

ORACLE

# MySQL for DBA

---

**박혜선(Huishan.piao@oracle.com)**

MySQL Solution Engineer

Oracle MySQL GBU, JAPAC

Sep, 2021

# 목차

1. MySQL 소개
2. MySQL 아키텍처
3. MySQL 설치 및 시작
4. MySQL 서버 설정
5. MySQL 관리 및 모니터링
6. MySQL Best Practice



# MySQL 소개



# Why MySQL ?

## 세계적으로 가장 많이 보급된 오픈소스 데이터베이스

- LAMP 스택의 “M”
- WordPress, Magento, Moodle 등 개발 플랫폼과 통합

## Web 애플리케이션 개발의 디폴트 표준

- 가볍고 성능 좋고 높은 신뢰성 제공
- 대규모 OLTP 처리 부하에 최적화
- 복제로 인한 쉬운 확장성

## 개발/운영의 사용 용의성

- 심플한 아키텍처 및 관리 툴 제공
- 다양한 플랫폼 및 개발 언어 지원
- SQL + NoSQL 지원
- 성숙된 오픈소스 커뮤니티 생태계
- 듀얼 라이선스

<https://www.mysql.com/support/supportedplatforms/database.html>  
<https://dev.mysql.com/downloads/>

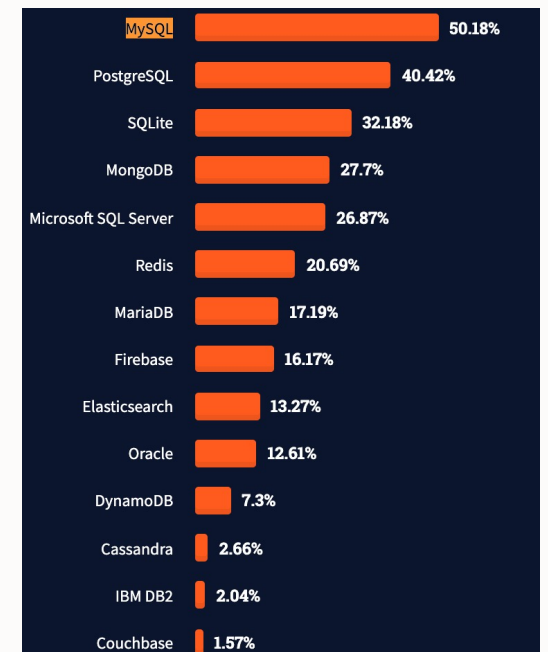
# 2021년 데이터베이스 순위

<https://db-engines.com/en/ranking>

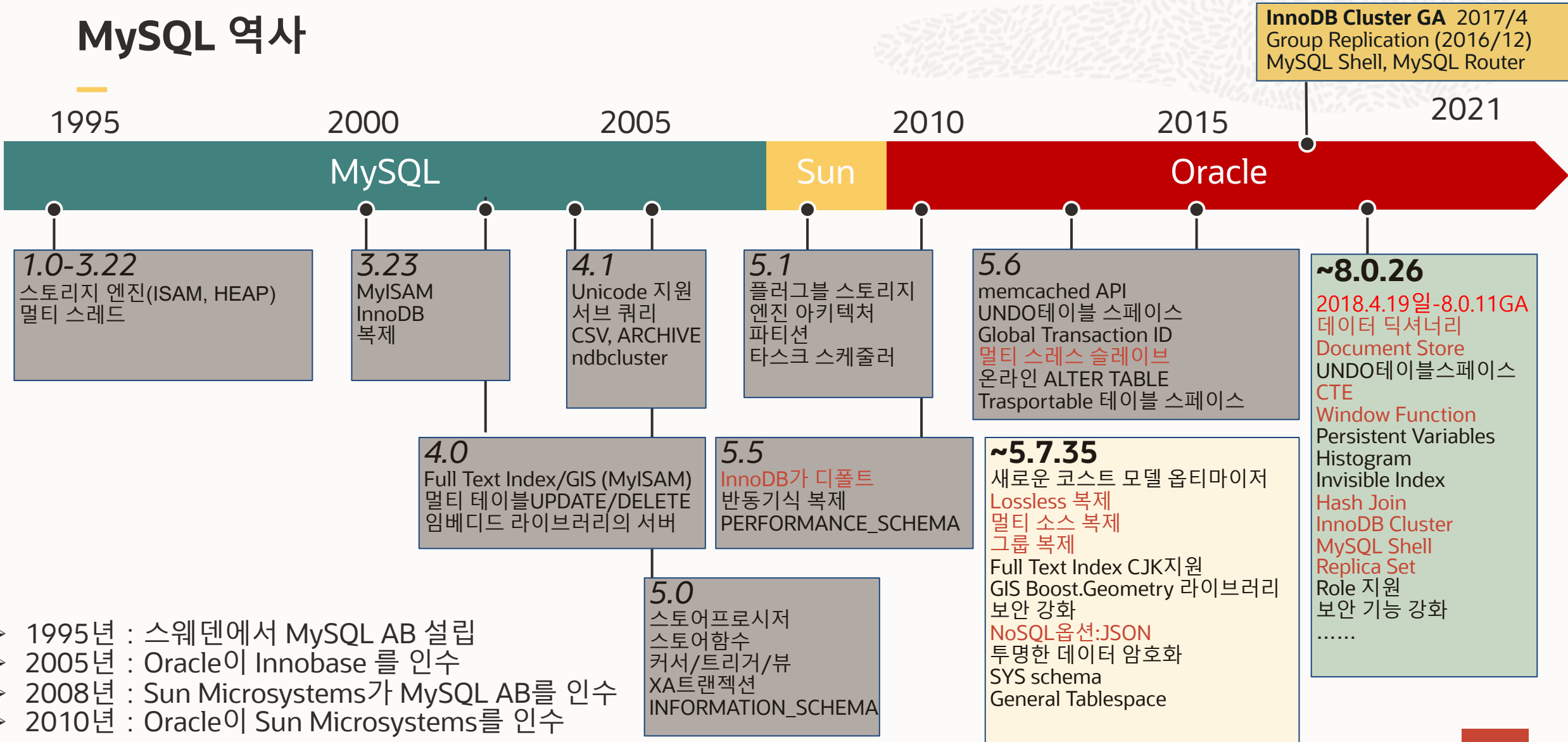
<https://insights.stackoverflow.com/survey/2021#section-most-popular-technologies-databases>

373 systems in ranking, August 2021

| Rank     |          |          | DBMS                   | Database Model                      | Score    |          |          |
|----------|----------|----------|------------------------|-------------------------------------|----------|----------|----------|
| Aug 2021 | Jul 2021 | Aug 2020 |                        |                                     | Aug 2021 | Jul 2021 | Aug 2020 |
| 1.       | 1.       | 1.       | Oracle +               | Relational, Multi-model <i>i</i>    | 1269.26  | +6.59    | -85.90   |
| 2.       | 2.       | 2.       | MySQL +                | Relational, Multi-model <i>i</i>    | 1238.22  | +9.84    | -23.36   |
| 3.       | 3.       | 3.       | Microsoft SQL Server + | Relational, Multi-model <i>i</i>    | 973.35   | -8.61    | -102.53  |
| 4.       | 4.       | 4.       | PostgreSQL +           | Relational, Multi-model <i>i</i>    | 577.05   | -0.10    | +40.28   |
| 5.       | 5.       | 5.       | MongoDB +              | Document, Multi-model <i>i</i>      | 496.54   | +0.38    | +52.98   |
| 6.       | 6.       | ↑ 7.     | Redis +                | Key-value, Multi-model <i>i</i>     | 169.88   | +1.58    | +17.01   |
| 7.       | 7.       | ↓ 6.     | IBM Db2                | Relational, Multi-model <i>i</i>    | 165.46   | +0.31    | +3.01    |
| 8.       | 8.       | 8.       | Elasticsearch          | Search engine, Multi-model <i>i</i> | 157.08   | +1.32    | +4.76    |
| 9.       | 9.       | 9.       | SQLite +               | Relational                          | 129.81   | -0.39    | +3.00    |
| 10.      | ↑ 11.    | 10.      | Microsoft Access       | Relational                          | 114.84   | +1.39    | -5.02    |



