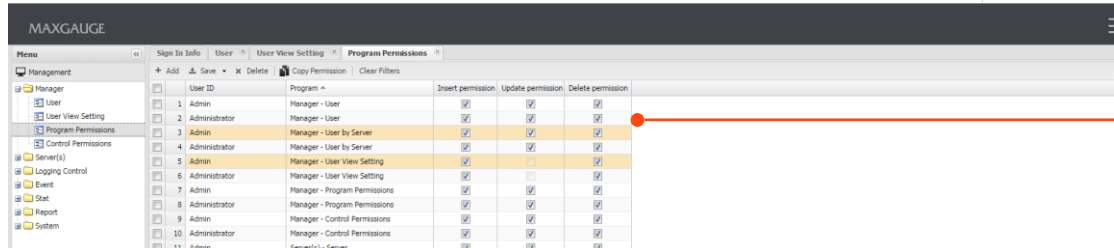
임계치 설정에 따른 Alarm 내역 확인 기능 제공



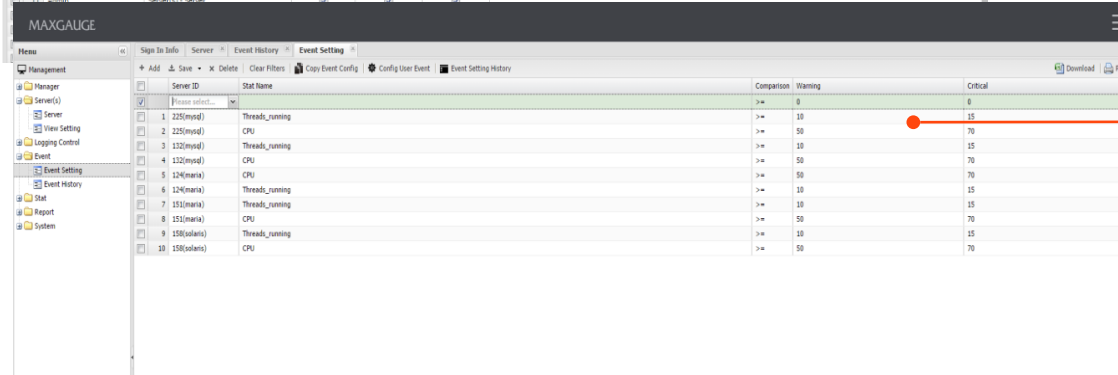
Admin 기능을 통한 간편한 Maxgauge 관리



간편한 서버 등록 & Test Connection

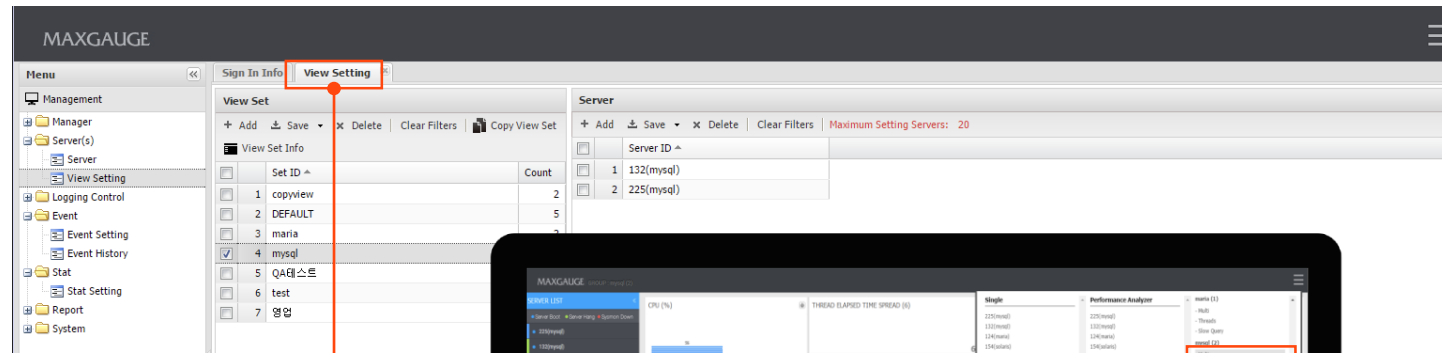


사용자 계정 추가 및 권한 부여

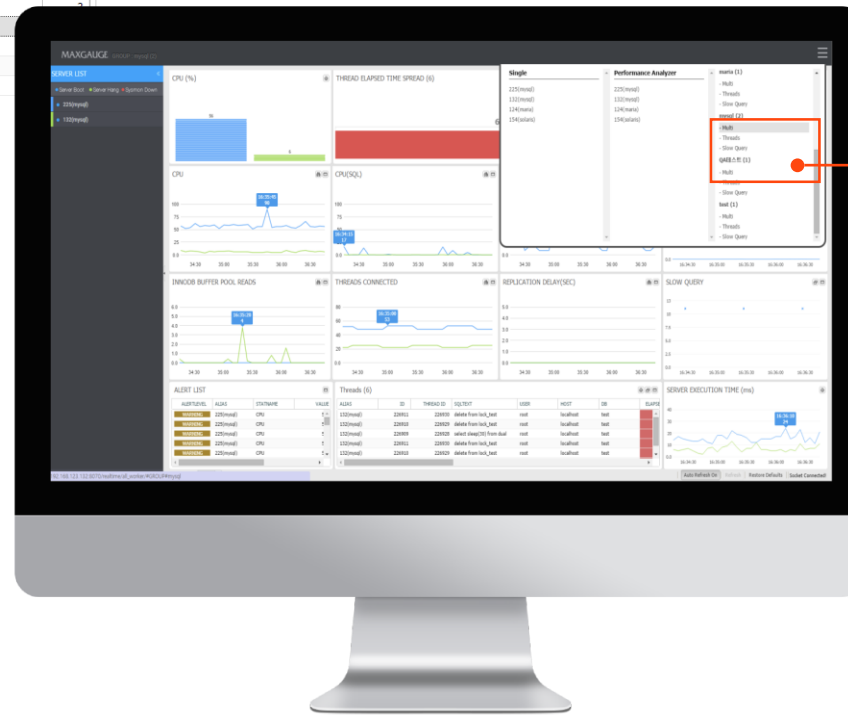


간편한 임계치 설정

작업 인스턴스별로 Admin Group 관리

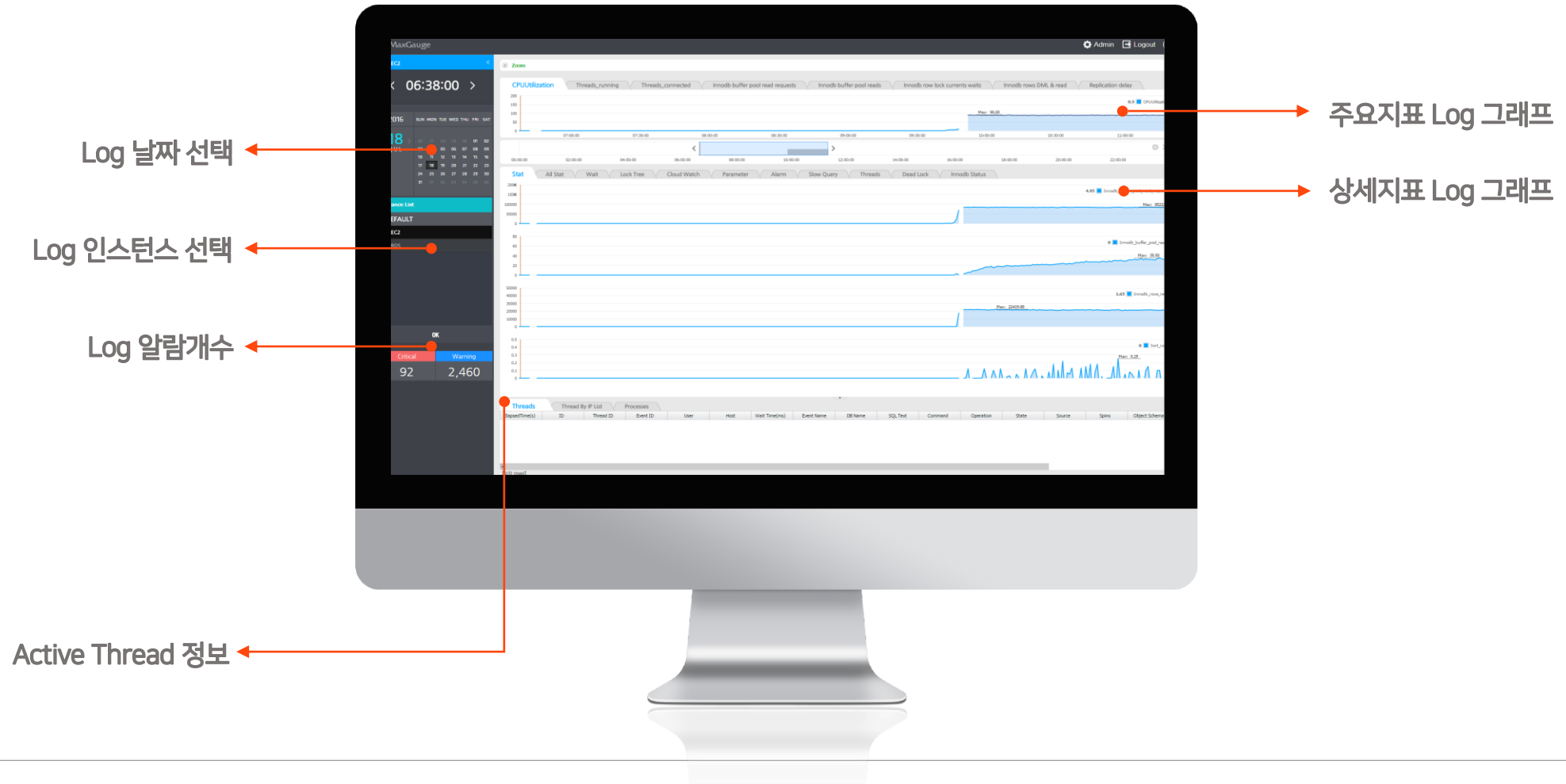


View Setting 탭에서
작업 인스턴스별로 Grouping 가능

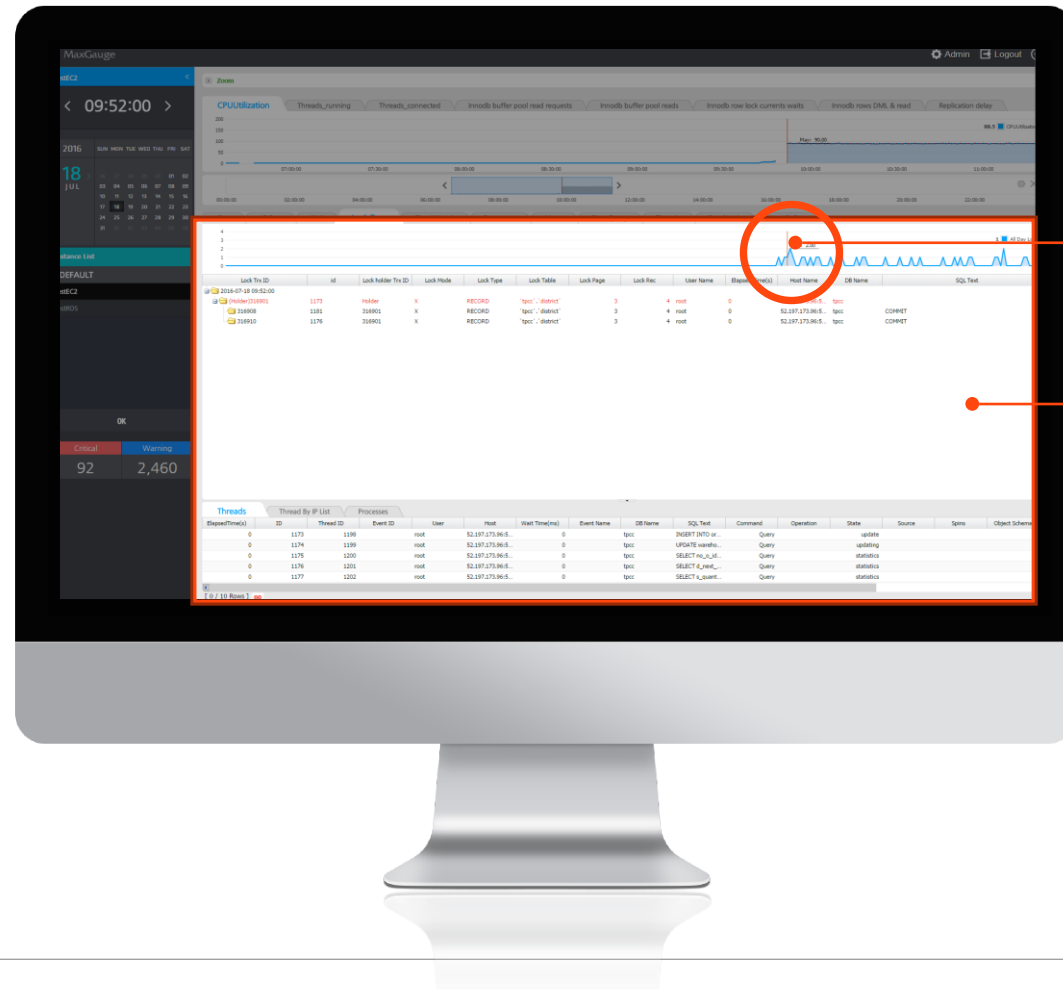


특정 Group
Multi View로 확인 가능

리파지토리에 저장된 데이터 분석을 위한 Performance Analyzer



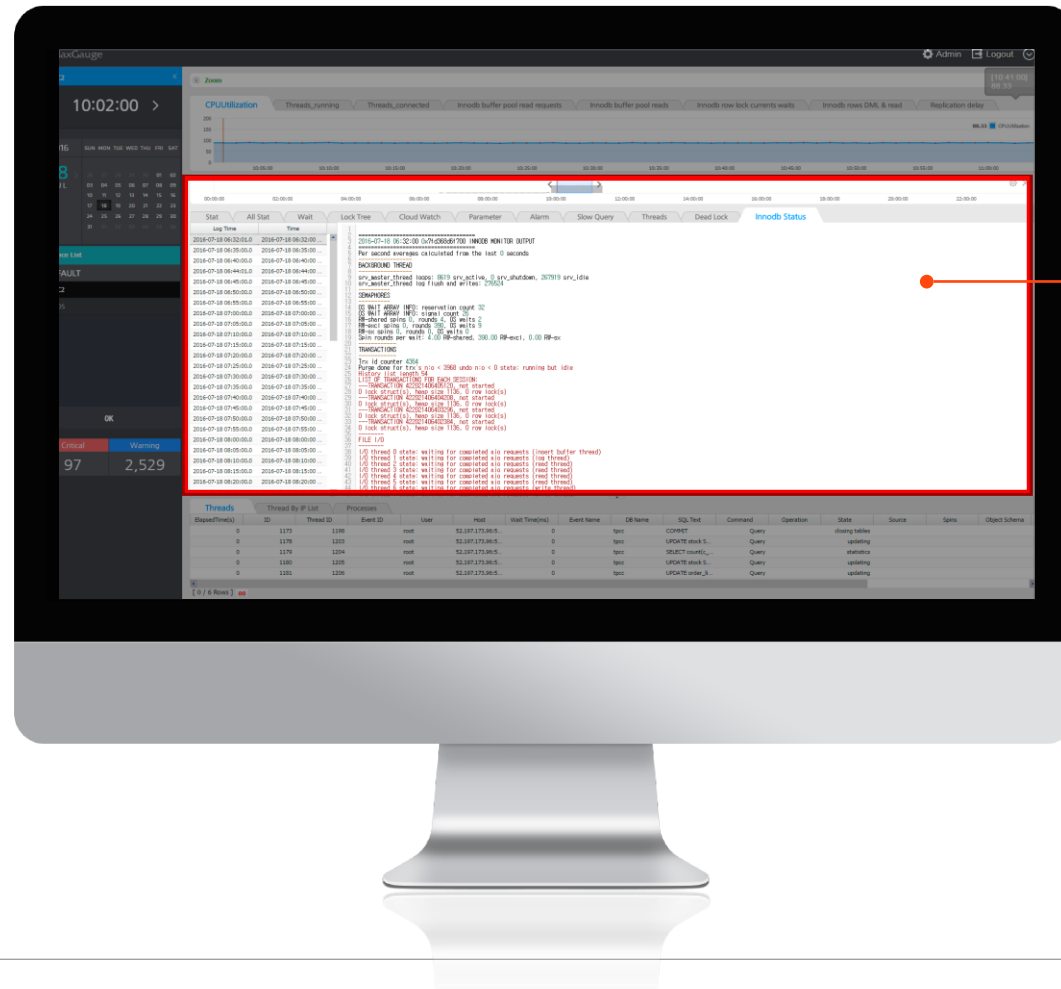
해당 시점의 LOCK 발생 내역을 트리 구조로 제공



특정 시간 때 드래그를 통해 상세 분석 가능

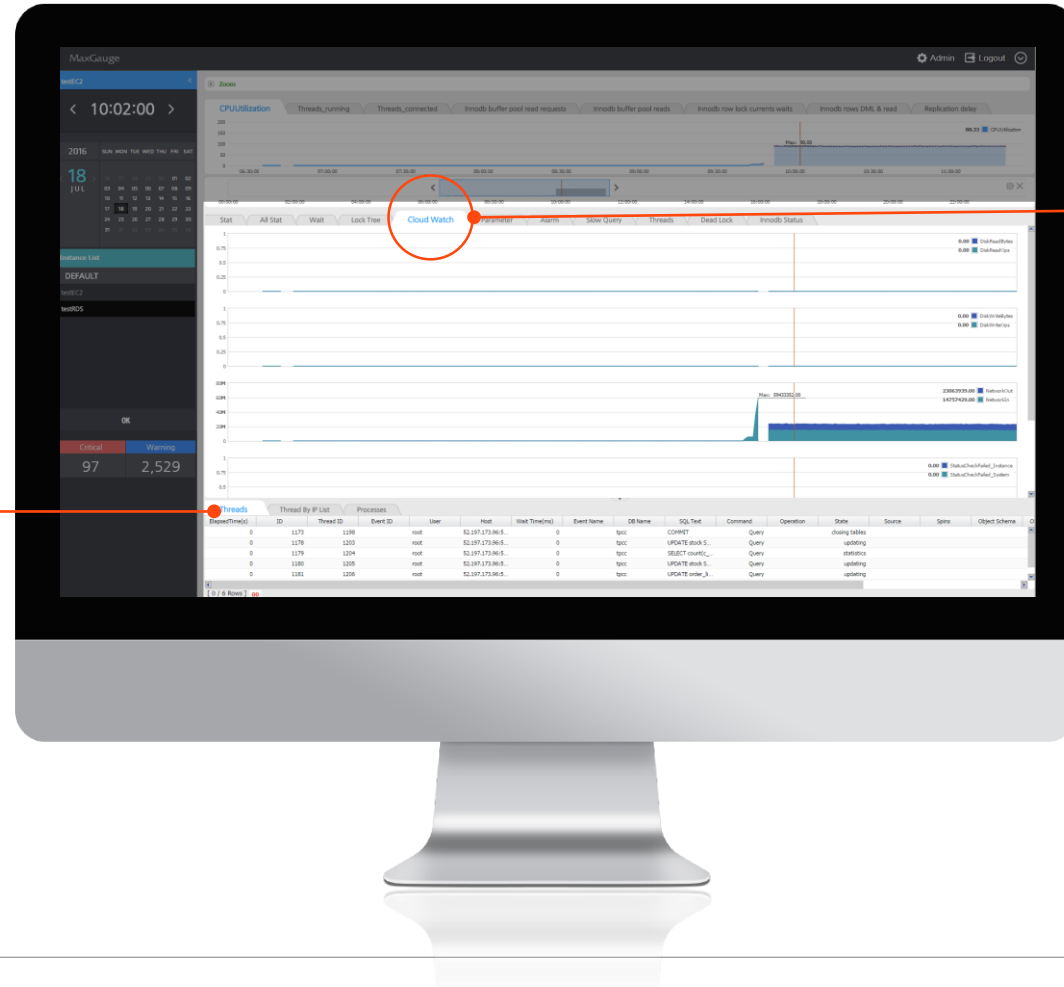
특정 시간 때의 Lock Tree 발생 확인 가능

InnoDB Status 정보 실시간 제공



