

MySQL Database

Introduction

Introduction

데이터베이스와 데이터베이스 관리 시스템

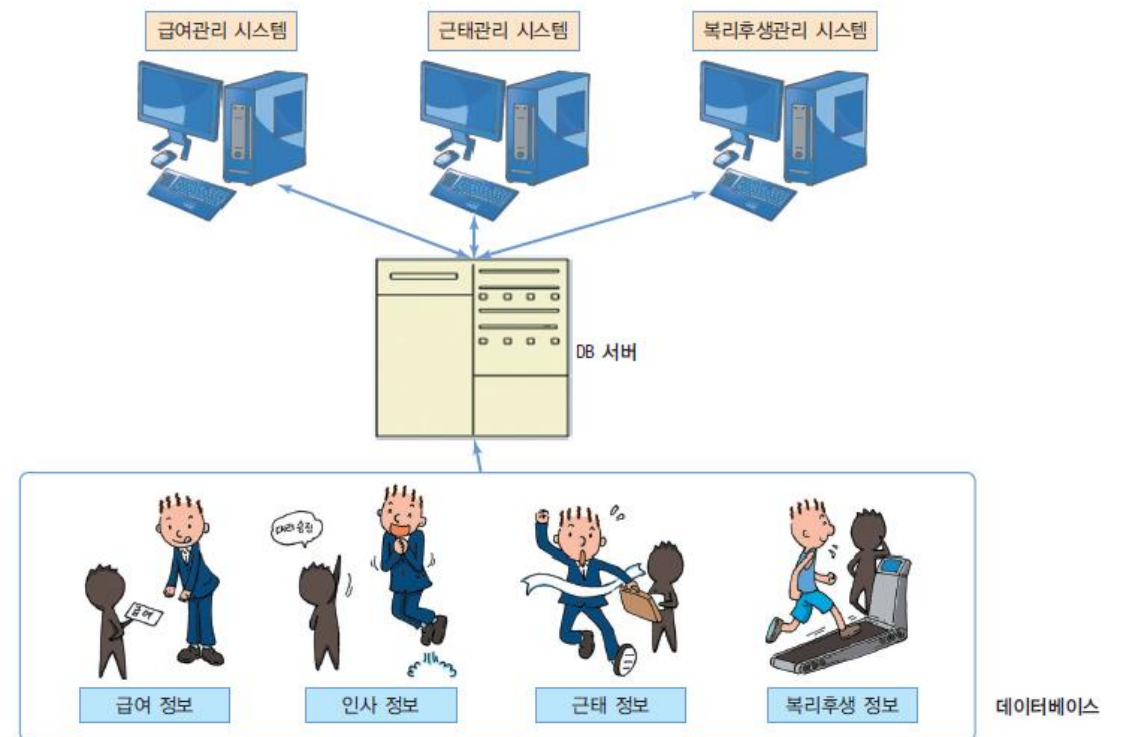
데이터베이스와 데이터베이스 관리 시스템

▶ Data

- ▶ 하나하나의 단편적인 정보
- ▶ 정보는 있으나 아직 체계화가 되지 않은 상태

▶ Database

- ▶ 데이터의 집합 (a Set of Data)
- ▶ 여러 응용 시스템(프로그램)들의 통합된 정보들을 저장하여 운영할 수 있는 공용(Shared) 데이터의 집합
- ▶ 효율적으로 저장, 검색, 갱신할 수 있도록 데이터 집합들끼리 연관시키고 조직화되어야 한다



데이터베이스와 데이터베이스 관리 시스템

▶ 데이터베이스의 특성

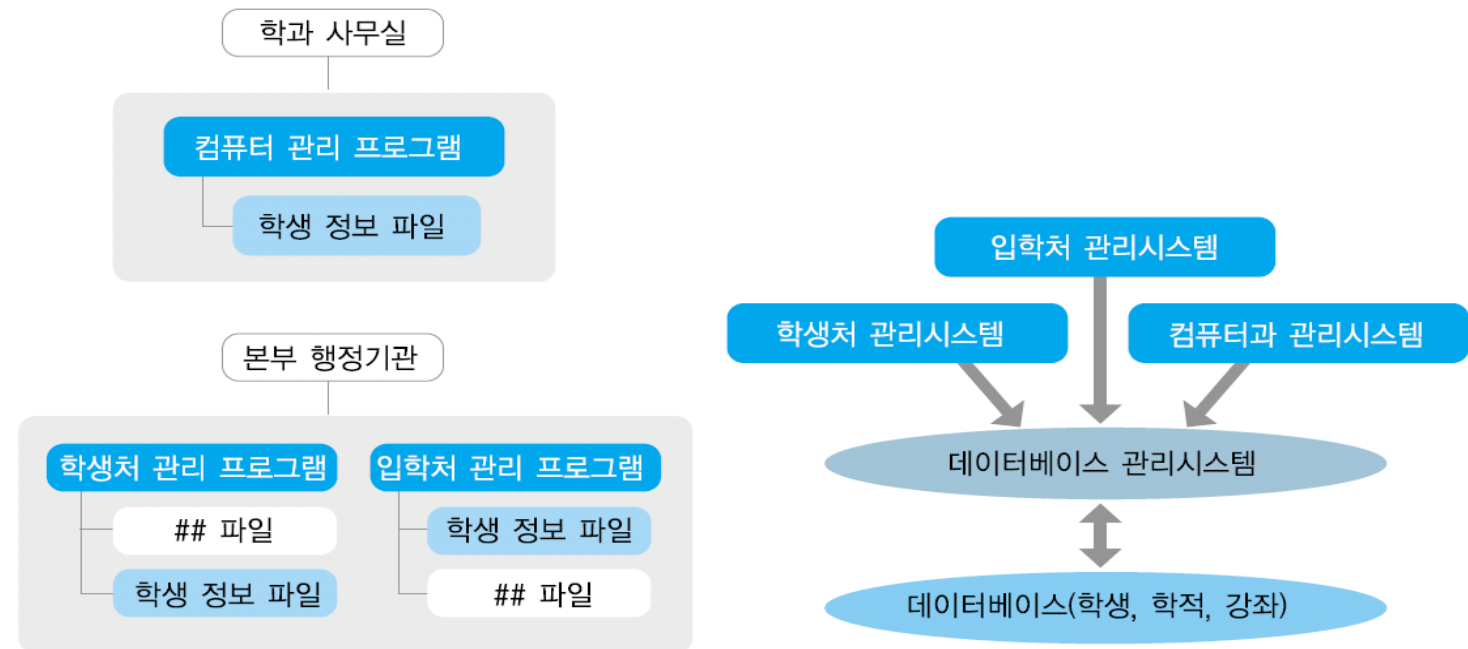
- ▶ 실시간 접근성 (**Real-time Accessibility**)
사용자의 요구를 즉시 처리할 수 있다
- ▶ 계속적인 변화 (**Continuous Evolution**)
정확한 값을 유지하기 위해 삽입, 삭제, 수정 작업 등을 이용하여 데이터를 지속적으로 갱신할 수 있다
- ▶ 동시 공유성 (**Concurrent Sharing**)
사용자마다 서로 다른 목적으로 사용하므로 동시에 여러 사람이 동일한 데이터에 접근하고 이용할 수 있다
- ▶ 내용 참조 (**Contents Reference**)
저장한 데이터 레코드의 위치나 주소가 아닌 사용자가 요구하는 데이터의 내용, 즉 데이터 값에 따라 참조할 수 있어야 한다

데이터베이스와 데이터베이스 관리 시스템

- ▶ 데이터베이스 관리 시스템 (Database Management System = DBMS)
 - ▶ 데이터베이스를 관리, 운영하는 소프트웨어
- ▶ 여러 응용 소프트웨어(프로그램) 또는 시스템이 동시에 데이터베이스에 접근하여 사용할 수 있게 한다
- ▶ 필수 3기능
 - ▶ 정의 기능 : 데이터베이스의 논리적, 물리적 구조를 정의
 - ▶ 조작 기능 : 데이터를 검색, 삽입, 갱신, 삭제하는 기능
 - ▶ 제어 기능 : 데이터베이스의 내용 정확성과 안전성을 유지하도록 제어하는 기능
- ▶ Oracle, Microsoft SQL Server, MySQL, DB2 등의 상용 또는 공개 DBMS가 있다