# MySQL 역사



- 1995년 : 스웨덴에서 MySQL AB 설립
- 2005년 : Oracle이 Innobase 를 인수
- 2008년 : Sun Microsystems가 MySQL AB를 인수
- 2010년 : Oracle이 Sun Microsystems를 인수

# MySQL 제품

<https://www.mysql.com/products/>

1 기능은 상용 제품에서만 제공합니다.

| 구분   | MySQL Standard Edition | MySQL Enterprise Edition | MySQL Cluster CGE |
|--|------------------------|--------------------------|-------------------|
| <b>MySQL 기능</b>  |                        |                          |                   |
| MySQL Database Server  | ○                      | ○                        | ○                 |
| MySQL Document Store   |                        | ○                        | ○                 |
| MySQL Connectors   | ○                      | ○                        | ○                 |
| MySQL Replication  | ○                      | ○                        | ○                 |
| MySQL Router   |                        | ○                        | ○                 |
| MySQL Partitioning   |                        | ○                        | ○                 |
| <b>MySQL Workbench<sup>1</sup></b>                           | ○                      | ○                        | ○                 |
| Storage Engine: MyISAM                                       | ○                      | ○                        | ○                 |
| Storage Engine: InnoDB                                       | ○                      | ○                        | ○                 |
| Storage Engine: NDB  |                        |                          | ○                 |
| Oracle Enterprise Manager for MySQL <sup>1</sup>             |                        | ○                        | ○                 |
| <b>MySQL Enterprise Monitor<sup>1</sup></b>                  |                        | ○                        | ○                 |
| MySQL Enterprise Dashboard <sup>1</sup>                      |                        | ○                        | ○                 |
| MySQL Enterprise Advisors <sup>1</sup>                       |                        | ○                        | ○                 |
| MySQL Query Analyzer <sup>1</sup>                            |                        | ○                        | ○                 |
| MySQL Replication Monitor <sup>1</sup>                       |                        | ○                        | ○                 |
| <b>MySQL Enterprise Backup<sup>1</sup></b>                   |                        | ○                        | ○                 |
| Hot backup for InnoDB <sup>1</sup>                           |                        | ○                        | ○                 |
| Full, Incremental, Partial, Optimistic Backups <sup>1</sup>  |                        | ○                        | ○                 |
| Full, Partial, Selective, Hot Selective restore <sup>1</sup> |                        | ○                        | ○                 |
| Encryption and Compression <sup>1</sup>                      |                        | ○                        | ○                 |
| Point-In-Time-Recovery <sup>1</sup>                          |                        | ○                        | ○                 |

| 구분  | MySQL Standard Edition | MySQL Enterprise Edition | MySQL Cluster CGE |
|---|------------------------|--------------------------|-------------------|
| <b>MySQL 기능</b>                                       |                        |                          |                   |
| <b>MySQL Enterprise Security<sup>1</sup></b>          |                        | ○                        | ○                 |
| MySQL Enterprise Authentication <sup>1</sup>          |                        | ○                        | ○                 |
| MySQL Enterprise TDE <sup>1</sup>                     |                        | ○                        | ○                 |
| MySQL Enterprise Encryption <sup>1</sup>              |                        | ○                        | ○                 |
| MySQL Enterprise Masking <sup>1</sup>                 |                        | ○                        | ○                 |
| MySQL Enterprise Firewall <sup>1</sup>                |                        | ○                        | ○                 |
| MySQL Enterprise Audit <sup>1</sup>                   |                        | ○                        | ○                 |
| <b>MySQL Enterprise Scalability<sup>1</sup></b>       |                        | ○                        | ○                 |
| Thread Pool <sup>1</sup>                              |                        | ○                        | ○                 |
| <b>MySQL Enterprise High-Availability<sup>1</sup></b> |                        | ○                        | ○                 |
| MySQL Group Replication                               |                        | ○                        | ○                 |
| MySQL InnoDB Cluster                                  |                        | ○                        | ○                 |
| <b>MySQL Cluster Manager<sup>1</sup></b>              |                        |                          | ○                 |
| 구성 & 프로비저닝 <sup>1</sup>                               |                        |                          | ○                 |
| 자동 스케일링 <sup>1</sup>                                  |                        |                          | ○                 |
| 관리 & 모니터링 <sup>1</sup>                                |                        |                          | ○                 |
| MySQL Cluster Geo-Replication                         |                        |                          | ○                 |
| <b>Oracle Premier Support</b>                         |                        |                          |                   |
| 24x7 지원   | ○                      | ○                        | ○                 |
| 무제한 지원  | ○                      | ○                        | ○                 |
| 지식 기반   | ○                      | ○                        | ○                 |
| 유지 보수 릴리스, 버그 수정, 패치, 업데이트                            | ○                      | ○                        | ○                 |
| MySQL 컨설팅 지원  | ○                      | ○                        | ○                 |



# MySQL 기타 제품

## MySQL Connectors

- 여러 개발 언어로 MySQL Server로 접속하기 위한 연결 제공

## MySQL Router

- 애플리케이션과 MySQL 서버사이의 투명한 라우팅을 제공하는 경량의 미들웨어
- 부하 분산 및고가용성 목적으로 사용됨
- MySQL InnoDB Cluster의 구성 요소

## MySQL Shell

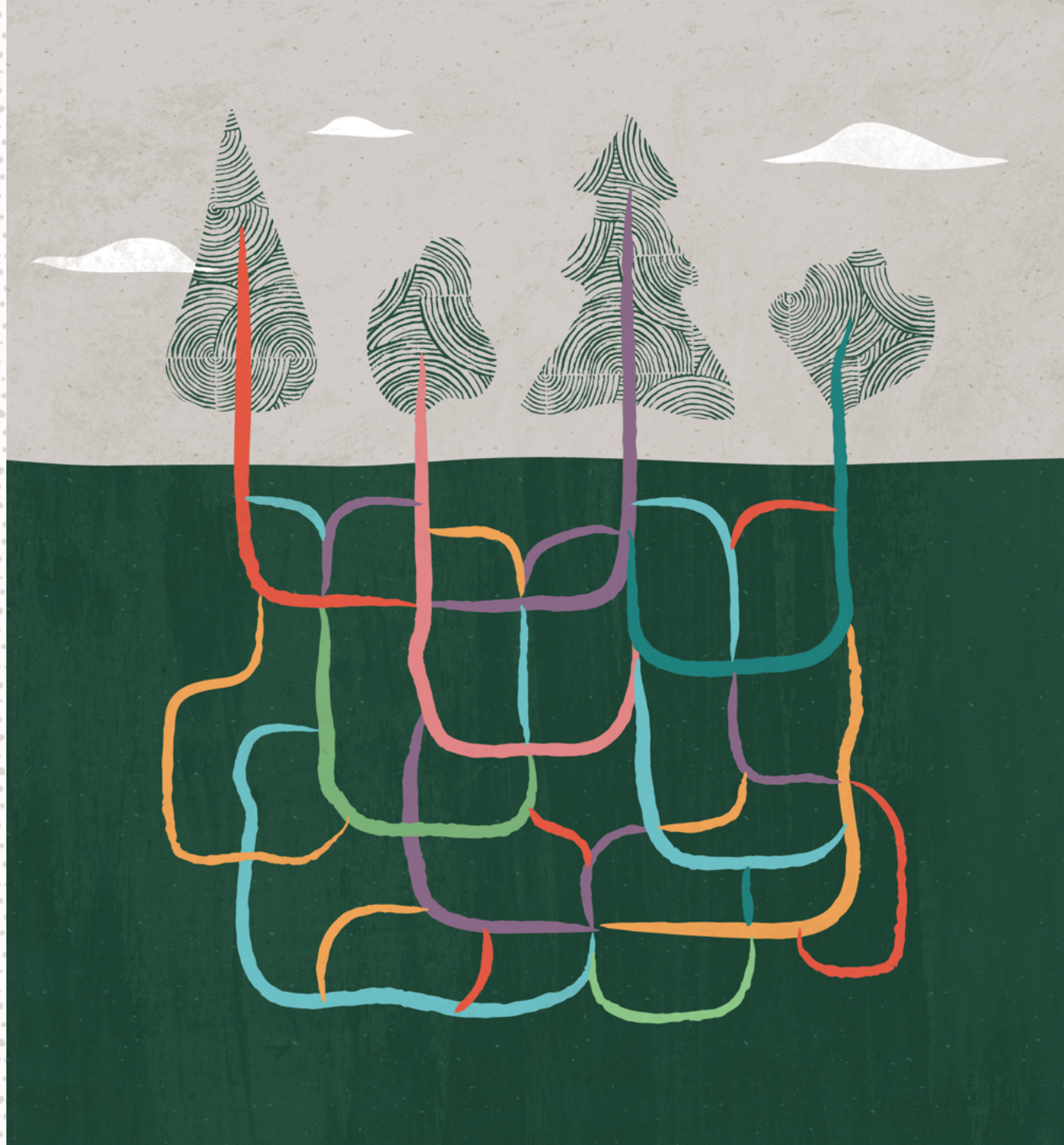
- JavaScript, Python, SQL모드를 지원하는 개발 관리 툴
- X-Protocol/Standard Protocol 지원
- MySQL InnoDB Cluster 관리 API