InnoDB Status를
5분 주기로 저장

Cloud Watch 로그수집 및 DB 세션 연계 분석



Cloud Watch 로그를 수집하고 조회 가능

Cloud Watch 지표와 DB 세션 수행 이력과 연계하여 문제 시점 분석 정보 제공

MySQL/InnoDB Internals Monitoring

The dashboard is titled "MaxGauge" and includes "Admin" and "Logout" links in the top right. It is divided into two main sections: MySQL and InnoDB.

MySQL Section

- Session:** Connected Users (8 EA), Running Users (4 EA), Active (50%).
- User list:** Lists IP addresses such as 192.128.123.45, 192.128.123.21, 192.128.123.36, 192.128.123.35, 192.128.123.212, 192.128.123.135, 192.128.123.107, and 192.128.123.143.
- Thread:** Client Thread (Active: 4, Inactive: 4). Thread Cached (Size: 8, Cached: 2). Query Cached (Type: ON, Size: 215.32 KB, Free: 26.48 KB, Queries: 3.21 KB). Sort Buffer (512 KB). Single Pass Sort (65%). QC Hit Rate (85%).
- Handler:** A central vertical bar showing activity for Write (1,231 row/s), Update (2,194 row/s), Delete (215 row/s), Index Scan (73 row/s), and Full Scan (73 row/s).

InnoDB Section

- Storage Engine:** Buffer Pool (128 MB, 54 KB Running Users, Hit Rate: 65%). Additional Memory Pool (Doublewrite Buffer: OFF, Adaptive Hash: ON). Undo (Trx ID: 23705568, History list length: 368, Readview: 10). Log Buffer (Total: 16.0 MB, LSN: 746467307).
- Disk:** Tablespace (System, User - No Information Available). Undo Segment (Count: 45 EA). Log Files (Groups: 2 EA, 128 MB each).

System Status: CPU 73%, Free Memory 473 MB / 700 MB.