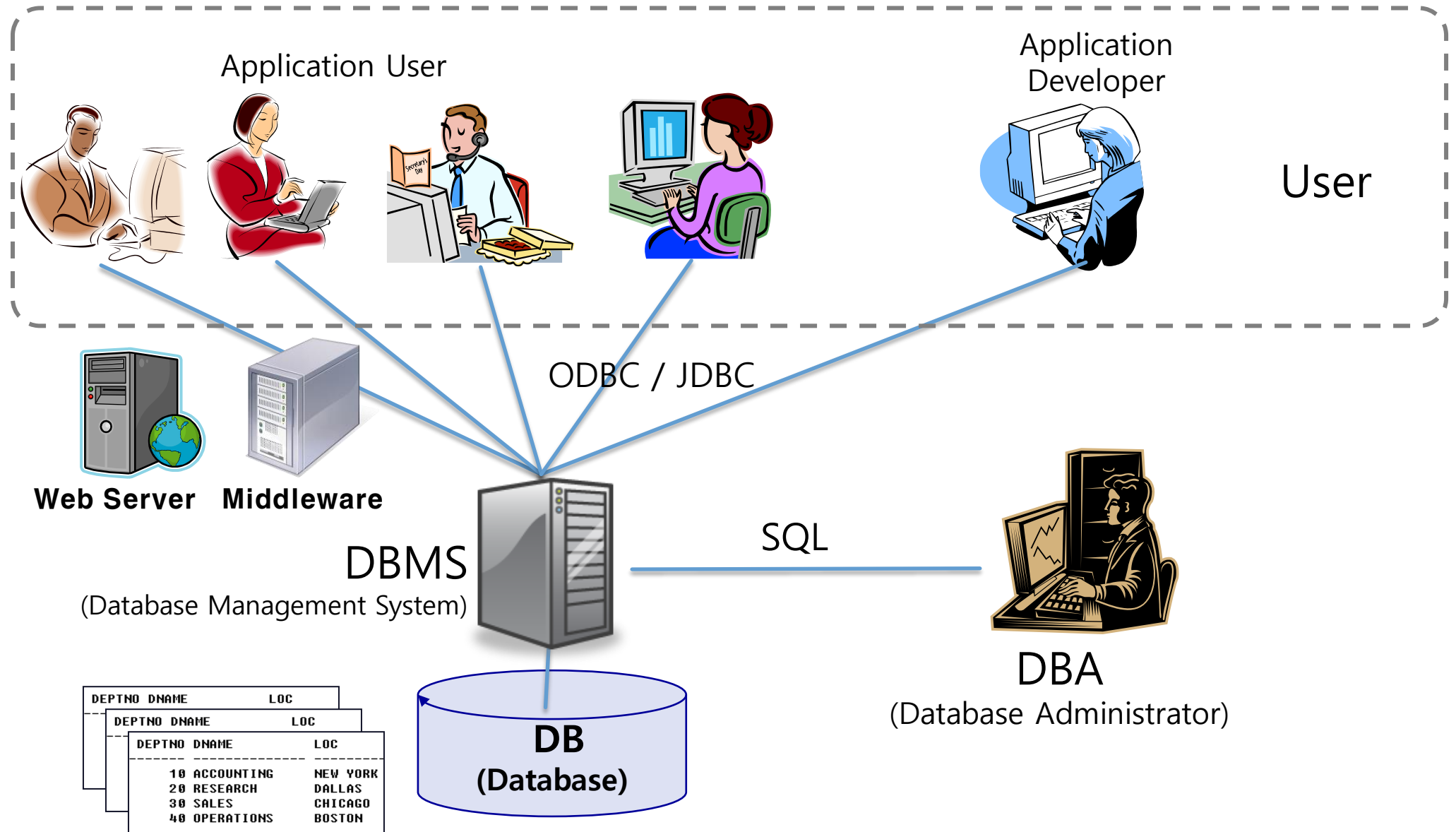
데이터베이스와 데이터베이스 관리 시스템

- ▶ 기존 파일 시스템의 문제
 - ▶ 데이터 종속성으로 인한 문제
 - ▶ 데이터 중복성으로 인한 문제
- ▶ 데이터베이스의 도입
 - ▶ 데이터의 종속성 보완
 - ▶ 중복성 제거



[그림 7-3] 파일관리시스템(좌)과 데이터베이스 관리시스템(우)

데이터베이스와 데이터베이스 관리 시스템



Database Schema

▶ Schema

- ▶ 데이터베이스의 논리적 정의

▶ 3단계 Schema

▶ External Schema (외부 스키마)

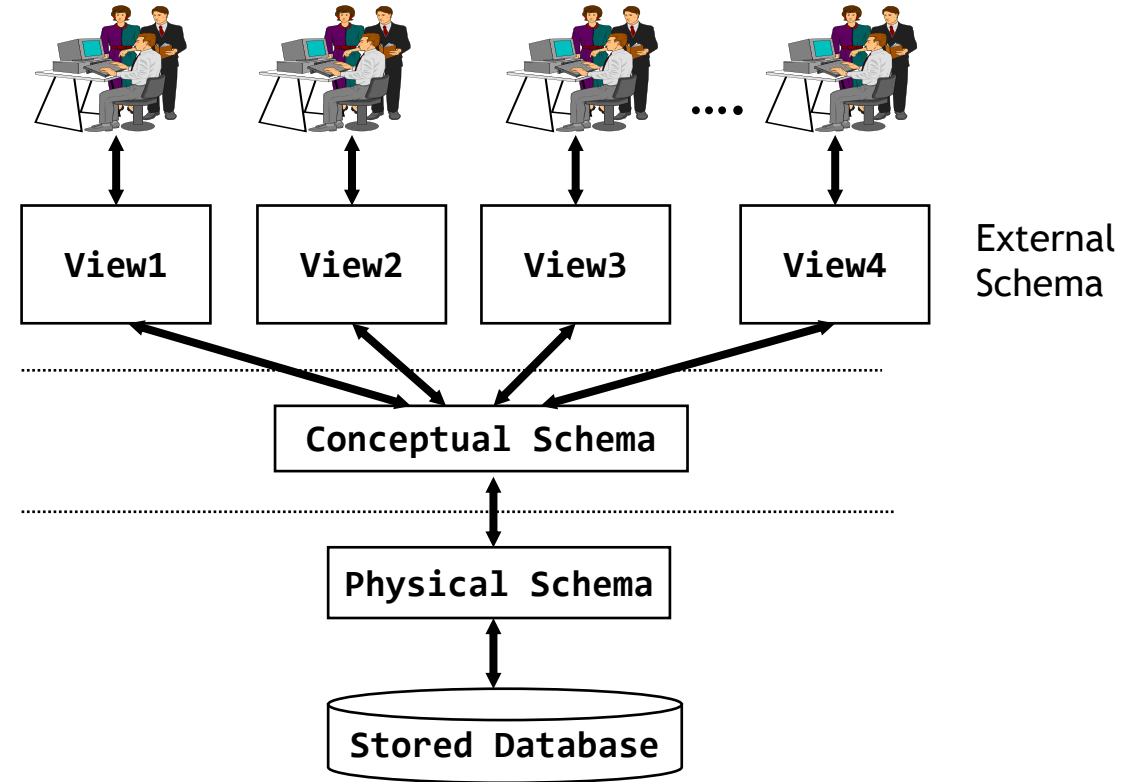
- ▶ 각 사용자 입장에서 본 Database 구조
- ▶ 사용자마다 서로 다른 Schema를 가짐
- ▶ 개념 Schema에 대한 서브 Schema

▶ Conceptual Schema (개념적 스키마)

- ▶ 조직 전체의 입장에서 본 Database 구조
- ▶ 한 개의 Schema만 존재하며, 서로 다른 사용자가 공유
- ▶ Data 객체(개체, 관계), 제약 조건에 대한 명세를 유지

▶ Physical Schema (물리적 스키마)

- ▶ 저장 장치의 입장에서 본 Database 구조
- ▶ 각 data 객체의 저장 구조를 표현
- ▶ 내부 레코드의 형식
- ▶ 저장 data 항목의 표현 방법



데이터베이스와 데이터베이스 관리 시스템

▶ 데이터베이스 관리 시스템의 장점

- ▶ 데이터 독립성 및 중복이 최소화
- ▶ 데이터의 일관성 및 무결성 유지
- ▶ 데이터 보안 보장
- ▶ 표준화되고 일관된 데이터 관리 기능
- ▶ 응용프로그램 개발 시간의 단축
- ▶ 데이터 동시 사용 가능
- ▶ 데이터 회복 가능

▶ 데이터베이스 관리 시스템의 단점

- ▶ 시스템 자원 요구로 운영비 증대
- ▶ 고급 프로그래밍 필요로 자료 처리의 복잡화
- ▶ 부분적 데이터베이스 손실이 전체 시스템을 정지
- ▶ 장애 발생 대비를 위한 복잡한 **Back Up**과 **Recovery** 작업 필요

데이터베이스와 데이터베이스 관리 시스템

▶ 데이터베이스의 종류

▶ 관계형 데이터베이스 (Relational Database = RDB)

- ▶ 1970년 IBM E. F. Codd에 의해 제안되어 수십년 동안 주류 데이터베이스로 성장 확대
- ▶ 키와 값들의 간단한 관계를 테이블화 시킨 매우 간단한 원칙의 개념을 가진 데이터베이스
- ▶ 일련의 정형화된 테이블로 구성된 데이터 항목들의 집합이며 각 테이블은 데이터의 성격에 따라 여러 개의 컬럼(키)이 포함된다
- ▶ 사용자는 **SQL**이라는 표준 질의어를 통해 데이터를 조작 또는 조회할 수 있다
 - ▶ SQL (Structured Query Language)