## MySQL on Windows (Installer & Tools)

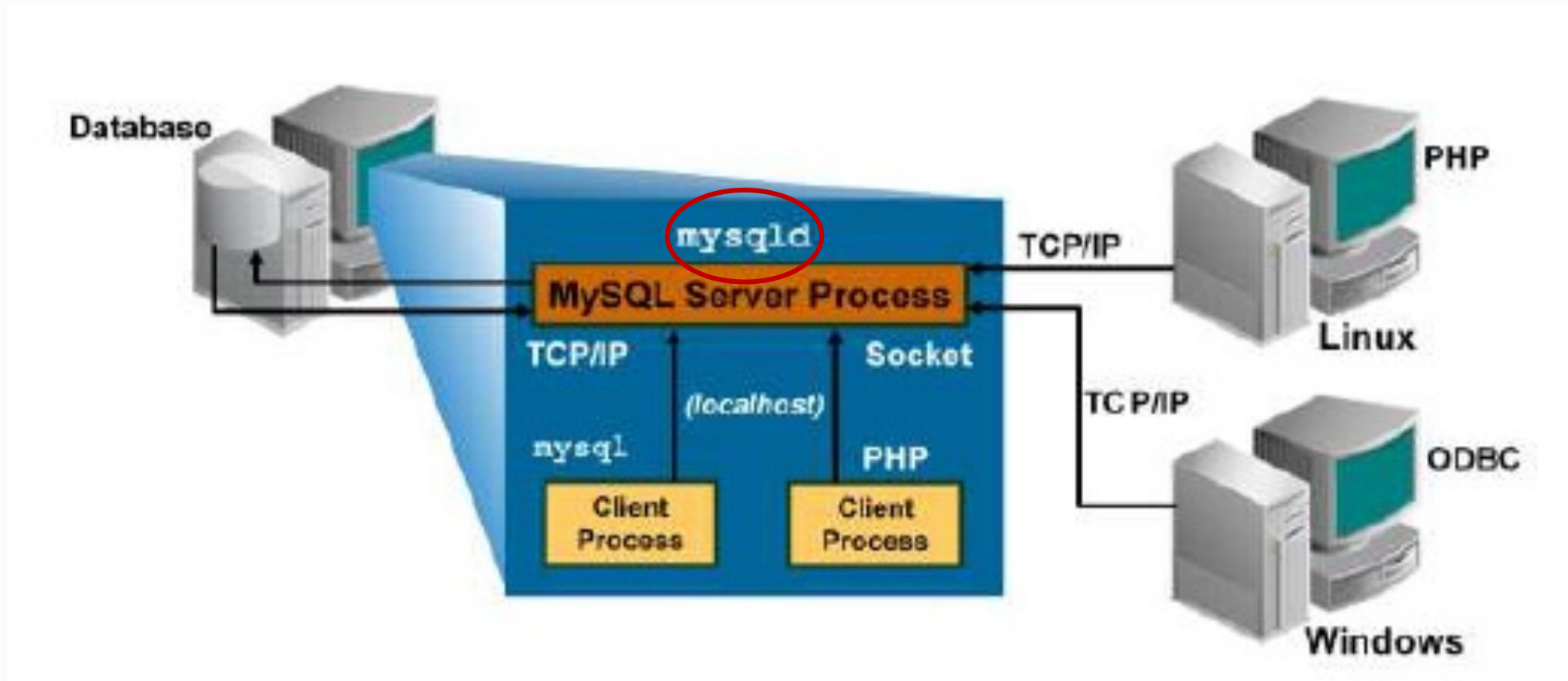
- Windows용 설치 프로그램을 사용하면 MySQL Server외에 MySQL for Excel등 Windows환경에 특화된 툴도 포함하여 설치할 수 있음