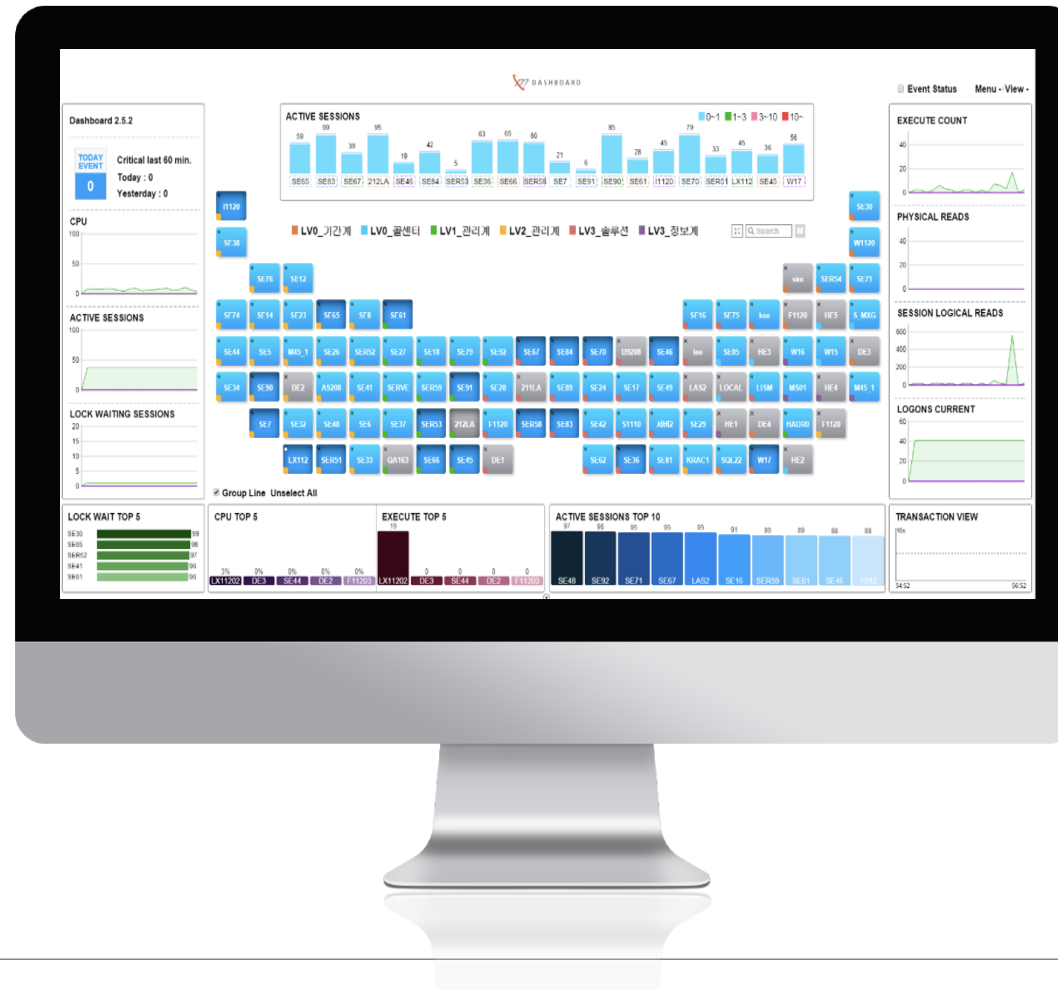
Enterprise Management System

ALERT & MONITORING
EXEM DASHBOARD



EXEM Dashboard 를 통한 여러 Instance를 한눈에 Health Check!

EXEM Dashboard 3.0은 실시간으로 최대 140개의 Instance를 한 화면에 구성할 수 있는 관제 화면으로
실시간 지표와 함께 장애 상황을 모니터 할 수 있습니다.



Real-Time Monitor / PA로 이어지는 강력한 연계 기능 제공

Dashboard에서 오른쪽 마우스 클릭으로 특정 인스턴스의 Real-Time Monitor 또는 Performance Analyzer로 이동이 가능합니다.



Dashboard 에 등록된 모든 인스턴스의 이벤트 이력/통계 내역을 제공

등록한 모든 인스턴스의 '이벤트 이력' 과 '이벤트 통계' 를 확인할 수 있습니다.

The screenshot displays the EXEM Dashboard interface. The main area is a table listing server events. The table has columns for instance ID, event type, status, IP address, error code, event time, and event details. A sidebar on the left contains a navigation menu. A pop-up window is visible at the bottom, showing details for a specific event.

관리 메뉴 ← (Management Menu)

서버 목록 List → (Server List)

등록한 인스턴스의 과거 이벤트 이력 List → (List of past events for registered instances)

이벤트 이력 확인 ← (Event History Confirmation)

현재 발생한 이벤트 이력 List → (List of currently occurring event history)



Replication





Replication

Aurora Replica

- Aurora DB Cluster에서 독립된 엔드포인트
- 읽기 연산을 조정하여 가용성을 높임
- 최대 복제본 수는 15개

<실습>

1. Amazon RDS – Create Aurora replica
2. Amazon RDS – Create cross region read replica
3. On- premise



Replication

RDS > Instances > e1

e1

Summary

Engine Aurora MySQL 5.6.10a	DB instance class info db.t2.small	DB instance status available	Pending maintenance none
--------------------------------	---	---------------------------------	-----------------------------

Instance actions ▼

- Create aurora replica
- Failover
- Create cross region read replica
- Promote read replica
- Take snapshot
- Restore to point in time
- Create clone
- Migrate latest snapshot
- Modify
- Stop
- Reboot
- Delete

CloudWatch (28)

Legend: e1

Add instance to compare Monitoring

CPU Utilization (Percent)

DB Connections (Count)

Freeable Memory (MB)

Network Receive Throughput



Replication – Create Aurora replica

RDS > Instances > e1 > Create read replica

Create Aurora replica

You are creating an Aurora replica DB instance in the source's DB cluster.

Network & Security

Availability zone
The EC2 Availability Zone that the database instance will be created in.

ap-northeast-2a

Publicly accessible

Yes
EC2 instances and devices outside of the VPC hosting the DB instance will connect to the DB instances. You must also select one or more VPC security groups that specify which EC2 instances and devices can connect to the DB instance.

No
DB instance will not have a public IP address assigned. No EC2 instance or devices outside of the VPC will be able to connect.

Instance specifications

DB instance class
Contains the compute and memory capacity of the DB instance.

db.t2.small — 1 vCPU, 2 GiB RAM

Settings

Aurora replica source
Source DB instance Identifier

e1 (DB Cluster: e1-cluster-1)

DB instance identifier
DB instance identifier. This is the unique key that identifies a DB instance. This parameter is stored as a lowercase string (e.g. mydbinstance).

Failover

Priority
Select a priority tier for this instance.

- During failover, Amazon RDS will promote the replica with the highest priority to primary
- If there is contention between 2 or more replicas in the same priority tier, then Amazon RDS will promote the replica that is the same size as the primary instance
- Priority tier logic: Tier 0 > Tier 1 > ... > Tier 15

No preference

Database options

Database port
Port number on which the database accepts connections.

3306

DB parameter group
Database parameter group to associate with this DB instance

default.aurora5.6

Monitoring

Enhanced monitoring

Enable enhanced monitoring
Enhanced monitoring metrics are useful when you want to see how different processes or threads use the CPU.

Disable enhanced monitoring

Maintenance

Auto minor version upgrade
Specifies if the DB instance should receive automatic engine version upgrades when they are available.



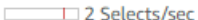



Yes

No

Cancel **Create Aurora replica**



Replication – Create Aurora replica

<input type="radio"/>	e1	Aurora MySQL	 available	 12.03%	 2 Selects/sec	none	db.t2.small	vpc-d38a7bbb	No	 writer
<input type="radio"/>	e1-replica	Aurora MySQL	 available			none	db.t2.small	vpc-d38a7bbb	No	 reader



Aurora MySQL DB 클러스터에 대한 이진 로깅 활성화

- binlog_format 파라미터를 ROW, STATEMENT 또는 MIXED로 설정
- MIXED는 특정 binlog 형식이 요구되는 경우를 제외하고 권장
- binlog_format 파라미터는 기본적으로 default.aurora5.6 클러스터 파라미터 그룹에 속하는 클러스터 수준 파라미터
- binlog_format 파라미터를 OFF에서 다른 값으로 변경하는 경우, Aurora DB 클러스터를 **재부팅**해야 변경 사항이 적용

aurora

Parameters										Cancel editing	Preview changes	Reset	Save changes		
<input type="text" value="binlog_"/>														< 1 >	
<input type="checkbox"/>	Name	Values	Allowed values	Modifiable	Source	Apply type	Data type	Description							
<input type="checkbox"/>	binlog_checksum	<input type="text"/>	NONE, CRC32	true	engine-default	dynamic	string	When enabled, this variable causes the master to write a checksum for each event in the binary log.							
<input type="checkbox"/>	binlog_format	MIXED	ROW, STATEMENT, MIXED, OFF	true	user	static	string	Binary logging format for replication							
<input type="checkbox"/>	binlog_row_image	MIXED OFF	full, minimal, noblob	false	engine-default	dynamic	string	Whether the server logs full or minimal rows with row-based replication.							
<input type="checkbox"/>	binlog_rows_query_log_events	ROW STATEMENT	0, 1	false	engine-default	dynamic	boolean	When enabled, it causes a MySQL 5.6.2 or later server to write informational log events such as row query log events into its binary log.							