▶ 객체지향 데이터베이스 (Object Oriented Database = OODB)

- ▶ 정보를 객체의 형태로 표현하는 데이터베이스
- ▶ 객체 모델이 그대로 데이터베이스에도 적용되어 데이터 모델을 그대로 응용프로그램에 적용, 데이터 변환과 질의 작업이 필요하지 않은 장점이 있다

데이터베이스와 데이터베이스 관리 시스템

▶ 데이터베이스의 종류

▶ 객체관계형 데이터베이스 (Object Relation Database = ORDB)

- ▶ 관계형 데이터베이스에서 사용하는 데이터를 확장
- ▶ 관계형 데이터베이스를 객체 지향 모델링과 데이터를 관리하는 기능을 갖도록 확장한 것

▶ NoSQL

- ▶ 대용량 데이터, 비정형 데이터의 웹 서비스와 SNS, 클라우드 컴퓨팅의 확대 보급과 대중화로 최근 주목받고 있는 데이터베이스 기술

▶ 그 외

- ▶ Hierarchical Database
- ▶ Network Database

관계형 데이터베이스

: SQL(Structured Query Language)

- ▶ 데이터베이스 스키마 생성, 자료의 검색, 수정, 그리고 데이터베이스 객체 접근 관리 등을 위해 고안된 언어
 - ▶ 비절차식 언어
 - ▶ 1970년대 IBM의 SYSTEM R 프로젝트를 통해 개발
- ▶ 다수의 데이터베이스 관련 프로그램의 표준 언어
 - ▶ RDBMS에서 사용하기 위해 ANSI에서 책정한 표준 언어
 - ▶ DBMS 제품별로 SQL에 대한 추가 및 확장

관계형 데이터베이스

: SQL(Structured Query Language)

▶ DML(Data Manipulation Language)

- ▶ 데이터 처리를 위해 응용프로그램과 데이터베이스 관리 시스템간의 인터페이스를 위한 언어
- ▶ 데이터 처리를 위한 연산의 집합으로 데이터의 검색, 삽입, 수정 및 삭제하기 위한 수단을 제공
- ▶ SELECT, INSERT, UPDATE, DELETE, MERGE

▶ DDL(Data Definition Language)

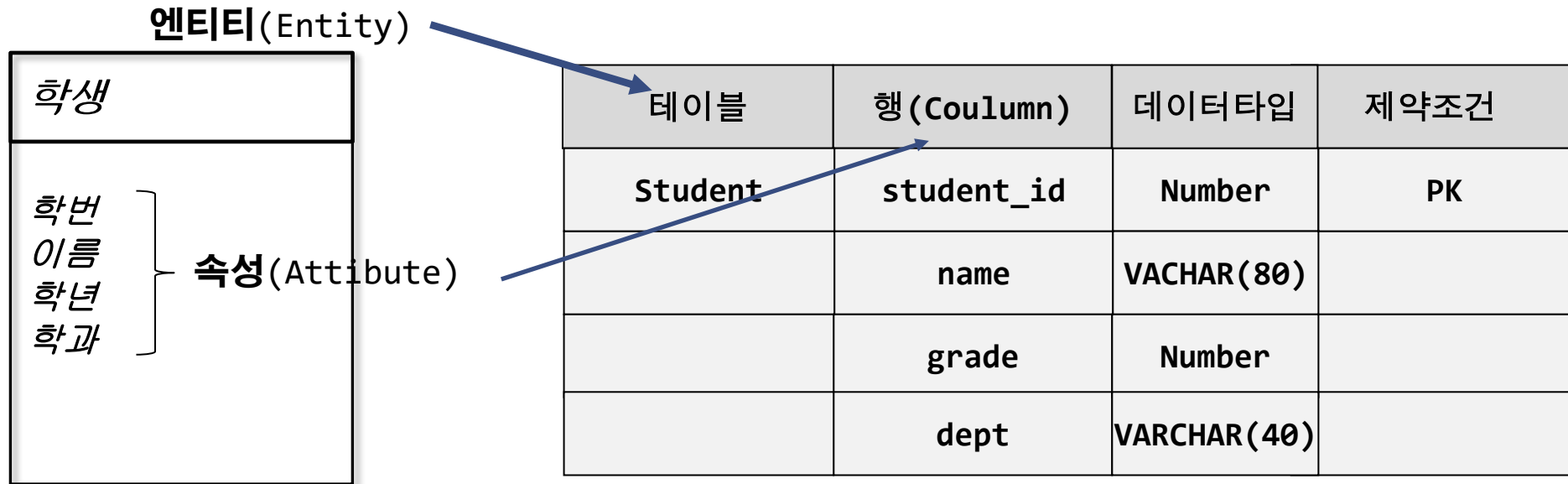
- ▶ 데이터베이스 구조, 데이터 형식, 접근 방식 등 데이터베이스를 구축, 변경할 목적으로 사용하는 언어
- ▶ DDL 컴파일러가 컴파일한 후 데이터 사전에 저장
- ▶ 데이터의 논리적, 물리적 구조를 생성, 변경, 삭제하는 기능을 제공
- ▶ CREATE, ALTER, DROP, RENAME

▶ DCL(Data Control Language)

- ▶ 보안 및 권한 제어, 무결성, 회복, 병행 제어를 위한 언어
- ▶ : 데이터에 대한 권한 관리 및 트랜잭션 제어
- ▶ GRANT, REVOKE 등

관계형 데이터베이스

- ▶ 논리적(개념적) 데이터 모델링과 물리적인 데이터베이스



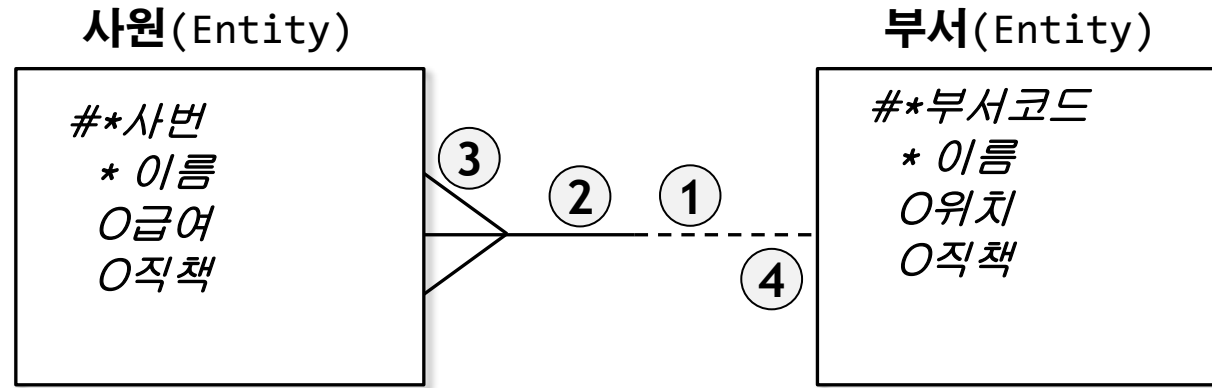
- ▶ RDBMS의 이해

- ▶ Relation 또는 2차원 Table을 이용하여 정보 저장

- ▶ Relation = Table 간의 연관
- ▶ Attribute = Column
- ▶ Tuple = Row

관계형 데이터베이스

▶ ERD 예시를 통해 보는 관계



1. 어떤 부서는 사원을 배치 받지 않을 수 있다 (점선)
2. 사원은 특정 부서에 소속되어 있다 (실선)
3. 한 부서에는 여러 명의 사원이 소속되어 있다 (다중선)
4. 한 사원은 하나의 부서에만 속한다 (단일선)

: 대표값 - PK(Primary Key)
* : NOT NULL
0 : NUL 가능

관계형 데이터베이스

▶ 테이블 Table Student column(Attribute)

student_id	name	grade	dept
1	정성진	1	컴퓨터
2	박현진	2	수학
3	홍길동	4	물리
...

row(record)

field field

테이블 : RDBMS의 기본적 저장 구조. 한 개 이상의 column과 0개 이상의 row로 구성

열(Column) : 테이블상에서의 단일 종류의 데이터를 나타냄. 특정 데이터 타입 및 크기를 가지고 있음

행(Row) : Column 값들의 조합. 튜플, 레코드라고 함.