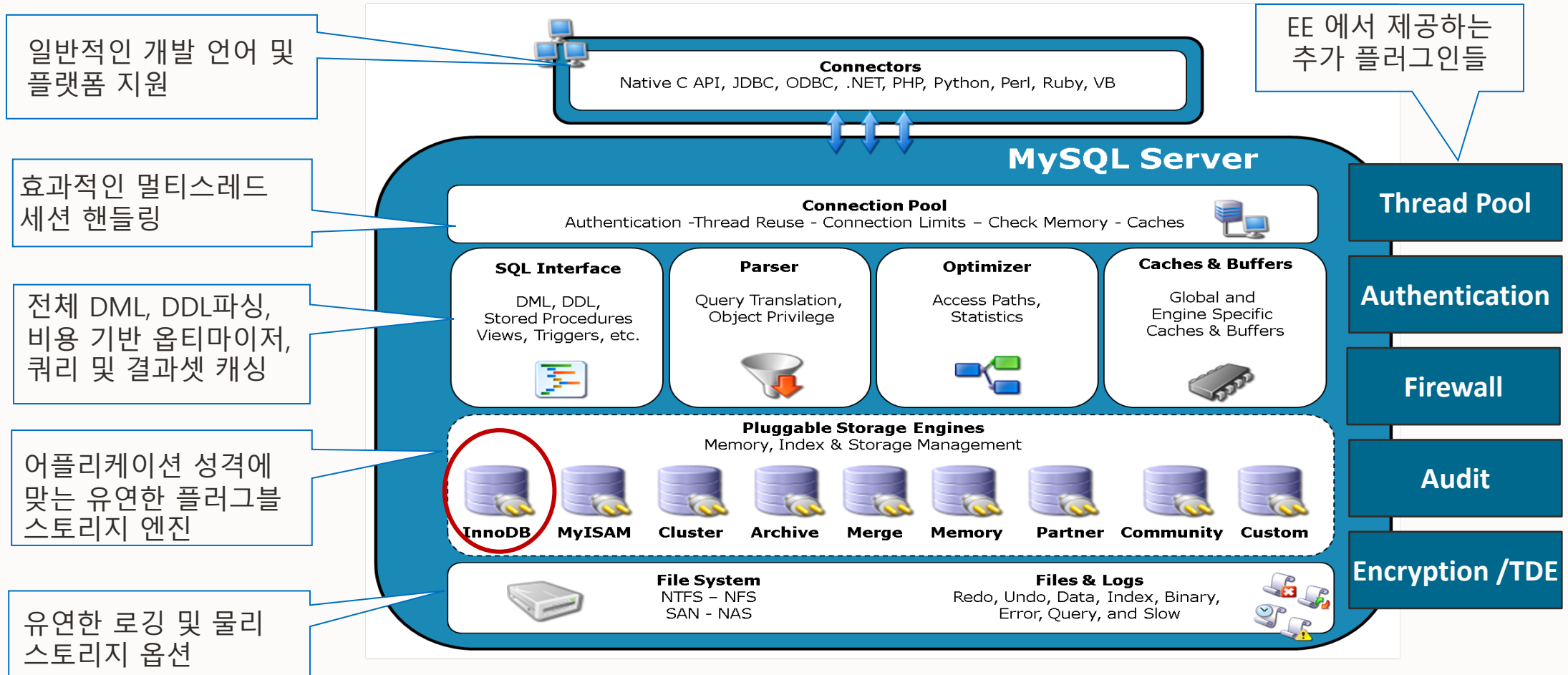
# MySQL 아키텍처



# MySQL High Level Client/Server Model



# MySQL Server 아키텍처



# InnoDB Storage Engine

## ANSI/ISO표준 ACID 트랜잭션 완전히 지원

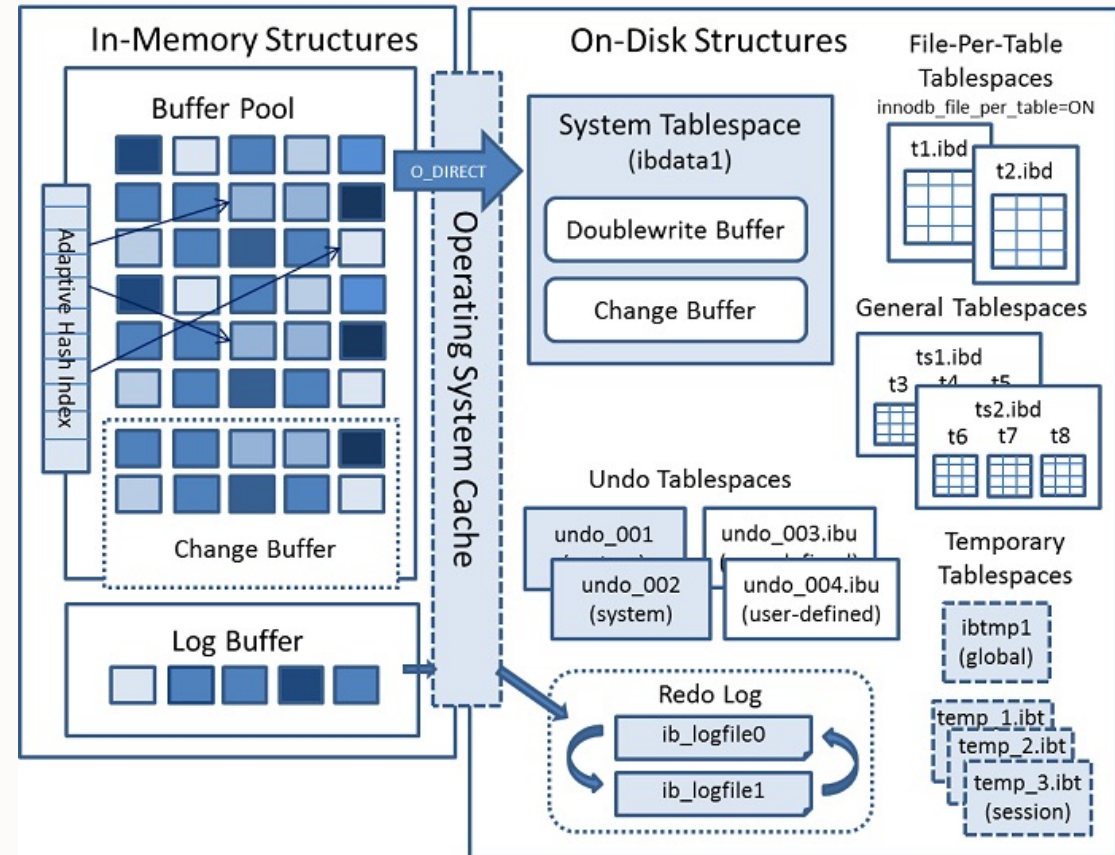
- 4가지 격리 레벨 지원(RR 디폴트)
- Commit/rollback/save point
- 자동 장애 복구
- Row 레벨 잠금
- 데이터 일관성 보장

## 인-메모리 구성

- InnoDB Buffer Pool - LRU 알고리즘
- Change Buffer - Secondary Index
- Redo Log Buffer

## 온-디스크 구성

- 여러 테이블스페이스 파일
- 여러 로그 파일



# InnoDB Tablespaces

## Data Tablespaces

- System tablespace 파일 : ibdata1:12M; /ext/ibdata2:20M: autoextend
  - 각종 메타 데이터, 롤백 시그먼트 등 저장
  - innodb\_file\_per\_table=OFF인 경우 실제 데이터도 저장
- File-per-table tablespace: tablename.ibd 파일
  - 한 테이블의 데이터, 인덱스 및 메타데이터가 싱글 파일에 포함됨
  - innodb\_file\_per\_table=ON인 경우, 테이블 단위로 파일이 작성됨(디폴트 설정)
  - 압축 및 **공간 회수에 유리**
- General tablespace 파일 : 테이블명.ibd 파일
  - 여러 테이블이 하나의 테이블 스페이스 파일을 공유, **공간 재활용에 유리**

```
CREATE TABLESPACE tablespace_name
  [ADD DATAFILE 'file_name']
  [FILE_BLOCK_SIZE = value]
  [ENGINE [=] engine_name]
```

# InnoDB Tablespaces

**Undo Tablespaces** : innodb\_undo\_001, innodb\_undo\_002, .....

- 변경 전 데이터를 저장, 롤백 및 읽기 일관성 보장
- 독립된 tablespace 파일에 저장(MySQL 8.0)
- 자동 축소됨
  - 디폴트로 128 번 Purge thread 실행될 때마다 사이즈를 줄임

## Temporary tablespaces

- Session (temp\_1.ibt, temp\_2.ibt): 사용자 및 내부 프로세스
  - innodb\_temp\_data\_file\_path = ibtmp1:12M:autoextend:max:5G
- Global (ibtemp1): undo log의 롤백시그먼트 저장

```
mysql> show status like '%tmp%';
+-----+-----+
| Variable_name          | Value |
+-----+-----+
| Created_tmp_disk_tables | 1     |
| Created_tmp_files      | 7     |
| Created_tmp_tables     | 18    |
+-----+-----+
```

## InnoDB 트랜잭션 로그 파일

InnoDB Redo 로그: `ib_logfile0` 、 `ib_logfile1` 、 .....

- InnoDB 변경 작업을 기록하여 크래시 복구를 위한 로그 파일
- 충분히 크게 설정(사용율 70%)
  - `innodb_log_files_in_group`
  - `innodb_log_file_size`
  - `innodb_log_writer_threads`(MySQL 8.0.22)

InnoDB Undo 로그:

- 트랜잭션 커밋하기 전 데이터 저장 – rollback segments
- 두개의 디폴트 undo tablespace에 저장

# 바이너리 로그 파일

실행한 쿼리 혹은 데이터 변경 처리 내용을 기록하여 **MySQL Replication**에 사용

- 바이너리 포맷, **mysqlbinlog** 로 텍스트로 전환 가능
  - `SHOW BINARY LOGS-binlog.index, binlog.000001, binlog.000002`
  - Rotate/flush logs - `FLUSH BINARY LOGS`
- `binlog_format`으로 기록 포맷 설정
  - ROW
  - STATEMENT
  - MIXED
- 필터링 가능(--binlog-do-db/--binlog-ignore-db)
- 데이터 **Point-in-time Recovery** 에 사용
- MySQL 8.0에서 디폴트로 활성화, 시스템 변수 `log_bin` 으로 제어
- MySQL 8.0에서 디폴트 자동 삭제 기능 활성화(`binlog-expire-logs-seconds`)
  - 기존 사용하던 `expire_logs_days` 는 권장하지 않고 폐지될 예정
- 수동 삭제 가능: `PURGE BINARY LOGS TO/BEFORE '파일명'`