Replication – Create cross region read replica

Create cross region read replica

Creating a new cross region replica will also create a new Aurora cluster in the target region. If the replication is disrupted, you will have to set up again. It is recommended that you select "Multi-AZ Deployment" to ensure high availability for the target cluster. Note that you will incur charges based on the target region's pricing. Please see [Aurora pricing](#) for more details.

Network & Security

Destination region

The region in which the replica will be launched

US East (N. Virginia) ▼

Destination DB subnet group

default ▼

Publicly accessible

- Yes
EC2 instances and devices outside of the VPC hosting the DB instance will connect to the DB instances. You must also select one or more VPC security groups that specify which EC2 instances and devices can connect to the DB instance.
- No
DB instance will not have a public IP address assigned. No EC2 instance or devices outside of the VPC will be able to connect.

Encryption

Encryption

- Enable Encryption
Select to encrypt the given instance. Master key ids and aliases appear in the list after they have been created using the Key Management Service(KMS) console. [Learn More](#).
- Disable Encryption

Master key info

(default) aws/rds ▼

Description	Account	KMS key ID
Default master key that protects my RDS database volumes when no other key is defined	This account(638505169563)	d5424ac7-1306-41fd-822a-2252829e5e3c

Instance specifications

DB instance class

Contains the compute and memory capacity of the DB instance.

- Select one - ▼

Multi-AZ deployment

Specifies if the DB instance should have a standby deployed in another availability zone.

- Yes
- No

Settings

Read replica source

Source DB instance identifier

e1 (DB Cluster: e1-cluster-1) ▼

DB instance identifier

DB instance identifier. This is the unique key that identifies a DB instance. This parameter is stored as a lowercase string (e.g. mydbinstance).

DB Cluster Identifier

You may optionally specify an identifier for the DB Cluster that will be created along with your instance. If you do not provide one, a default identifier based on the instance identifier will be used. The cluster identifier is used in determining the cluster's endpoint.

Failover

Priority

Select a priority tier for this instance.

- During failover, Amazon RDS will promote the replica with the highest priority to primary
- If there is contention between 2 or more replicas in the same priority tier, then Amazon RDS will promote the replica that is the same size as the primary instance
- Priority tier logic: Tier 0 > Tier 1 > ... > Tier 15

No preference ▼

Database options

Database port

Port number on which the database accepts connections.

3306

Monitoring

Enhanced monitoring

- Enable enhanced monitoring
Enhanced monitoring metrics are useful when you want to see how different processes or threads use the CPU.
- Disable enhanced monitoring

Maintenance

Auto minor version upgrade

Specifies if the DB instance should receive automatic engine version upgrades when they are available.

- Yes
- No

Cancel

Create



Create cross region read replica

Creating a new cross region replica will also create a new Aurora cluster in the target region. If the replication is disrupted, you will have to set up again. It is recommended that you select "Multi-AZ Deployment" to ensure high availability for the target cluster. Note that you will incur charges based on the target region's pricing. Please see [Aurora pricing](#) for more details.

✔ **Your Read Replica creation has been initiated**

Please note that as part of cross region read replica creation we initiated a set up process, which entails creating a snapshot and transferring it to the destination region. Depending on the amount of data to be copied and the region you choose, this set up process could take several hours to complete. Click [here](#) to view your cross region read replica in the destination region. To learn more please refer to the [Cross Region Read Replica documentation](#)

Cancel

Close



RDS > Instances

Instances (2)

Filter instances

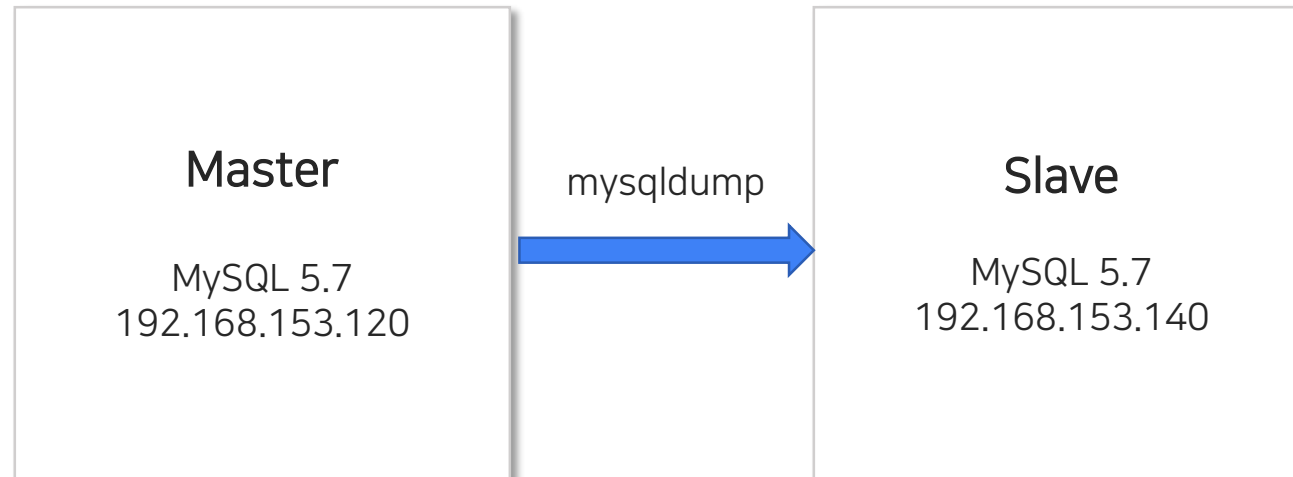
Instance actions Launch DB instance Restore from S3

DB instance	Engine	Status	CPU	Current activity	Maintenance	Class	VPC	Multi-AZ	Replication
<input type="radio"/> e1-replica2	Aurora MySQL	creating			none	db.t2.small	vpc-e7f7309f	2 Zones	reader
<input type="radio"/> e1-replica2-us-east-1d	Aurora MySQL	creating			none	db.t2.small	vpc-e7f7309f	2 Zones	reader





On-premise MySQL Replication (MySQL Ver. 5.7)





On-premise MySQL Replication (MySQL Ver. 5.7)

1. 준비사항

A. MySQL VM이미지 2개 준비

B. IP 고정 192.168.153.120 / 192.168.153.140

- vi /etc/sysconfig/network-scripts/ifcfg-eth1
- BOOTPROTO=static
- IPADDR=192.168.153.120

cf) 만약 동일한 이미지일 경우, MAC (HWADDR)가 상이한지 확인 후, 동일하면 변경해야 함



On-premise MySQL Replication (MySQL Ver. 5.7)

2. Master (192.168.153.120) 설정

A. mysqld stop된 상태에서 my.cnf 수정

```
[mysqld]  
log-bin=mysql-bin  
server-id=1
```

B. mysqld start

```
/etc/init.d/mysqld start
```

C. mysql 접속

```
mysql -uroot -p
```



On-premise MySQL Replication (MySQL Ver. 5.7)

2. Master (192.168.153.120) 설정

D. user 생성

```
grant replication slave on *.* to 'rep1'@'%' identified by 'rep1';
```

E. 변경사항 적용

```
flush privileges;
```

F. exit



On-premise MySQL Replication (MySQL Ver. 5.7)

3. Slave (192.168.153.140) 설정

A. mysqld stop된 상태에서 my.cnf 수정

```
[mysqld]  
log-bin=mysql-bin  
server-id=2  
read_only=1
```

B. mysqld start

```
/etc/init.d/mysqld start
```



On-premise MySQL Replication (MySQL Ver. 5.7)