기본키(PK)에 의해 구분된다. 기본키는 중복을 허용하지 않으며 없어서는 안된다

Field : Row와 Column의 교차점으로 Field는 데이터를 포함할 수 있고 없을 때는 NULL 값을 가지고 있다

관계형 데이터베이스

: 무결성 제약조건(Integrity Constraint)

▶ 개체 무결성(Entity Integrity)

- ▶ Table은 중복된 ROW를 가질 수 없으며 모든 Table은 각각의 ROW를 유일하게 식별할 수 있는 Column의 집합을 가진다. 이러한 Column의 집합 중에서 대표되는 컬럼을 Primary Key(PK)로 정의한다
- ▶ Primary Key의 값은 항상 유일(Unique)하며 널(Null)을 허용해서는 안된다

▶ 참조 무결성(Referential Integrity)

- ▶ Table들은 Foreign Key(FK)를 통해 서로 연결되어 있다. Foreign Key는 다른 Table 또는 자신 Table의 PK 값을 참조하기 위해 복사하여 가지고 있는 Column을 말한다
- ▶ 참조 무결성이 지켜지기 위해서 FK Column의 값은 참조하는 PK 컬럼 값 중의 하나이거나 Null이어야 한다

▶ 무결성 제약은 DBMS 시스템이 자동으로 수행한다

관계형 데이터베이스

▶ Primary Key (PK) : 기본키

- ▶ 관계(Relation)에서 튜플을 구분하기 위하여 사용하는 기본 키
- ▶ 하나의 애트리뷰트, 또는 애트리뷰트의 집합(복합키) 가능
- ▶ 관리자에 의해 릴레이션 생성시 정의된 (자동으로 Index 생성)
- ▶ 동일한 PK를 지닌 레코드는 존재할 수 없음

▶ 기타

- ▶ Candidate Key (후보키) : 튜플을 식별할 수 있는 최소한의 애트리뷰트 집합
 - ▶ 하나의 릴레이션에는 PK가 될 수 있는 키가 여러 개 있을 수 있음
 - ▶ 유일성과 희소성이 있으면 Candidate Key가 될 수 있음
- ▶ Alternative Key (대체키) : 후보키 중 기본 키가 아닌 것
- ▶ Composite Key (복합키) : 둘 이상의 애트리뷰트가 하나의 Key를 이루는 것

관계형 데이터베이스

▶ 학생

- ▶ 기본키 : 학번
- ▶ 후보키 : 기본키와 동일
- ▶ 주민등록번호 등이 있다면 후보키가 될 수 있음

학생
(STUDENT)

<u>학번</u> (Sno)	이름 (Sname)	학년 (Year)	학과 (Dept)
100	나 수 영	4	컴퓨터
200	이 찬 수	3	전기
300	정 기 태	1	컴퓨터
400	송 병 길	4	컴퓨터

▶ 등록

- ▶ 기본키 : (학번, 과목번호)
- ▶ 후보키 : 기본키와 동일
- ▶ 학번이나 과목번호만으로는 키가 되지 못함
- ▶ 등록번호와 같이 별도의 단일키를 추가하여 PK로 지정할 수도 있음

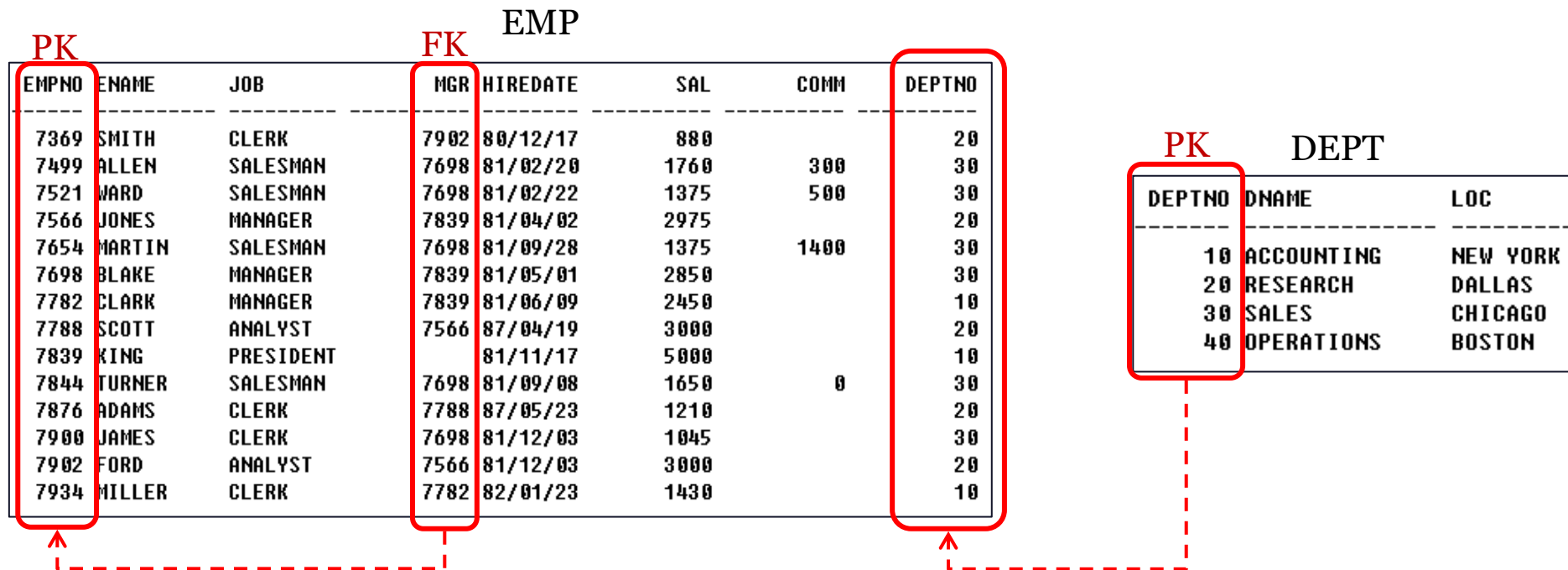
등록
(ENROL)

<u>학번</u> (Sno)	<u>과목번호</u> (Cno)	성적 (Grade)
100	C413	A
100	E412	A
200	C123	B
300	C312	A

관계형 데이터베이스

▶ Foreign Key (FK) : 외래키

- ▶ 기본키를 참조하는 애트리뷰트
- ▶ 다른 릴레이션의 튜플을 대표
- ▶ 릴레이션 간의 관계를 나타내기 위해 사용
- ▶ NULL 가능 (참조되지 않음을 의미)



Installation

MySQL 8과 Workbench 설치

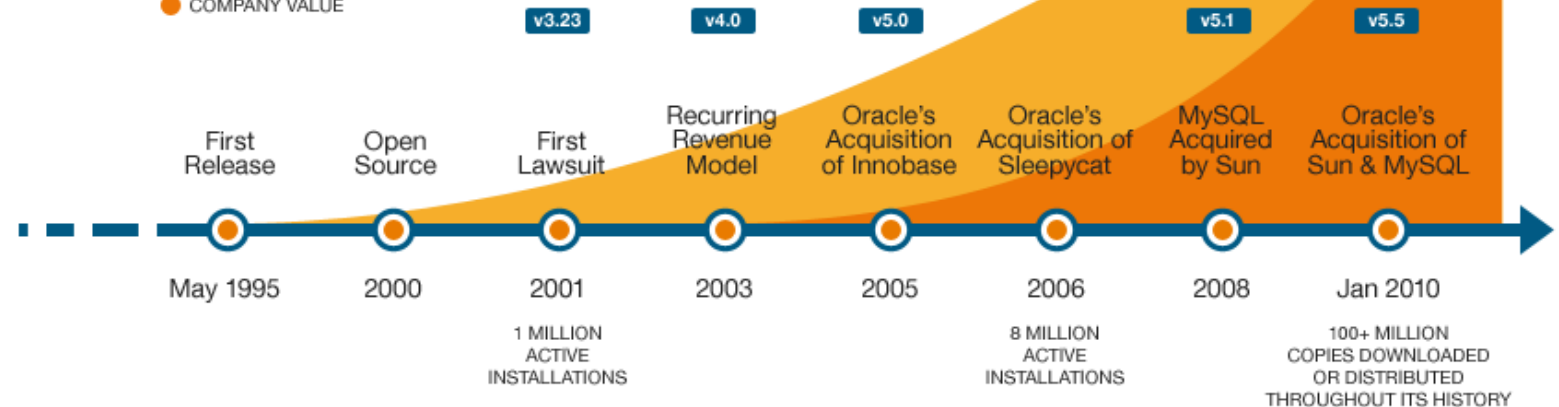
About MySQL

- ▶ 세계에서 가장 인기 있는 Open Source Database Management System
- ▶ 1995년 첫 공식 버전 출시
- ▶ 2001년 GNU GPL 등록
- ▶ 2008년 1월, Sun Microsystems에서 인수
- ▶ 2010년 1월, Oracle이 Sun Microsystems와 함께 인수
- ▶ Open Source LAMP Stack으로 급성장

MySQL Growth History

APPROXIMATE GROWTH

- COMMUNITY
- COMPANY VALUE



About MySQL

▶ 주요 기능


- ▶ 최상의 신뢰성과 보안성을 제공하는 오픈 소스 데이터베이스
- ▶ **Stored Procedure, Trigger, View** 등 RDBMS로서 기본 기능에 충실
- ▶ 사용자의 편의에 따른 **Pluggable Storage Engine** 기능
- ▶ 다양한 **Third Party** 엔진 지원
- ▶ 마법사 툴을 이용한 쉬운 설치 및 환경 설정
- ▶ 다양한 관리자용 **GUI Tool** 제공(Administration, Migration, Backup, Workbench, Query Browser 등)
- ▶ 중앙 집중 관리(보안, 스키마 관리, **Replication**, 성능 모니터링 등)
- ▶ 다양한 플랫폼과 다양한 언어 지원
- ▶ 가격 대비 최대 성능 효과의 **TCO** 절감 **DBMS**

Installation

: MySQL 8.0

- ▶ MySQL 홈페이지에서 MySQL 8.0 Community 다운로드 후 설치
- ▶ 직접 다운로드 경로
 - ▶ <https://dev.mysql.com/downloads/mysql/>
 - ▶ 시스템 운영체제에 맞는 설치 파일 선택 후 다운로드
- ▶ 다운로드 받은 Installer를 실행

Recommended Download:



MySQL Installer
for Windows

**All MySQL Products. For All Windows Platforms.
In One Package.**

Starting with MySQL 5.6 the MySQL Installer package replaces the standalone MSI packages.