# InnoDB Work Flow

Crash safe를 보장할 수 있는 옵션들:

`innodb_flush_log_at_trx_commit = 1`

`sync_binlog = 1`

`gtid_mode = on`

`enforce_gtid_consistency = 1`

`binlog_gtid_simple_recovery = 1`

`relay_log_recovery = ON`

`master_info_repository = TABLE`

`relay_log_info_repository = TABLE`

추가 복제 관련 옵션:

`plugin-`

`load="rpl_semi_sync_master=semisync_master.`

`so;rpl_semi_sync_slave=semisync_slave.so"`

`rpl-semi-sync-master-enabled = 1`

`rpl-semi-sync-slave-enabled = 1`

`rpl_semi_sync_master_wait_no_slave = 1`

2. Write old data

1. Load data page

3. Update page & Lock

4.1 Write redo log

4.2 Prepare to Commit & Flush redo log

5.1 Prepare to Commit & Write binlog

Bin log cache

Executor

InnoDB Storage Engine

InnoDB Buffer Pool

Redo Log Buffer

7. Page cleaner thread

5.2 Sync binlog

Undo Log

Data File

Redo Log

Bin log

6. Write binlog info & ACK & Commit



## 기타 로그 파일

---

### 에러 로그

- 시작/중지 및 서버측 오류 혹은 경고 메시지에 관련된 로그 파일
- 디폴트 출력 위치는 플랫폼에 따라 다름
  - log\_error
  - log\_error\_verbosity
- JSON 포맷 지원(8.0버전)
  - <https://dev.mysql.com/doc/refman/8.0/en/error-log-json.html>
- 테이블에 기록(MySQL 8.0.22)
  - SELECT \* FROM performance\_schema.error\_log\G

## 기타 로그 파일

---

### 일반 쿼리 로그

- 클라이언트로 부터의 연결 및 실행한 전체 SQL문을 기록
- log\_output /general\_log 로 위치 설정

### Slow 쿼리 로그

- 지정된 시간 이상으로 실행된 쿼리를 기록
- log\_output /slow\_query\_log로 위치 설정
- **long\_query\_time** : 초단위로 지정(0.5로 지정하는 경우 500ms)
- log\_queries\_not\_using\_indexes : 인덱스를 사용하지 않은 쿼리를 기록

### Enterprise Audit Log

- EE에서 지원, 데이터베이스에 대한 액티비티를 기록
- 정책 기반 필터링 및 암호화 지원

# MySQL 설치 및 시작



# 설치 방법(Linux/yum/rpm)

공식 레파지토리를 설치하고 yum 명령문으로 자동으로 최신 GA버전을 설치(네트워크 연결이 필요)

1. 공식 레파지토리의 설치: <http://dev.mysql.com/downloads/repo/>

```
shell > wget https://dev.mysql.com/get/mysql80-community-release-el7-3.noarch.rpm
shell > sudo yum install -y mysql80-community-release-el7-3.noarch.rpm
```

## 2. MySQL 설치

```
Shell > sudo yum install mysql
shell > sudo yum install mysql-server
shell > sudo yum install mysql-shell
```

## 3. MySQL 시작/중지(처음으로 시작시 DB도 생성됨)

```
shell> sudo service mysqld start
shell> sudo service mysqld stop
※EL7의 경우
shell> sudo systemctl start mysqld
shell> sudo systemctl stop mysqld
```

# Install from tar ball

패키지 다운로드:

사용자 추가: `sudo groupadd mysql`

그룹 추가: `sudo useradd -r -g mysql -s /bin/false mysql`

디렉토리 만들기: `sudo mkdir /your directory/mysql-8.0`

압축 풀기: `sudo tar xvf /your directory/mysql-VERSION-OS.tar.xz`(위 디렉토리에서)

심볼릭 링크 (옵션사항) 만들기 : `sudo ln -s your-install-directory mysql`

설정 파일 작성 : `sudo vi /your directory/my.cnf`

파일 권한 변경 : `sudo chown mysql:mysql /your directory/mysql-8.0`

환경 변수 추가: `sudo export PATH=$PATH:/ mysql's bin dir`

데이터베이스 초기화: `sudo /mysql's bin dir/mysqld --defaults-file=/your directory/my.cnf --initialize`

데이터베이스 시작:

`sudo /mysql's bin dir/mysqld --defaults-file=/your directory/my.cnf &`

`sudo /mysql's bin dir/mysqld_safe --defaults-file=/your directory/my.cnf &`

비밀번호 확인: `cat /your error log dir/<error log>`

데이터베이스 접속:

`mysql --defaults-file=/your directory/my.cnf -uroot -h127.0.0.1 -P3306 -p`

`mysqlsh root@localhost:3306`

EE Trial : <https://edelivery.oracle.com/osdc/faces/Home.jsp>

Community: <https://dev.mysql.com/downloads/mysql/>

# MySQL 관리 및 모니터링



# MySQL 서버 상태

서버 상태 변수:

- `SHOW GLOBAL|SESSION STATUS`

```
mysql> SHOW GLOBAL STATUS;
+-----+-----+
| Variable_name | Value |
+-----+-----+
| Aborted_clients | 0 |
| Aborted_connects | 0 |
| Binlog_cache_disk_use | 0 |
| Bytes_received | 169 |
| Bytes_sent | 331 |
| Com_admin_commands | 0 |
| ...
```

- `Innodb_buffer_pool_wait_free`
- `Innodb_log_waits`
- `threads_%(connected|created|running|cached)`
- `connections`
- `max_used_connections`
- `created_tmp_disk_tables`
- `handler_read_first`
- `open_table_%`
- `opened_table_%`
- `select_full_join`
- `slow_queries`
- `com_xxx_%(commit|rollback)`
- `uptime`



# MySQL Metadata 정보

**INFORMATION\_SCHEMA** 를 통해 MySQL서버의 metadata 정보 (view) 확인

```
$ USE INFORMATION_SCHEMA;
```

```
$ SHOW TABLES;
```

```
$ SELECT TABLE_NAME, FROM INFORMATION_SCHEMA.TABLES
```

```
WHERE TABLE_SCHEMA = 'information_schema'
```

```
ORDER BY TABLE_NAME\G
```

```
$ DESCRIBE(DESC) TABLES;
```

```
$ SHOW CREATE TABLES;
```

```
$ SHOW CHARACTER SET
```

```
$ SHOW COLLATION;
```

# MySQL 성능 지표

PERFORMANCE\_SCHEMA를 통해 MySQL서버의 성능 지표 수집, 메모리에 저장

- \$ USE PERFORMANCE\_SCHEMA;
- \$ SHOW TABLES;

동적으로 설정 가능

- setup\_actors
- setup\_consumers
- setup\_instruments
- setup\_objects
- setup\_threads

```
mysql> UPDATE performance_schema.setup_instruments
      SET ENABLED = 'YES', TIMED = 'YES';
Query OK, 560 rows affected (0.04 sec)
mysql> UPDATE performance_schema.setup_consumers
      SET ENABLED = 'YES';
Query OK, 10 rows affected (0.00 sec)
```

# Table Maintenance

## SQL 명령문

- `ANALYZE TABLE t UPDATE HISTOGRAM ON c1, c2, c3 WITH 10 BUCKETS ; (8.0)`
- `CHECK TABLE`: 테이블 및 뷰 오류 체크
- `REPAIR TABLE`: MYISAM, ARCHIVE, CSV, **InnoDB 제외**
- `OPTIMIZE TABLE t ;` 테이블 데이터 및 인덱스 공간 정리, 통계 갱신, **InnoDB 제외**

## InnoDB table

- 자동 장애 복구
- `innochecksum`: 오프라인 체크섬 툴
- `mysqldump + mysqlbinlog` 로 복구
- `innodb force recovery = 1` 옵션으로 재시작
- `ALTER TABLE t FORCE;` 공간 정리 및 통계 갱신