4. Master (192.168.153.120)에서 데이터 덤프

A. 모든 테이블에 락 설정

```
flush tables with read lock;
```

B. 마스터 상태 보기 (* 결과값의 file, position 번호를 알아야 나중에 사용할 수 있음)

```
show master status;
```

C. root에서 mysqldump 실행

```
mysqldump -uroot -p --all-databases --lock-all-tables --events > mysql_dump.sql
```

D. 테이블 락을 풀음

```
unlock tables;
```



On-premise MySQL Replication (MySQL Ver. 5.7)

5. Slave (192.168.153.140) 데이터를 전달받음

A. Master 데이터덤프를 Slave로 transfer

```
scp ./mysql_dump.sql root@192.168.153.140:/home/mysql
```

B. import data dump

```
mysql -uroot -p <mysql_dump.sql
```

C. mysql 접속

```
mysql -uroot -p
```



On-premise MySQL Replication (MySQL Ver. 5.7)

5. Slave (192.168.153.140) 데이터를 전달받음

D. Change Master

```
change master to  
master_host='192.168.153.120',  
master_user='rep1',  
master_password='rep1',  
master_log_file='mysql-bin.000001',  
master_log_pos=491;
```

E. replication 시작

```
start slave;
```

F. show status

```
show slave status\WG
```

cf) UUID 에러 - 데이터 디렉토리 아래에 있는 auto.cnf를 삭제



On-premise MySQL Replication (MySQL Ver. 5.7)

6. 테스트 exem_i 데이터베이스에 t2 테이블 생성

```
create table t2 (c1 varchar(3));  
select * from t2; (Master, Slave 모두 해당)
```



Migration





Step1. Oracle 원본 데이터베이스 구성 & Step2. Aurora 대상 데이터베이스 구성



첫 번째 복제 작업 설정

단계 1: 시작

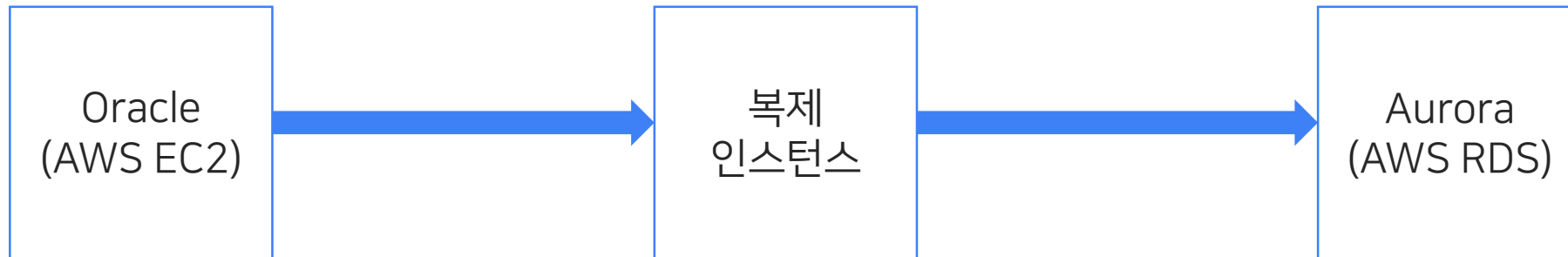
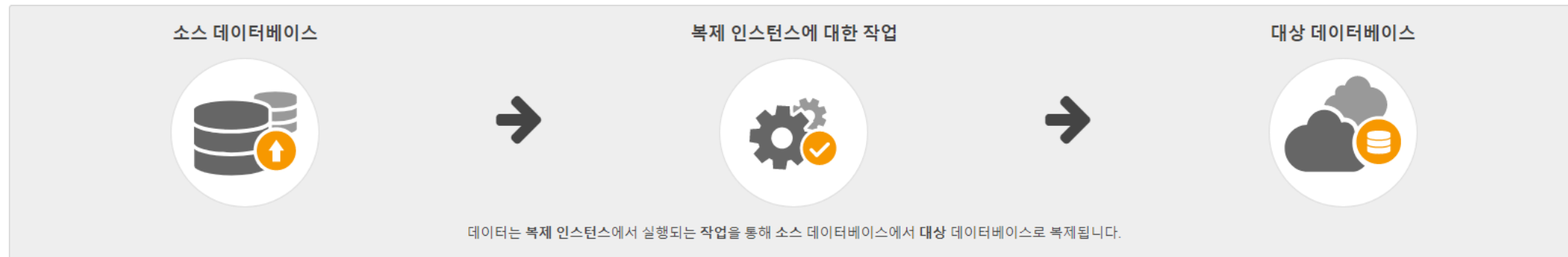
단계 2: 복제 인스턴스

단계 3: 데이터베이스 엔드포인트

단계 4: 작업

AWS Database Migration Service 소개

AWS Database Migration Service 작업에는 소스, 대상 및 복제 인스턴스가 최소한 하나씩 필요합니다. 소스는 데이터를 이동할 원본 데이터베이스이고 대상은 데이터를 이동할 대상 데이터베이스입니다. 복제 인스턴스는 마이그레이션 작업을 처리하며 VPC 내의 소스 및 대상 엔드포인트에 액세스할 수 있어야 합니다. 복제 인스턴스는 사용자의 성능 필요에 따라 다양한 크기로 제공됩니다. 다른 데이터베이스 엔진으로 마이그레이션하려는 경우 AWS Schema Conversion Tool를 사용하여 새로운 스키마를 생성할 수 있습니다. [AWS Schema Conversion Tool 다운로드](#)





Step3. 복제 인스턴스 생성

첫 번째 복제 작업 설정

단계 1: 시작

단계 2: 복제 인스턴스

단계 3: 데이터베이스 엔드포인트

단계 4: 작업

복제 인스턴스 생성

복제 인스턴스는 소스 데이터베이스와 대상 데이터베이스 간에 연결을 시작하고, 데이터를 전송한 다음, 첫 번째 데이터 로드 중 소스 데이터베이스에 발생하는 모든 변경 사항을 캐시합니다. 아래 필드를 사용하여 네트워크 및 보안 정보, 암호화 세부 정보 및 성능 특성을 포함한 새 복제 인스턴스의 파라미터를 구성하십시오. 마이그레이션이 완료되면 추가 사용 요금이 발생하지 않도록 복제 인스턴스를 종료하는 것이 좋습니다.

이름* replication-instance-2 ⓘ

설명* ReplicationInstance1 description ⓘ

인스턴스 클래스* dms.t2.medium ▼ ⓘ

복제 엔진 버전* 2.4.0 ▼ ⓘ

VPC* vpc-d38a7bbb ▼ ⓘ

다중 AZ* 아니요 ▼ ⓘ

공개 액세스 ⓘ

▶ 고급

▶ Maintenance

취소

이전

다음



Step3. 복제 인스턴스 생성

옵션		수행할 작업
기본	이름	여러 복제 인스턴스를 시작하거나 계정을 공유하려고 계획한 경우, 여러 복제 인스턴스를 빠르게 구분하도록 해주는 이름을 선택
	설명	용도에 대한 아이디어를 제공하고, 사고 예방을 할 수 있음
	인스턴스 클래스	AWS DMS는 상당히 많은 메모리와 CPU를 사용하기 때문에 대규모 데이터베이스가 있거나 많은 LOB 데이터 유형을 사용하는 경우, 큰 인스턴스를 설정하는 것을 추천함
	복제 엔진 버전	1.9.0 / 2.4.0
	VPC	원본, 대상 데이터베이스가 있는 동일한 VPC를 선택하는 것을 추천함. 데이터베이스 엔드포인트 중 하나 또는 모두가 VPC 밖에 있으면 AWS DMS 액세스를 허용하도록 방화벽 규칙을 수정해야 함
	다중 AZ	별도의 가용 영역에서 기본 및 보조 복제 인스턴스 생성하는 것으로 대다수의 경우 다중 AZ는 불필요함. 초기 데이터 로드에는 오랜 시간이 소요되고, 원본과 대상 데이터베이스 동기화를 상당한 시간동안 유지해야 하는 경우 다중 AZ 구성으로 마이그레이션 서버를 실행하는 것을 추천함.
공개 액세스	원본 또는 대상 데이터베이스가 복제 서버가 있는 VPC 외부에 있는 경우, 복제 인스턴스를 공개적으로 액세스할 수 있도록 해야함	