Windows (x86, 32 & 64-bit), MySQL Installer MSI [Go to Download Page >](#)

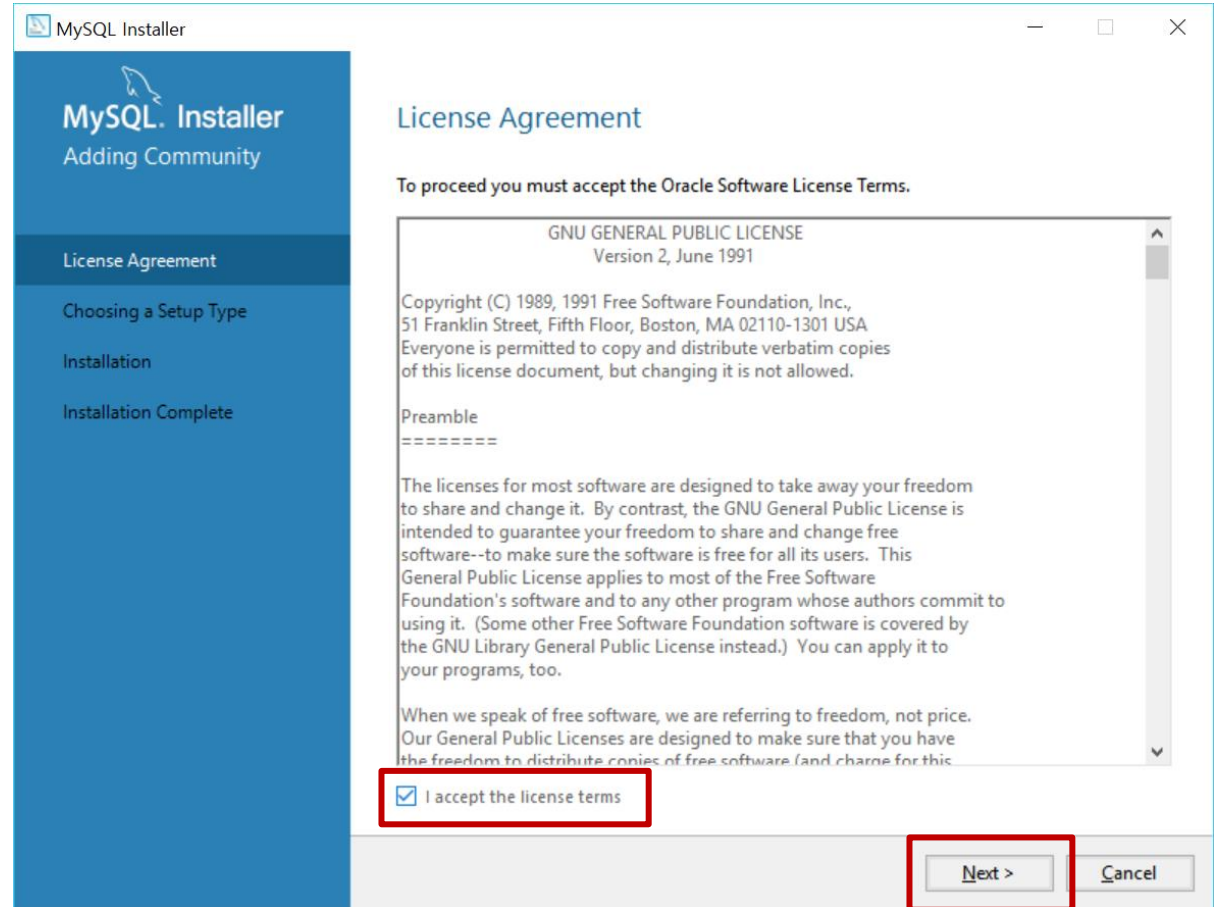
Other Downloads:

Windows (x86, 64-bit), ZIP Archive <small>(mysql-8.0.12-winx64.zip)</small>	8.0.12	182.9M	Download
<small>MD5: 2093a8d1cc66224b135d03c6883a17cb Signature</small>			
Windows (x86, 64-bit), ZIP Archive Debug Binaries & Test Suite <small>(mysql-8.0.12-winx64-debug-test.zip)</small>	8.0.12	239.1M	Download
<small>MD5: 391b4299fca29a3843bdac556819be59 Signature</small>			