# MySQL 보안 기능

## 연결 암호화 지원

- TLS 프로토콜
- SSL/RSA 인증

## 패스워드 인증 관리

- 강화된 인증 알고리즘
  - caching\_sha2\_password (MySQL 8.0)
- 강화된 암호 정책
  - [mysql\\_secure\\_installation](#)

## 데이터 암호화

- <https://dev.mysql.com/doc/refman/8.0/en/encryption-functions.html>

## Assess - 평가

- MySQL Enterprise Monitor

## Prevent - 방어

- MySQL Enterprise Authentication
- MySQL Enterprise Firewall
- MySQL Enterprise Encryption
- MySQL Enterprise Data Masking

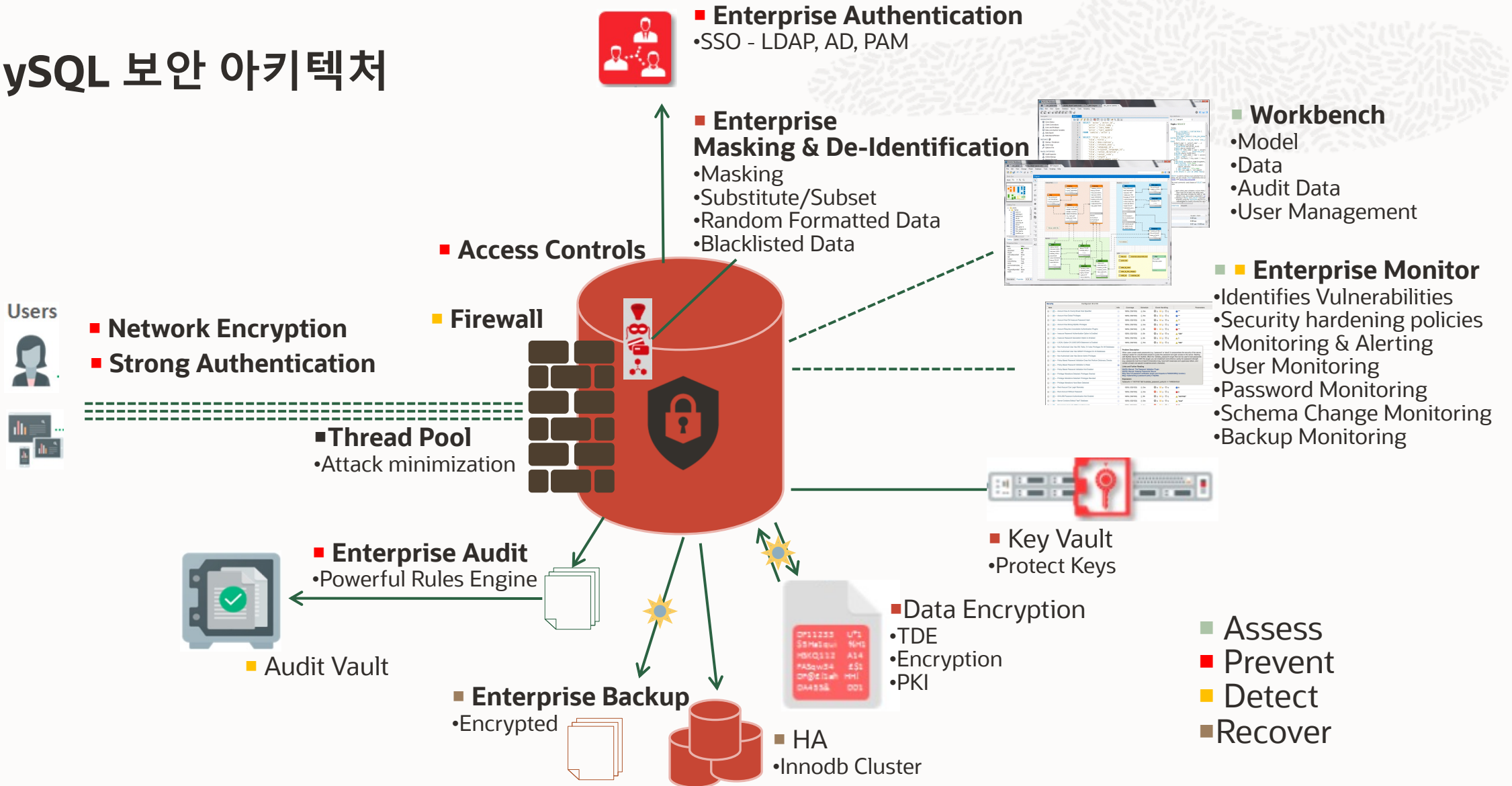
## Detect - 감지

- MySQL Enterprise Audit
- MySQL Enterprise Firewall

## Recover - 복구

- MySQL Enterprise HA
- MySQL Enterprise Backup

# MySQL 보안 아키텍처



# MySQL 상태 모니터링 - Who, What, How?

Who ?

- show processlist;
- kill <thread id>;

What configuration do I have ?

- show variables like '%옵션명%';

What status do I have ?

- mysql 상태: show status like '%변수명%';
- 시스템 상태: top|free|vmstat|mpstat|netstat|iostat
- 프로시저: `sys.diagnostics()`

```
mysql> tee diag.out;
mysql> CALL sys.diagnostics(120, 30, 'current');
mysql> notee;
```

How does Server/InnoDB/Performance\_Schema process the jobs ?

- [EXPLAIN ANALYZE\(MySQL 8.0.20\)](#)
- SHOW engine INNODB status
- SHOW engine INNODB mutex
- SHOW engine PERFORMANCE\_SCHEMA status\G

# MySQL 성능 모니터링

## Performance\_Schema

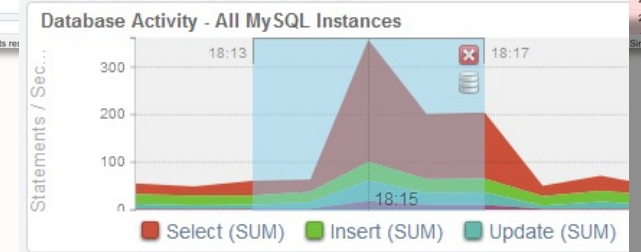
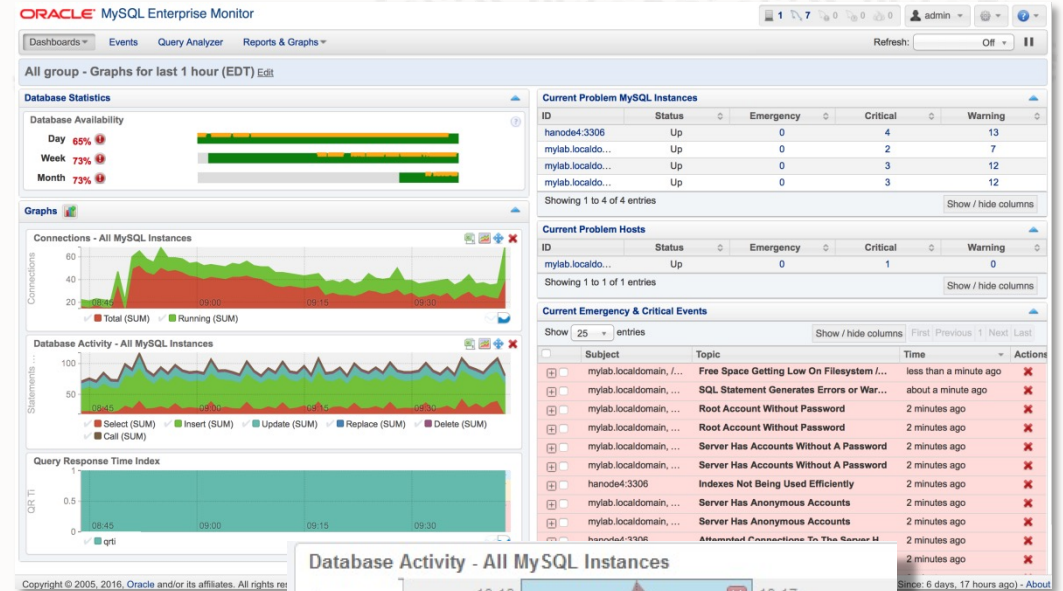
- Error
- statements
- transactions
- waits
- file
- memory
- Replication
- table\_io
- table\_lock