Step3. 복제 인스턴스 생성

▼ 고급

할당된 스토리지(GB)*

복제 서브넷 그룹* ⓘ

가용 영역* ⓘ

VPC 보안 그룹 ⓘ
launch-wizard-1
default
launch-wizard-4

KMS 마스터 키 ⓘ

키 ARN alias/aws/dms

▼ Maintenance

Auto minor version upgrade ⓘ

Maintenance window* : for hrs. (UTC+9) ⓘ



Step3. 복제 인스턴스 생성

옵션		수행할 작업
고급	할당된 스토리지 (GB)	스토리지는 주로 로그 파일과 캐시된 트랜잭션에서 소모됨 즉, AWS DMS는 스토리지를 많이 사용하지 않음. 대다수 스토리지의 할당량은 충분함. 스토리지 관련 지표에 유의하고 기본 할당량보다 더 많이 소모하게 되면 스토리지를 확장하는 것이 좋음.
	복제 서브넷 그룹	다중 AZ 구성으로 실행하는 경우, 최소 2개의 서브넷 그룹이 필요함
	가용 영역	대상 데이터베이스와 동일한 가용 영역에 기본 복제 서버를 배치함
	VPC 보안 그룹	보안 그룹을 사용하여 VPC에 대한 수신 및 송신이 제어 가능함
	KMS 마스터 키	AWS DMS를 통해 모든 데이터는 KMS 암호화 키를 사용하여 암호화됨. 기본적으로 사용자의 복제 서버에서 새 암호화 키를 생성하지만 필요에 따라 기존 키를 사용할 수 있음
	키 ARN	ARN(Amazon Resource Name) AWS 리소스를 고유하게 식별함
Maintenance	Auto minor version upgrade	Yes: MySQL DB 엔진의 부(minor) 버전 업그레이드가 있을 때 Aurora DB 클러스터가 자동으로 업그레이드를 실행
	Maintenance window	시스템 유지 관리를 실행하는 기간을 주 단위로 선택함.



Step4. Oracle 원본 엔드포인트 생성, Step5. Aurora 대상 엔드포인트 생성

소스 및 대상 데이터베이스 엔드포인트 연결



복제 인스턴스를 생성했습니다.

데이터베이스 엔드포인트는 온-프레미스, EC2, RDS 또는 클라우드에 있을 수 있습니다. 아래에서 연결 세부 정보를 정의하십시오. 나중에 오류를 방지하려면 여기서 엔드포인트 연결을 테스트하는 것이 좋습니다.

소스 데이터베이스 연결 세부 정보

엔드포인트 식별자* ⓘ

소스 엔진* ⓘ

서버 이름* ⓘ

포트* ⓘ

SSL 모드* ⓘ

사용자 이름* ⓘ

암호* ⓘ

SID* ⓘ

▶ 고급

테스트 실행

✓ 연결을 성공적으로 테스트했습니다.

대상 데이터베이스 연결 세부 정보

엔드포인트 식별자* ⓘ

대상 엔진* ⓘ

서버 이름* ⓘ

포트* ⓘ

SSL 모드* ⓘ

사용자 이름* ⓘ

암호* ⓘ

▶ 고급

테스트 실행

✓ 연결을 성공적으로 테스트했습니다.



Step4. Oracle 원본 엔드포인트 생성

옵션		수행할 작업
엔드포인트 식별자		이 식별자는 사용중인 AWS 계정이 소유하는 모든 복제 인스턴스에 대해 고유해야함
소스 엔진		Oracle 선택
서버 이름		On-premise의 경우 IP주소 입력, Amazon EC2 · RDS의 경우 Public DNS
포트		데이터베이스가 연결을 위해 수신하고 있는 포트 (1521)
SSL 모드		엔드포인트에서 연결 암호화를 활성화해야 하는 경우 Secure Socket Layer 적용 유형
사용자 이름		DMS에서 이 엔드포인트의 데이터베이스에 연결할 때 사용할 사용자 이름
암호		계정에 대한 암호
고급	추가 연결 속성	기타 엔드포인트 값((예: 사용자 이름, 암호 등)에 추가하여 사용할 추가 연결 속성
	KMS 마스터 키	데이터베이스 볼륨을 암호화하는 데 사용된 데이터 키를 보호하는 고객 마스터 키(CMK) 기본 키(예: alias aws/dms)를 선택하거나 IAM 콘솔의 [암호화 키] 탭을 사용하여 생성된 고객이 관리하는 CMK를 선택



Step5. Aurora 대상 엔드포인트 생성

대상 데이터베이스 연결 세부 정보

엔드포인트 식별자* ⓘ

대상 엔진* ⓘ

서버 이름*

포트* ⓘ

SSL 모드* ⓘ

사용자 이름* ⓘ

암호* ⓘ

▼ 고급

추가 연결 속성 ⓘ

KMS 마스터 키 ⓘ

ARN/ID* ⓘ

설명

계정 638505169563

키 ARN arn:aws:rds:ap-northeast-2:638505169563:db:edu1-m

✓ 연결을 성공적으로 테스트했습니다.

엔드포인트로 설정한 Amazon RDS DB 인스턴트에 연결할 수 없는 이유

- 사용자 이름과 암호 조합 확인
- 인스턴스의 Amazon RDS 콘솔에 표시된 **엔드포인트 값**이 AWS DMS 엔드포인트를 생성하는 데 사용된 엔드포인트 식별자와 동일한지 확인
- 인스턴스의 Amazon RDS 콘솔에 표시된 **포트 값**이 AWS DMS 엔드포인트에 할당된 포트와 동일한지 확인
- Amazon RDS DB 인스턴스에 할당된 **보안 그룹**이 AWS DMS 복제 인스턴스의 연결을 허용하는지 확인
- AWS DMS 복제 인스턴스와 Amazon RDS DB 인스턴스가 **동일한 VPC**에 없는 경우, DB 인스턴스를 공개적으로 액세스할 수 있는지 확인



Step5. Aurora 대상 엔드포인트 생성

옵션		수행할 작업
엔드포인트 식별자		이 식별자는 사용중인 AWS 계정이 소유하는 모든 복제 인스턴스에 대해 고유해야함
대상 엔진		Aurora 선택
서버 이름		Aurora 인스턴스의 writer 엔드포인트를 입력
포트		인스턴스에 할당된 포트
SSL 모드		Secure Socket Layer 적용 유형
사용자 이름		DMS에서 이 엔드포인트의 데이터베이스에 연결할 때 사용할 사용자 이름
암호		계정에 대한 암호
고급	추가 연결 속성	기타 엔드포인트 값((예: 사용자 이름, 암호 등)에 추가하여 사용할 추가 연결 속성
	KMS 마스터 키	데이터베이스 볼륨을 암호화하는 데 사용된 데이터 키를 보호하는 고객 마스터 키(CMK) 기본 키(예: alias aws/dms)를 선택하거나 IAM 콘솔의 [암호화 키] 탭을 사용하여 생성된 고객이 관리하는 CMK를 선택



Step6. 마이그레이션 작업 생성

첫 번째 복제 작업 설정

단계 1: 시작

단계 2: 복제 인스턴스

단계 3: 데이터베이스 엔드포인트

단계 4: 작업

작업 생성

작업에는 어떤 데이터가 소스에서 대상으로 이동하는지를 정의하는 테이블 매핑이 하나 이상 포함될 수 있습니다. 대상에 테이블이 없는 경우 자동으로 생성될 수 있습니다.