Installation

: MySQL 8.0

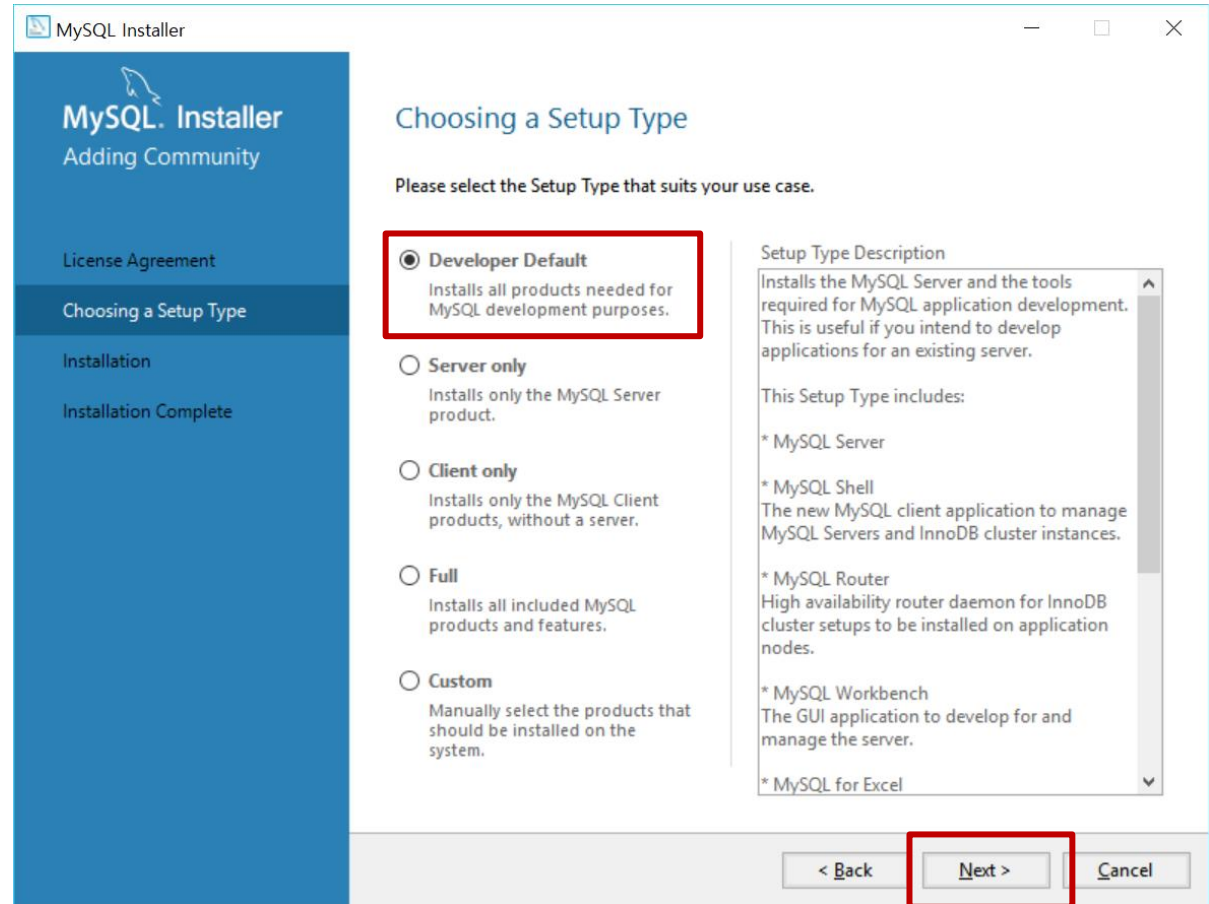
▶ License 허용 > Next



Installation

: MySQL 8.0

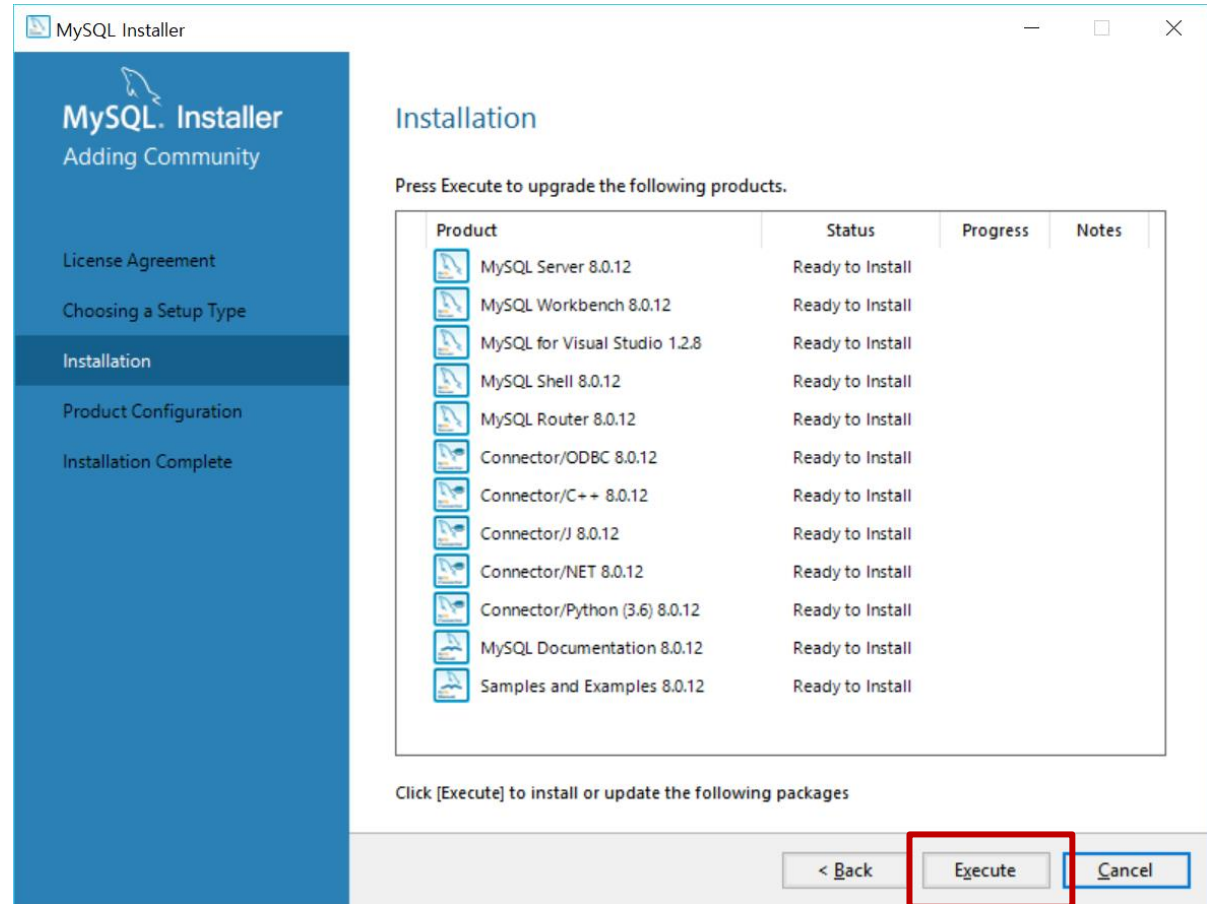
- Setup Type : Developer Default
> Next



Installation

: MySQL 8.0

- ▶ 설치할 Product 선택 > Execute
- ▶ 설치 완료 후 > Next

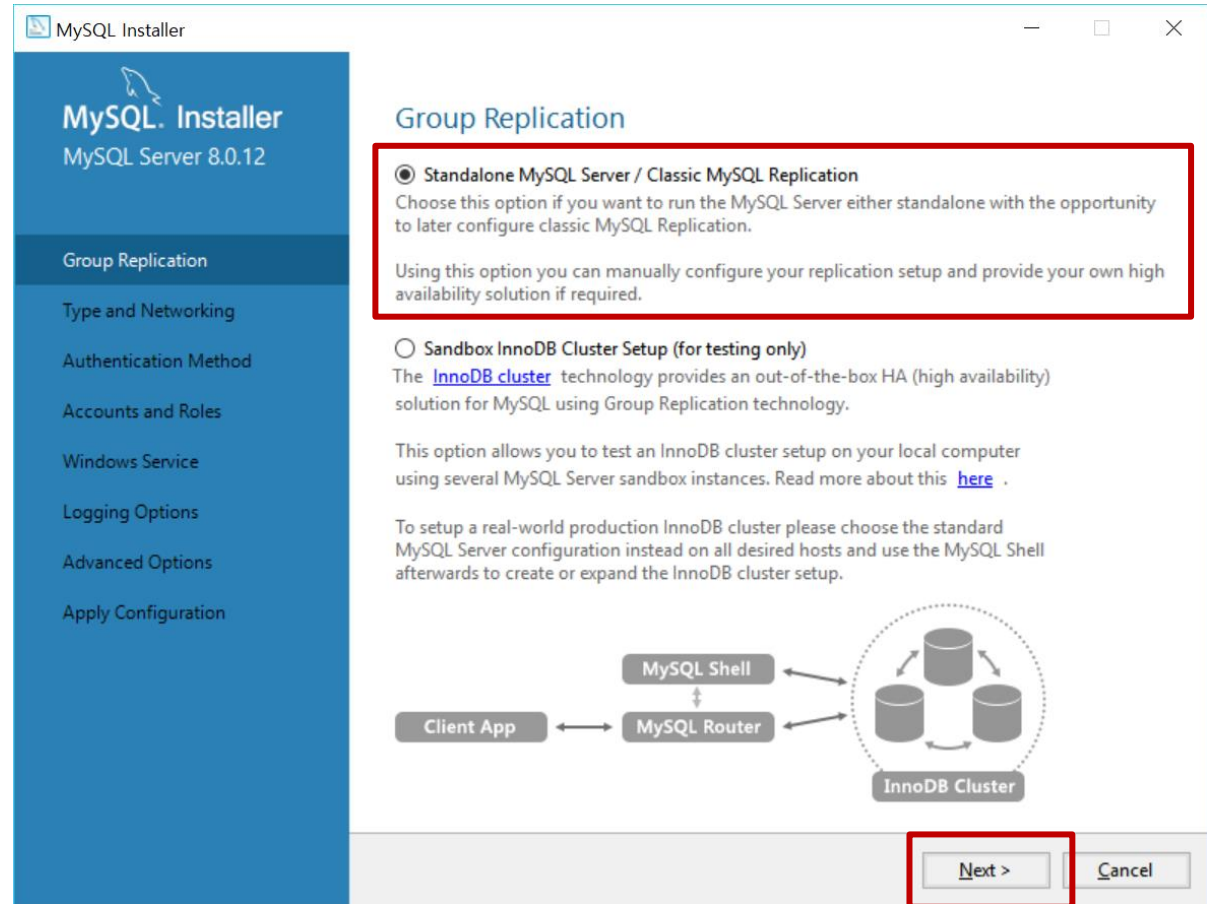


Installation

: MySQL 8.0

▶ Product Configuration

- ▶ Standalone MySQL Server 선택
> Next



Installation

: MySQL 8.0

▶ Product Configuration

▶ MySQL 접속 정보 설정

▶ MySQL의 기본 포트는 3306

MySQL Installer

MySQL Server 8.0.12

Group Replication

Type and Networking

Authentication Method

Accounts and Roles

Windows Service

Apply Configuration

Type and Networking

Server Configuration Type

Choose the correct server configuration type for this MySQL Server installation. This setting will define how much system resources are assigned to the MySQL Server instance.

Config Type: Development Computer

Connectivity

Use the following controls to select how you would like to connect to this server.

☒ TCP/IP Port: 3306 X Protocol Port: 33060

☒ Open Windows Firewall ports for network access

☐ Named Pipe Pipe Name: MYSQL

☐ Shared Memory Memory Name: MYSQL

Advanced Configuration

Select the check box below to get additional configuration pages where you can set advanced and logging options for this server instance.