MySQL Enterprise Monitor

Oracle Enterprise Monitor

MySQL Workbench

Error log, Slow query log, General log



|   | Current | Worst | Subject                                   | Topic   |
|---|---------|-------|---|---|
| + | !       | !     | mylab.localdomain, mylab.localdomain:3306 | Root Account Without Password                 |
| + | !       | !     | mylab.localdomain, mylab.localdomain:3306 | Server Has Accounts Without A Password        |
| + | ✓       | !     | mylab.localdomain, mylab.localdomain:3306 | Average Statement Execution Time Excess...    |
| + | ✓       | !     | mylab.localdomain, mylab.localdomain:3306 | SQL Statement Generates Errors or Warnings    |
| + | !       | !     | mylab.localdomain, mylab.localdomain:3306 | Server Has Anonymous Accounts                 |
| + | ✓       | !     | mylab.localdomain, mylab.localdomain:3306 | MySQL Instance Is Experiencing A Query P...   |
| + | !       | !     | mylab.localdomain, mylab.localdomain:3306 | InnoDB Log Buffer Flushed To Disk After Ea... |
| + | !       | !     | mylab.localdomain, mylab.localdomain:3306 | User Has Rights To Database That Does Not...  |



# MySQL Enterprise Monitor

## MySQL보안 위기 및 방지

- 보안 취약점 발견 및 해결
- Firewall 및 Audit 모니터링
- 유저 액세스 및 권한 모니터링

## 시스템 성능 관리 및 개선

- 시각적인 쿼리 분석기 및 쿼리 성능 모니터링
- 데이터베이스 가용성 모니터링
- CPU 사용량, RAM 사용량, swap 사용량

## 복제 토폴로지 및 성능

- Replication
- InnoDB Cluster
- MySQL Cluster

## 개발자 최적화 가이드 - Advisor

- 임계 값 기반 경고 및 알람 발송

| 성능 문제 발생 원인        |
|--------------------|
| 테이블 스캔을 수행하는 Query |
| 디스크 임시 테이블 과다      |
| CPU 스파이크           |
| 디스크 I/O 포화         |
| 내부 잠금              |
| 하드웨어 문제            |
| 데이터베이스 및 스키마 변경사항  |
| 새로운 Query 도입       |
| MySQL 설정 및 구성 문제   |



# MySQL Best Practice



## 초기 설정 파일: my.cnf

### 서버 디폴트 설정 권장

파일 경로 설정: 로그 파일, IO분산 목적, SSD 활용  
추가 모니터링 옵션

- [innodb monitor enable](#)
- SHOW VARIABLES LIKE 'perf%';

### 용량 설정

- [innodb buffer pool size](#)(70~80% of memory)
- [innodb log file size](#)(70% usage )
- [max connections](#): 151
- [table definition cache](#) > 테이블 개수
- [table open cache](#) : 4000
- [table open cache instances](#) = 16

### 여러 버퍼 값은 디폴트로 시작, 세션별 동적 설정 권장

- [join buffer size](#)
- [sort buffer size](#)
- [tmp table size](#)

### 데이터 일관성

- [innodb flush log at trx commit](#) = 1
- [sync\\_binlog](#) = 1

### 확인 및 변경 방법:

- SHOW GLOBAL/SESSION VARIABLES LIKE '%변수명%';
- SELECT @@global/session.변수명;
- SET @@sql\_mode = 'TRADITIONAL';
- SET **PERSIST** max\_connections = 1000;(MySQL 8.0)

# MySQL Databases 설계

데이터 타입을 선택할 때 **공간, 효율성, 운영** 문제 고려

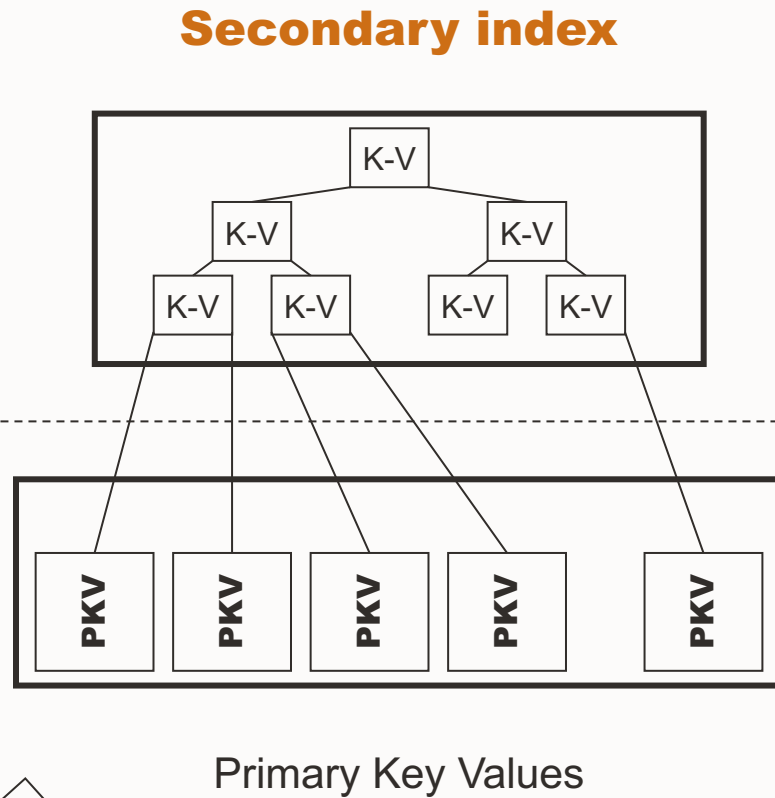
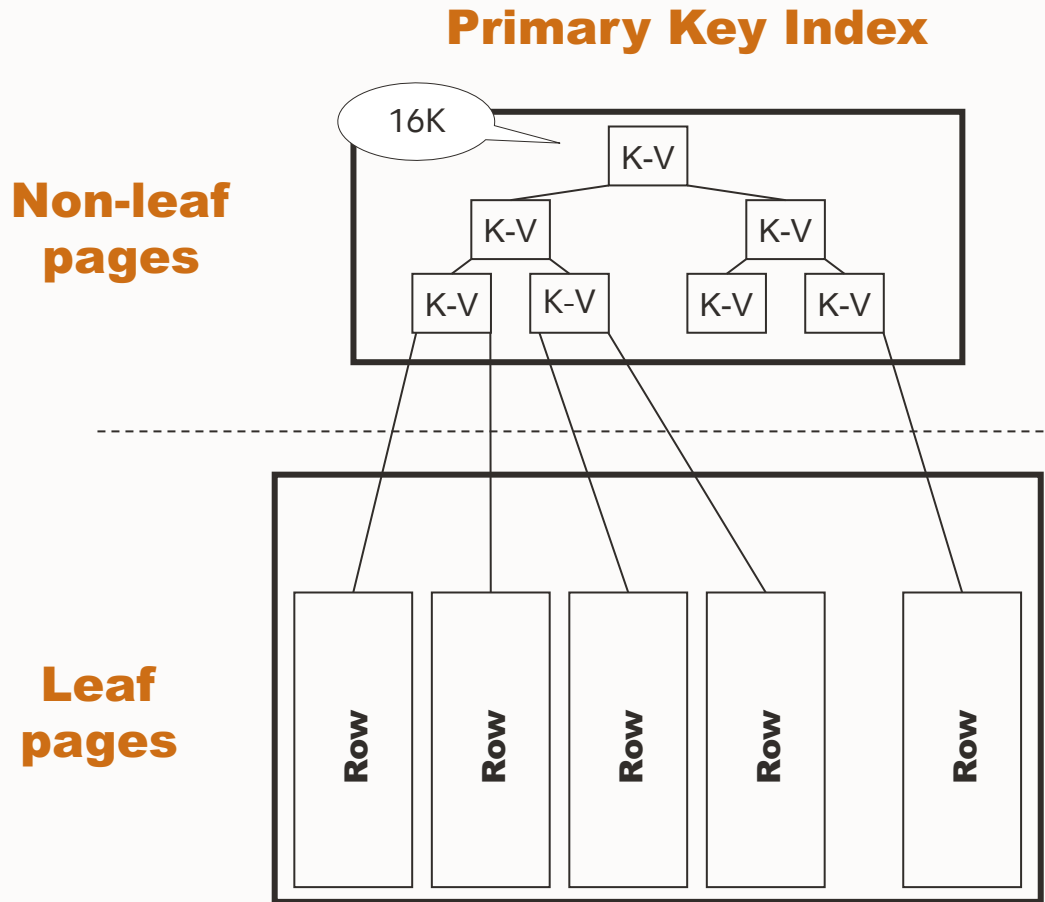
- PK 를 AUTO\_INCREMENT 로 정의할 경우 INT 혹은 BIGINT ?
- Unsigned 대신에 Signed – 매출 건수
- DECIMAL(n, m) 대신에 INT – 금액
- [INET\\_NTOA\(\)](#), [INET\\_ATON\(\)](#) – IP 주소
- String타입에 대해서 UTF8MB4 권장
  - ALTER TABLE t **CONVERT TO** CHARSET **utf8mb4**;
- 상태 값은 CHAR(1) + CHECK (**8.0.16**)
  - CONSTRAINT 'status\_chk' CHECK (status in ('S','T','D','U'))
- JSON 타입 (**8.0.17**) – 사용자 등록 테이블, 사용자 태그
  - JSON 함수 지원
  - Functional 인덱스 지원