작업 이름* ⓘ

작업 설명* ⓘ

소스 엔드포인트 prodendpoint2

대상 엔드포인트 testendpoint2

복제 인스턴스 replication-instance-2

마이그레이션 유형* ⓘ

생성 시 작업 시작



Step6. 마이그레이션 작업 생성

옵션	수행할 작업
작업 이름	작업 이름 부여
작업 설명	작업 설명 입력
복제 인스턴스	단계별 생성 시 자동 부여 / 개별 생성 시 복제 인스턴스 선택
소스 엔드포인트	단계별 생성 시 자동 부여 / 개별 생성 시 소스 엔드포인트 선택
대상 엔드포인트	단계별 생성 시 자동 부여 / 개별 생성 시 대상 엔드포인트 선택
마이그레이션 유형	<ol style="list-style-type: none"> 1. 기존 데이터 마이그레이션 기존 데이터만 마이그레이션하고, 원본 데이터에 대한 변경 사항이 대상에 적용되지 않음 2. 기존 데이터 마이그레이션 및 지속적인 변경 사항 복제 기존 데이터를 마이그레이션 하는 동안 변경 사항을 캡처함. - 애플리케이션 종료 - 최종 변경사항을 대상에 반영되도록 함 - 외래 키/트리거 활성화 같은 작업 수행 - 새 대상 데이터베이스를 가리키는 애플리케이션 시작 3. 데이터 변경 사항만 복제 일반적으로 같은 유형의 마이그레이션에만 적용
생성 시 작업 시작	선택 시 작업을 즉시 시작시킴. 인스턴스가 로깅 수준을 변경할 수 있도록 작업 시작을 지연할 수도 있음



Step6. 마이그레이션 작업 생성

작업 설정

대상 테이블 준비 모드 아무 작업 안 함 ⓘ
 대상에서 테이블 삭제
 자르기

복제에 LOB 열 포함 LOB 열을 포함시키지 마십시오. ⓘ
 전체 LOB 모드
 제한적 LOB 모드

최대 LOB 크기(kb) ⓘ

검증 활성화 ⓘ

로깅 활성화

[고급 설정](#)



Step6. 마이그레이션 작업 생성

옵션	수행할 작업
대상 테이블 준비 모드	<p>로드 전에 대상 테이블이 준비되는 방식을 지정할 수 있음</p> <ol style="list-style-type: none"> 아무 작업 안함 (Do nothing) 테이블 구조는 그대로 유지되고, 기존 데이터는 테이블에 남음 이 옵션을 사용하여 여러 시스템의 데이터를 통합할 수 있음 대상에서 테이블 삭제 (Drop tables on target) 자동으로 대상을 생성하도록 하는 경우 대개 이 옵션을 사용 마이그레이션 하기 전에 마이그레이션 할 테이블을 삭제 및 재생성 자르기 (Truncate) 대상 시스템에서 일부 또는 전체 테이블을 미리 생성할 경우 이 옵션을 사용 로드하기 전에 대상 테이블을 truncate하고, 대상 테이블이 없는 경우, 자동으로 테이블을 생성
복제에 LOB열 포함	<ol style="list-style-type: none"> LOB 열을 포함시키지 마십시오 이중 마이그레이션 할 시 LOB 저장 방법을 재고해야 함. 전체 LOB 모드 크기에 상관 없이 모든 LOB를 마이그레이션 함. 한번에 하나씩 조각별로 마이그레이션되므로 속도가 느릴 수 있음 제한적 LOB 모드 메모리를 미리 할당하고 LOB 데이터를 대량으로 로드할 수 있음. 크기를 초과하는 LOB 파일은 잘리고, 로그 파일에 경고가 발행됨
최대 LOB 크기	제한적 LOB 모드에서는 최대 LOB 크기를 초과하는 LOB 열이 최대 LOB 크기로 잘림
검증 활성화	원본과 대상 간 데이터 비교 및 검증
로깅 활성화	선택 시 ClouWatch 로그 사용량에 대해 표준 요금으로 청구됨






Step6. 마이그레이션 작업 생성

원본에서 대상으로 마이그레이션 해야 하는 테이블을 지정
대다수의 경우 테이블 매핑을 사용자 지정해야 함

▼ 테이블 매핑

가이드 JSON

선택 규칙 ⓘ

여기서 스키마 이름은(는) 'SCOTT' 형식이며 테이블 이름은 'EMP2' 포함 형식입니다   

장소 ⓘ

스키마 이름 SCOTT ▼

테이블 이름의 형식 %

와일드카드 %를 사용하십시오.

작업 포함 ⓘ

필터 ⓘ

[열 필터 추가](#)

취소 [선택 규칙 저장](#)

+ 선택 규칙 추가

변환 규칙 ⓘ

+ 변환 규칙 추가



Step7. 마이그레이션 작업 모니터링

ID	상태	소스	대상	유형	완료 %	경과 시간	테이블 로드 상태	테이블 로드 중	테이블 대기 상태	테이블에서 오류
<input type="checkbox"/> task2	생성 중	prodendpoint2	testendpoint2	Full Load	<div style="width: 0%;"></div> 0	0	0	0	0	0
<input checked="" type="checkbox"/> task2	실행 중	prodendpoint2	testendpoint2	Full Load	<div style="width: 0%;"></div> 0	0	0	0	0	0
<input checked="" type="checkbox"/> task2	업로드 완료	prodendpoint2	testendpoint2	Full Load	<div style="width: 100%;"></div> 100	0m	1	0	0	0

task2

개요 | 작업 모니터링 | **테이블 통계** | 로그 | 평가 결과

테이블 데이터 삭제 후 다시 로드 ↻

필터:

스키마	테이블	로드 상태	삽입	삭제	업데이트	DDL	행 전체 로드	합계	검증 상태	검증 보류	검증 실패	검증 일시 중지됨	최종 업데이트 날짜
<input type="checkbox"/> SCOTT	EMP2	테이블 완료됨	0	0	0	0	14	14	활성화되지 않음	0	0	0	18. 1. 17. 오후 1:21



Step7. 마이그레이션 작업 모니터링

```
E1_aurora> show databases;
+-----+
| Database          |
+-----+
| information_schema |
| SCOTT              |
| TPCC               |
| aurora             |
| awsdms_control    |
| mysql              |
| performance_schema |
| tmp                |
| tpcc               |
+-----+

E1_aurora> use SCOTT
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A

Database changed

E1_aurora> show tables;
+-----+
| Tables_in_SCOTT   |
+-----+
| EMP2              |
+-----+
```




Step7. 마이그레이션 작업 모니터링

```
E1_aurora> select * from EMP2;
```

EMPNO	ENAME	JOB	MGR	HIREDATE	SAL	COMM	DEPTNO
7369	SMITH	CLERK	7902	1980-12-17 00:00:00	800.00	NULL	20
7499	ALLEN	SALESMAN	7698	1981-02-20 00:00:00	1600.00	300.00	30
7521	WARD	SALESMAN	7698	1981-02-22 00:00:00	1250.00	500.00	30
7566	JONES	MANAGER	7839	1981-04-02 00:00:00	2975.00	NULL	20
7654	MARTIN	SALESMAN	7698	1981-09-28 00:00:00	1250.00	1400.00	30
7698	BLAKE	MANAGER	7839	1981-05-01 00:00:00	2850.00	NULL	30
7782	CLARK	MANAGER	7839	1981-06-09 00:00:00	2450.00	NULL	10
7788	SCOTT	ANALYST	7566	1982-12-09 00:00:00	3000.00	NULL	20
7839	KING	PRESIDENT	NULL	1981-11-17 00:00:00	5000.00	NULL	10
7844	TURNER	SALESMAN	7698	1981-09-08 00:00:00	1500.00	NULL	30
7876	ADAMS	CLERK	7788	1983-01-12 00:00:00	1100.00	NULL	20
7900	JAMES	CLERK	7698	1981-12-03 00:00:00	950.00	NULL	30
7902	FORD	ANALYST	7566	1981-12-03 00:00:00	3000.00	NULL	20
7934	MILLER	CLERK	7782	1982-01-23 00:00:00	1300.00	NULL	10

```
14 rows in set (0.01 sec)
```



감사합니다

엑셈 DB 기술 연구소

엑셈아카데미 <http://exem-academy.com>

Onsite 세미나 및 컨설팅 문의 edu@ex-em.com