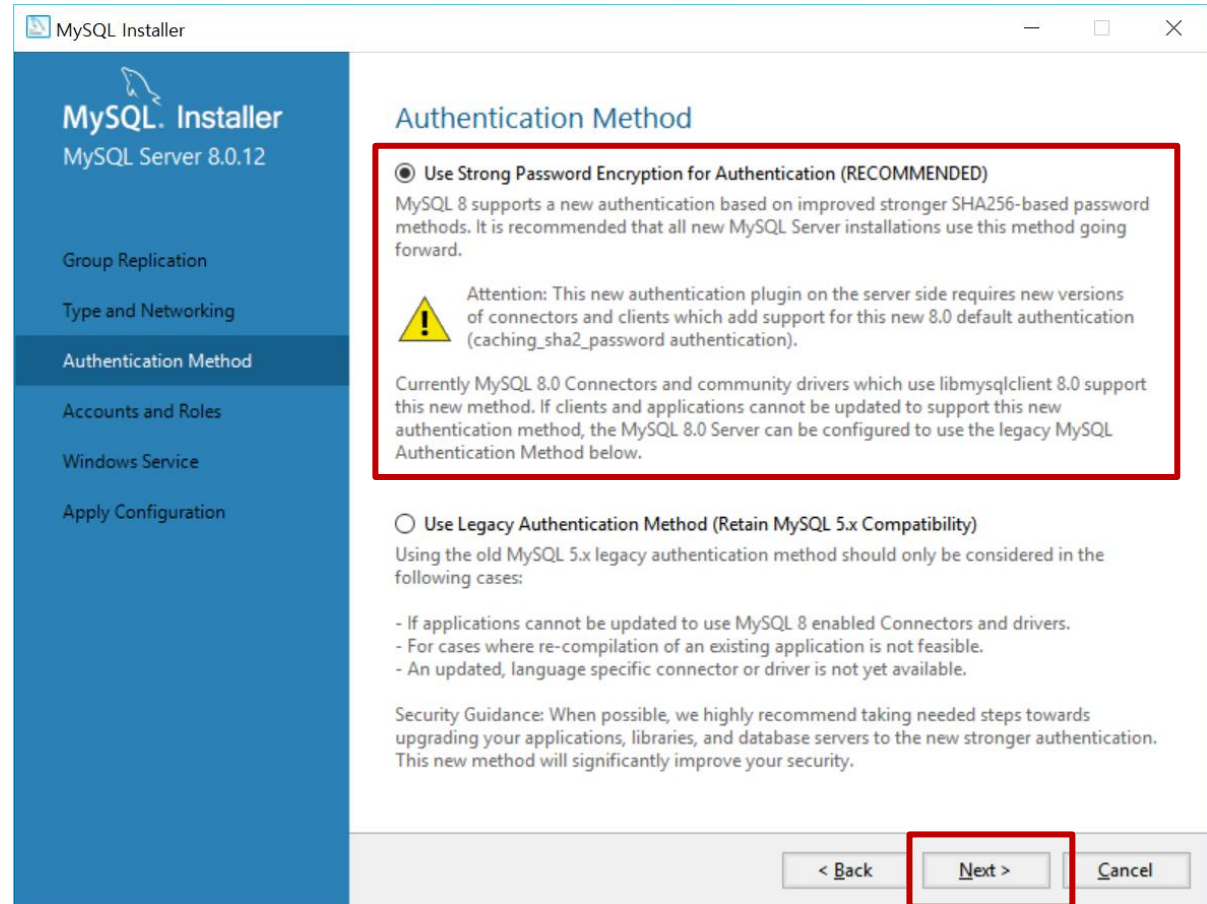
☐ Show Advanced and Logging Options

< Back Next > Cancel

Installation

: MySQL 8.0

- ▶ 인증 방식의 설정
- > Next



Installation

: MySQL 8.0

- ▶ 관리자 계정 암호 설정
 - ▶ MySQL의 관리자 계정은 **root**
 - ▶ **root** 계정은 DB의 모든 기능을 조작, 제어할 수 있으므로 **root** 계정의 암호는 주의해서 관리 및 사용

MySQL Installer

MySQL Server 8.0.12

Group Replication

Type and Networking

Authentication Method

Accounts and Roles

Windows Service

Apply Configuration

Accounts and Roles

Root Account Password
Enter the password for the root account. Please remember to store this password in a secure place.

MySQL Root Password:

Repeat Password:

Password strength: **Weak**

MySQL User Accounts

Create MySQL user accounts for your users and applications. Assign a role to the user that consists of a set of privileges.

MySQL Username	Host	User Role
----------------	------	-----------

[Add User](#)
[Edit User](#)
[Delete](#)