## 인덱스의 설계

- 조회 성능을 향상하기 위한 데이터 구조

```
CREATE TABLE UserLogin (  
  userId BIGINT NOT NULL,  
  loginInfo JSON,  
  PRIMARY KEY(userId));
```

# InnoDB Clustered Index



# MySQL 인덱스 설계

## 인덱스는 많으면 많을 수록 좋을까요?

- 조회에 꼭 필요한 인덱스만 만들고, 사용하지 않은 인덱스는 삭제할 것을 권장
  - Select \* from `sys.schema_unused_indexes` where object\_schema != 'performance\_schema';
  - Alter index idx\_name `invisible/visible`;(MySQL8.0)
- 복합 인덱스 활용할 것을 권장 (a,b)
  - 인덱스 개수를 줄임
  - Using filesort를 피할 수 있음: select \* from table where a = ? order by b desc;
  - Covering Index로 Back Table을 피할 수 있음 : (Extra: Using Index)

## 인덱스 컬럼 사이즈는 영향이 있을까요?

- 공간 절약, 인덱스 스캔 효율성 향상
  - 16byte 혹은 이하로 권장, 32byte이상 피해야 함
  - Prefix 인덱스 활용
  - MD5로 해시키 생성

## MySQL 인덱스 설계 - 예시 1

```
CREATE TABLE User (  
  id BIGINT AUTO_INCREMENT,  
  name VARCHAR(128) NOT NULL,  
  money BIG INT NOT NULL DEFAULT 0,  
  register_date DATETIME(6) NOT NULL DEFAULT  
  CURRENT_TIMESTAMP(6),  
  
  last_modify_date DATETIME(6) NOT NULL DEFAULT  
  CURRENT_TIMESTAMP(6)  
  ON UPDATE CURRENT_TIMESTAMP(6),  
  CHECK (sex = 'M' OR sex = 'F'),  
  PRIMARY KEY(id), -- 순서 입력  
  UNIQUE KEY idx_name(name), -- SI, 횟수 제한  
  KEY idx_register_date(register_date), -- SI, 변경 없음  
  KEY idx_last_modify_date(last_modify_date)-- SI  
);
```

2021년 1월 등록 사용자 조회 쿼리

```
SELECT * FROM User  
WHERE DATE_FORMAT(register_date,'%Y-%m') =  
'2021-01'
```

```
SELECT * FROM User  
WHERE register_date > '2021-01-01'  
AND register_date < '2021-02-01'
```

성능 문제 임시 해결 할 수 있으나, 인덱스 중복

```
ALTER TABLE User ADD INDEX  
idx_func_register_date((DATE_FORMAT(register_d  
ate,'%Y-%m')));
```

## MySQL 인덱스 설계 - 예시 2

```
CREATE TABLE `orders` (  
  `O_ORDERKEY` int NOT NULL,  
  `O_CUSTKEY` int NOT NULL,  
  `O_ORDERSTATUS` char(1) NOT NULL,  
  `O_TOTALPRICE` decimal(15,2) NOT NULL,  
  `O_ORDERDATE` date NOT NULL,  
  `O_ORDERPRIORITY` char(15) NOT NULL,  
  `O_CLERK` char(15) NOT NULL,  
  `O_SHIPPRIORITY` int NOT NULL,  
  `O_COMMENT` varchar(79) NOT NULL,  
  PRIMARY KEY (`O_ORDERKEY`), --PK  
  KEY `ORDERS_FK1` (`O_CUSTKEY`), --SI  
  CONSTRAINT `orders_ibfk_1` FOREIGN KEY  
  (`O_CUSTKEY`) REFERENCES `customer`  
  (`C_CUSTKEY`)  
) ENGINE=InnoDB DEFAULT
```

사용자 소비 이력 조회:

```
SELECT * FROM orders  
WHERE o_custkey = 147601 ORDER BY o_orderdate  
DESC
```

복합 인덱스 추가:

```
ALTER TABLE orders ADD INDEX  
idx_custkey_orderdate(o_custkey, o_orderdate);
```

```
SELECT o_custkey,o_orderdate,o_totalprice  
FROM orders WHERE o_custkey = 147601\G
```

복합 인덱스 추가:

```
ALTER TABLE `orders` ADD INDEX  
idx_custkey_orderdate_totalprice(o_custkey,o_orderda  
te,o_totalprice);
```

사용자 별 소비 총 가격 조회, VIP고객 선별 등 용도

```
SELECT o_custkey, SUM(o_totalprice)  
FROM orders GROUP BY o_custkey;
```

```
ALTER TABLE orders
ALTER INDEX idx_custkey_orderdate_totalprice INVISIBLE;
EXPLAIN SELECT o_custkey,SUM(o_totalprice)
FROM orders GROUP BY o_custkey\G
```

```
***** 1. row *****
```

```
id: 1
select_type: SIMPLE
table: orders
partitions: NULL
type: index
possible_keys:
idx_custkey_orderdate,ORDERS_FK1
```

**key: ORDERS\_FK1**

```
key_len: 4
ref: NULL
rows: 5778755
filtered: 100.00
Extra: NULL
```

```
EXPLAIN FORMAT=tree
SELECT o_custkey,SUM(o_totalprice)
FROM orders GROUP BY o_custkey\G
```

```
***** 1. row *****
```

```
EXPLAIN: -> Group aggregate: sum(orders.O_TOTALPRICE)
-> Index scan on orders using
ORDERS_FK1 (cost=590131.50 rows=5778755)
```

12.35s

```
ALTER TABLE orders
ALTER INDEX idx_custkey_orderdate_totalprice VISIBLE;
EXPLAIN SELECT o_custkey,SUM(o_totalprice)
FROM orders GROUP BY o_custkey\G
```

```
***** 1. row *****
```

```
id: 1
select_type: SIMPLE
table: orders
partitions: NULL
type: index
possible_keys:
idx_custkey_orderdate,ORDERS_FK1,idx_custkey_orderd
```

```
ate_totalprice
```

**key: idx\_custkey\_orderdate\_totalprice**

```
key_len: 14
ref: NULL
rows: 5778755
filtered: 100.00
```

**Extra: Using index**

```
1 row in set, 1 warning (0.00 sec)
```

1.04s





감사합니다.

Our mission is to help people see  
data in new ways, discover insights,  
unlock endless possibilities.



ORACLE