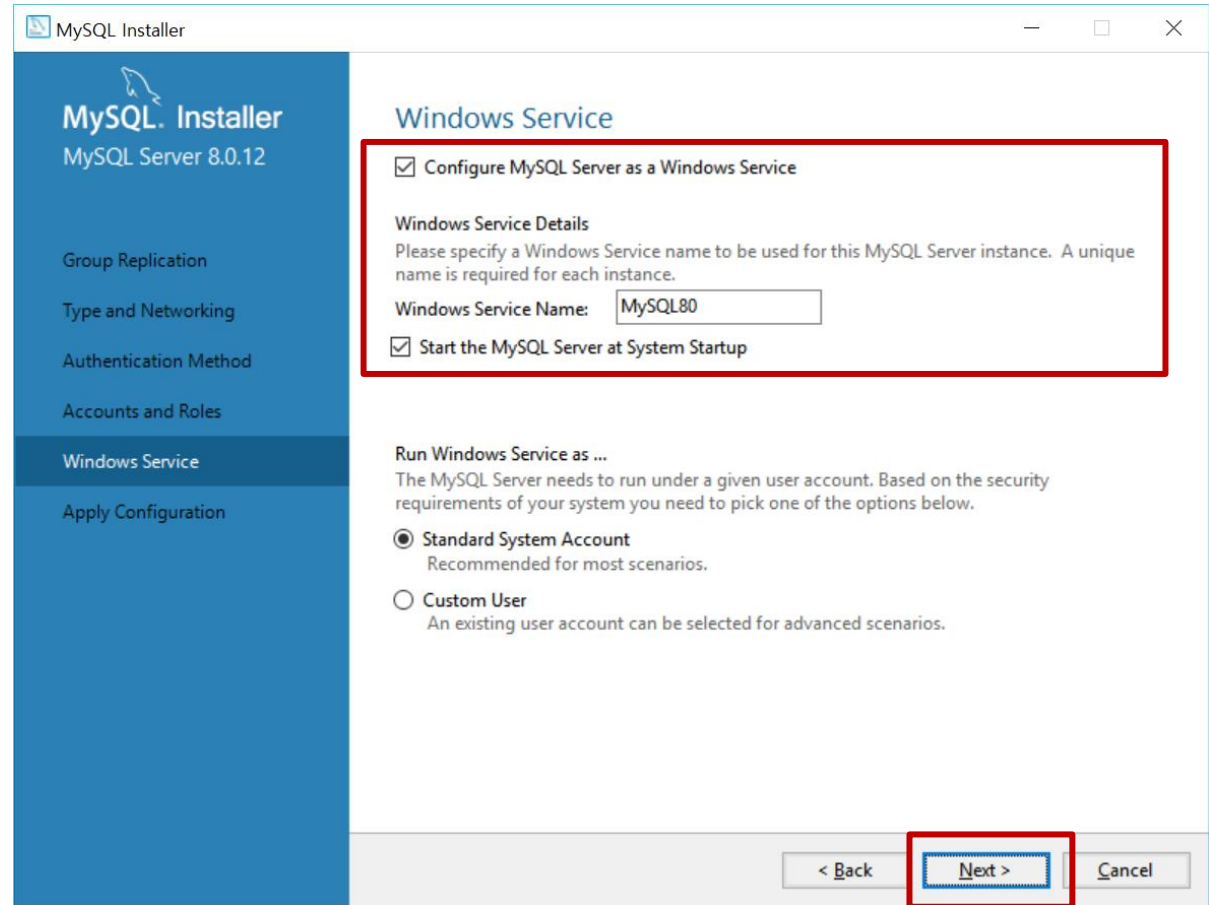
< Back **Next >** Cancel

Installation

: MySQL 8.0

▶ Windows Service에 등록

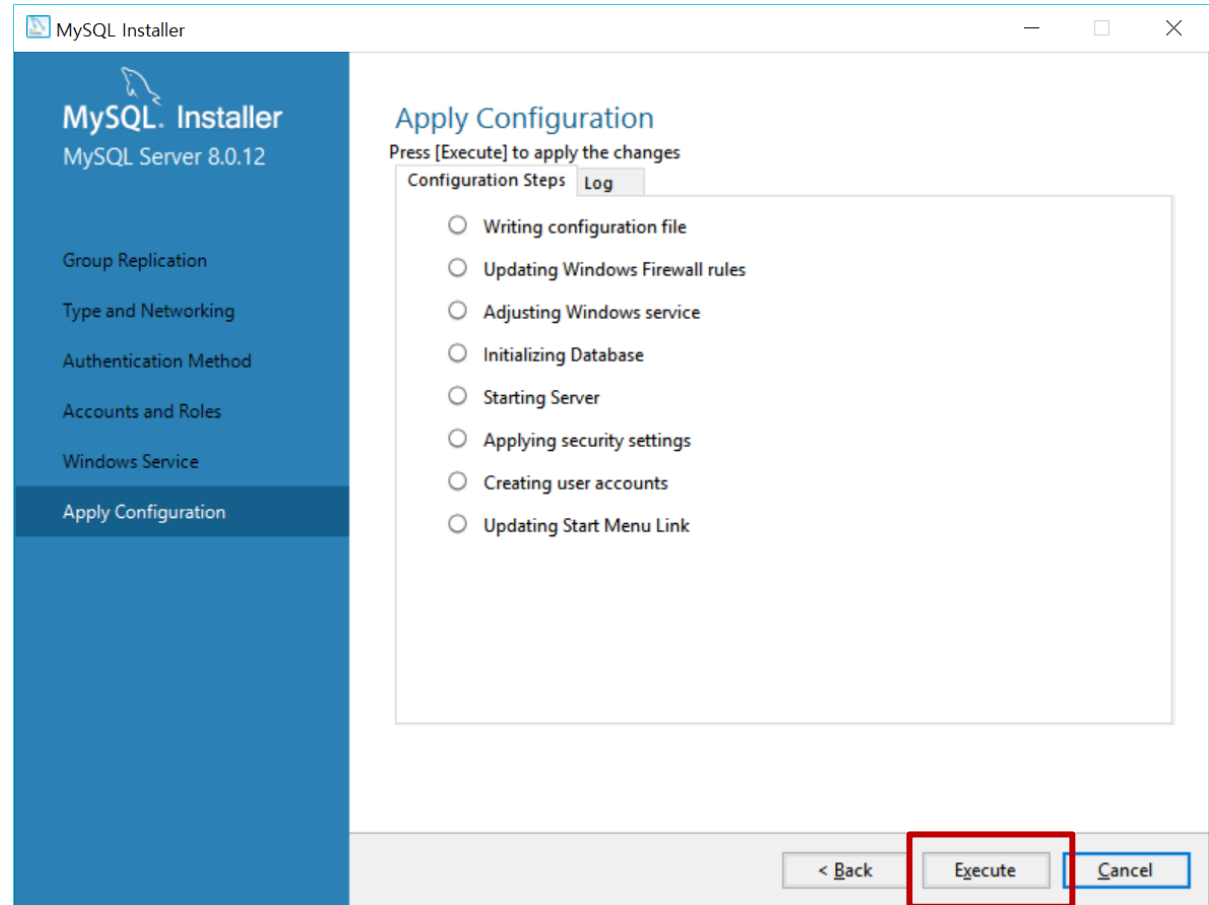
- ▶ Service에 등록하면 윈도우가 실행될 때 자동으로 관리되고 실행되도록 설정할 수 있음



Installation

: MySQL 8.0

- ▶ 설정 저장
> Execute
- ▶ Production 별로 추가 설정해야 할
내용이 있다면 Instruction을 따라서
설정 후 반영



Connect to Database

: with Shell Command

- ▶ Win + R > cmd
- ▶ 셸 프롬프트에 다음과 같이 입력

```
mysql -uroot -p
```

- ▶ root 계정의 비밀번호 입력

```
Welcome to the MySQL monitor.  Commands end with ; or \g.  
Your MySQL connection id is 15  
Server version: 8.0.12 MySQL Community Server - GPL
```

```
...
```

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

```
mysql>
```

Basic Queries

: with Shell Command

▶ 보류 데이터베이스 확인

```
mysql> SHOW DATABASES;
+-----+
| Database                |
+-----+
| information_schema      |
| mysql                   |
| performance_schema     |
| sakila                   |
| sys                     |
| world                   |
+-----+
6 rows in set (0.02 sec)
```

▶ 데이터베이스 사용

```
mysql> USE sakila
Database changed
mysql> SHOW TABLES;
+-----+
| Tables_in_sakila        |
+-----+
| actor                   |
| actor_info              |
| address                 |
| ...                     |
| staff                   |
| staff_list              |
| store                   |
+-----+
23 rows in set (0.01 sec)
```

Basic Queries

: with Shell Command

▶ 테이블 구조 확인

```
mysql> DESCRIBE actor;
```

Field	Type	Null	Key	Default	Extra
actor_id	smallint(5) unsigned	NO	PRI	NULL	auto_increment
first_name	varchar(45)	NO		NULL	
last_name	varchar(45)	NO	MUL	NULL	
last_update	timestamp	NO		CURRENT_TIMESTAMP	on update CURRENT_TIMESTAMP

4 rows in set (0.00 sec)

▶ DESCRIBE 명령은 DESC로 줄여 사용할 수 있다

▶ 연습) 다른 테이블의 구조도 확인해 봅시다

Basic Queries

: with Shell Command

▶ 간단한 쿼리 실행

```
mysql> SELECT VERSION(), CURRENT_DATE;  
+-----+-----+  
| VERSION() | CURRENT_DATE |  
+-----+-----+  
| 8.0.12    | 2018-10-17   |  
+-----+-----+  
1 row in set (0.00 sec)
```

- ▶ 대부분 MySQL 문장은 semicolon(;)으로 끝난다
- ▶ MySQL은 찾은 전체 row를 출력하고 마지막에 전체 row 수와 쿼리 실행에 걸린 시간을 표시한다.

Basic Queries

: with Shell Command

- ▶ 키워드는 대소문자 구별을 하지 않는다
- ▶ 다음 쿼리들은 모두 같다

```
mysql> SELECT VERSION(), CURRENT_DATE;  
mysql> select version(), current_date;  
mysql> SeLeCt VeRsIoN(), current_DATE;
```

- ▶ 다양한 수학 함수를 내장하고 있으며 계산식의 처리도 수행

```
mysql> SELECT SIN(PI()/4), (4+1)*5;  
+-----+-----+  
| SIN(PI()/4) | (4+1)*5 |  
+-----+-----+  
| 0.7071067811865476 | 25 |  
+-----+-----+
```

Basic Queries

: with Shell Command

- ▶ 여러 문장을 세미콜론(;)을 붙여 연속으로 입력하면 여러 쿼리를 순차적으로 실행할 수 있다.

```
mysql> SELECT VERSION(); SELECT NOW();
```

```
+-----+
```

```
| VERSION() |
```

```
+-----+
```

```
| 8.0.12 |
```

```
+-----+
```

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

```
+-----+
```

```
| NOW() |
```

```
+-----+
```

```
| 2018-10-17 14:14:00 |
```

```
+-----+
```

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

Basic Queries

: with Shell Command

► Multi-Line Commands

- MySQL은 세미콜론(;)을 문장의 마지막으로 인식하기 때문에, 여러 줄에 걸쳐 문장을 쓰는 것도 가능
- 긴 쿼리 작성시 가독성을 높이고자 할 때 유용

```
mysql> SELECT
->   USER(),
->   CURRENT_DATE
-> ;
```

USER()	CURRENT_DATE
root@localhost	2018-10-17

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


Basic Queries

: with Shell Command

▶ Command 취소

- ▶ 긴 쿼리를 작성하다가 중간에 취소해야 하는 경우
즉시 \c를 붙여 쿼리 작성을 중단할 수 있다.

```
mysql> SELECT  
      -> USER()  
      -> \c  
mysql>
```

▶ MySQL Shell 종료 (연결 끊기)

```
mysql> QUIT  
mysql> EXIT
```

Sample Database

▶ Employees Database 설치

- ▶ https://github.com/datacharmer/test_db 저장소 다운로드
- ▶ 쉘 프롬프트에서 다음과 같이 sql 파일을 실행

```
C:>mysql -uroot -p < employees.sql
```

- ▶ 데이터베이스가 잘 설치되었는지 확인

```
C:>mysql -t -uroot -p < test_employees_md5.sql
```

```
...
```

+-----+	+-----+	+-----+
table_name	records_match	crc_match
+-----+	+-----+	+-----+
employees	OK	ok
...		
salaries	OK	ok
+-----+	+-----+	+-----+

Sample Database

▶ 사용자 생성

- ▶ 사용자 생성을 위해서는 'CREATE USER' command 이용

```
CREATE USER 'dev'@'localhost' IDENTIFIED BY '{password}';
```

- ▶ 특정 데이터베이스에 접근할 권한을 주려면 'GRANT' command 이용

```
GRANT all privileges ON employees.* TO 'dev'@'localhost';
```

▶ 사용자 삭제

- ▶ 사용자 삭제를 위해서는 'DROP USER' command 이용

```
DROP USER 'dev'@'localhost';
```

- ▶ dev@localhost 사용자를 만들고 employees의 모든 테이블에 대해 전체 권한을 부여한 후, 해당 계정으로 데이터베이스에 접속하여 employees 데이터베이스를 사용해 봅니다.

Sample Database

: 연습

- ▶ 저장소의 **sql** 파일을 받아 **hr** 샘플 데이터베이스를 설치해 봅니다.
 - ▶ SQL 디렉터리 **hr.sql**
- ▶ **bituser@localhost** 사용자를 생성하고 **hr** 데이터베이스에 대한 모든 권한을 부여해 봅니다.
- ▶ **hr-schema.pdf** 파일을 통해 **HR** 데이터베이스의 구성을 살펴 보시다.