데이터베이스설계 (ICE4016)

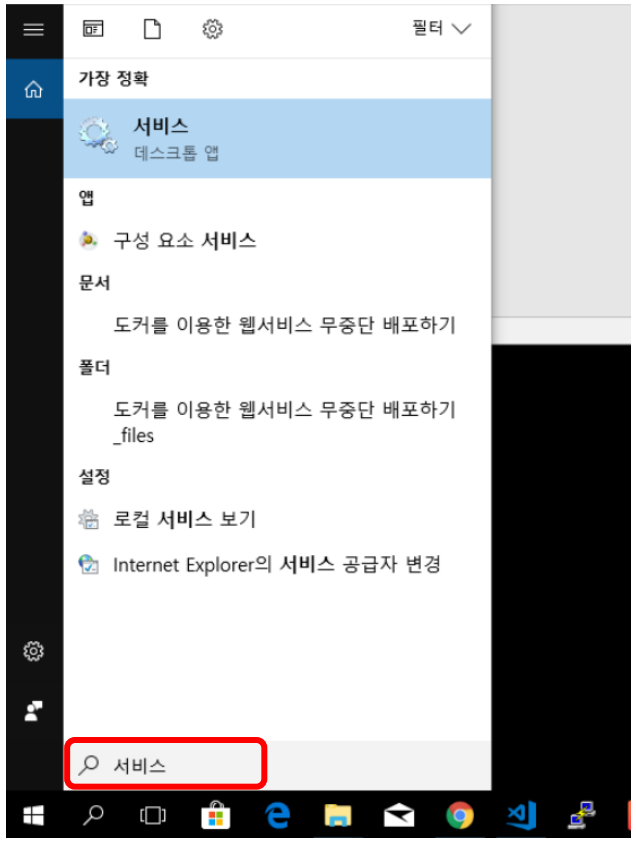
MySQL 기본 사용법

Prof. Wonik Choi

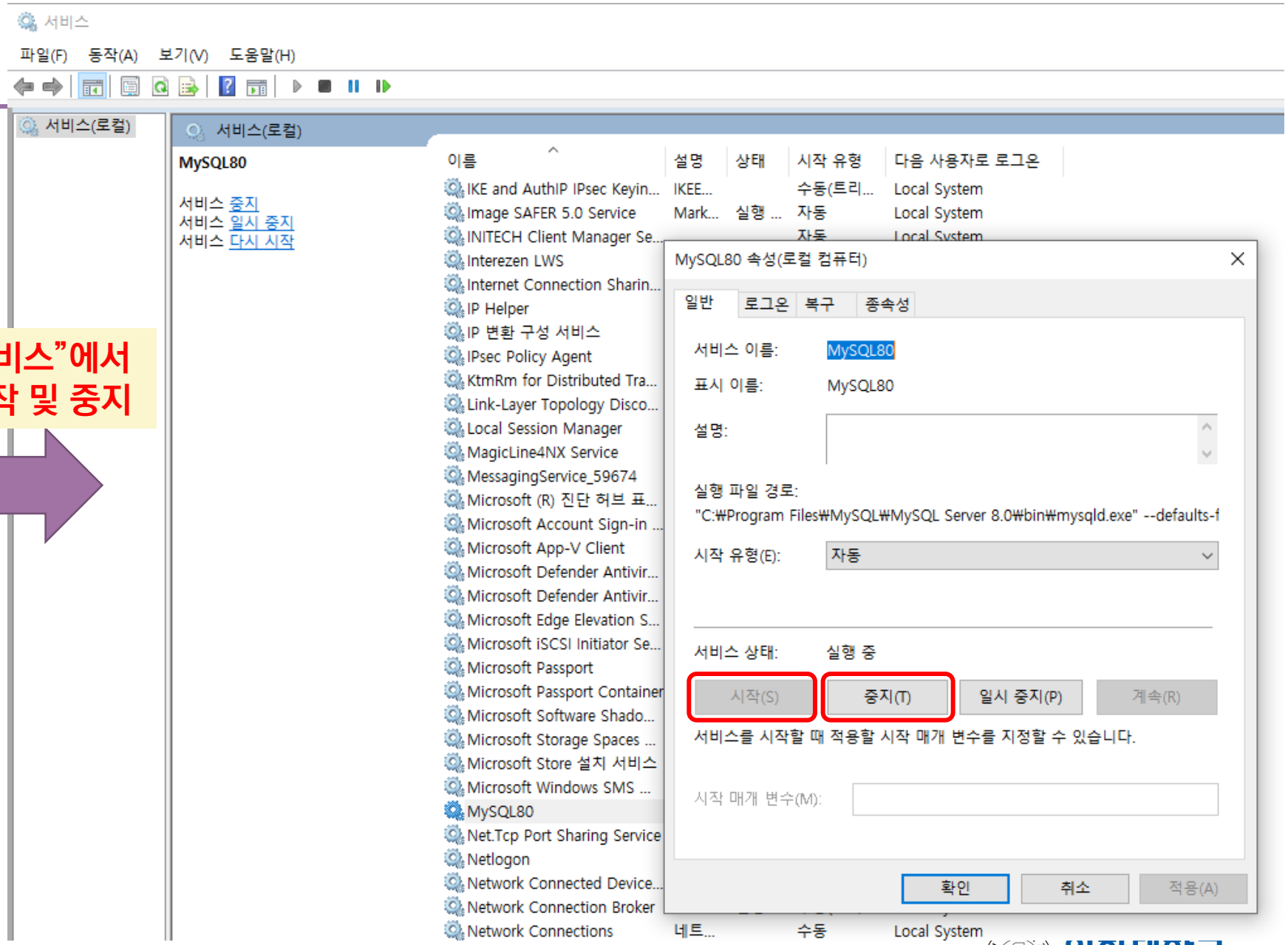
데이터 인텔리전스 연구실
Data Intelligence Laboratory

인하대학교
INHA UNIVERSITY

실행 및 중지

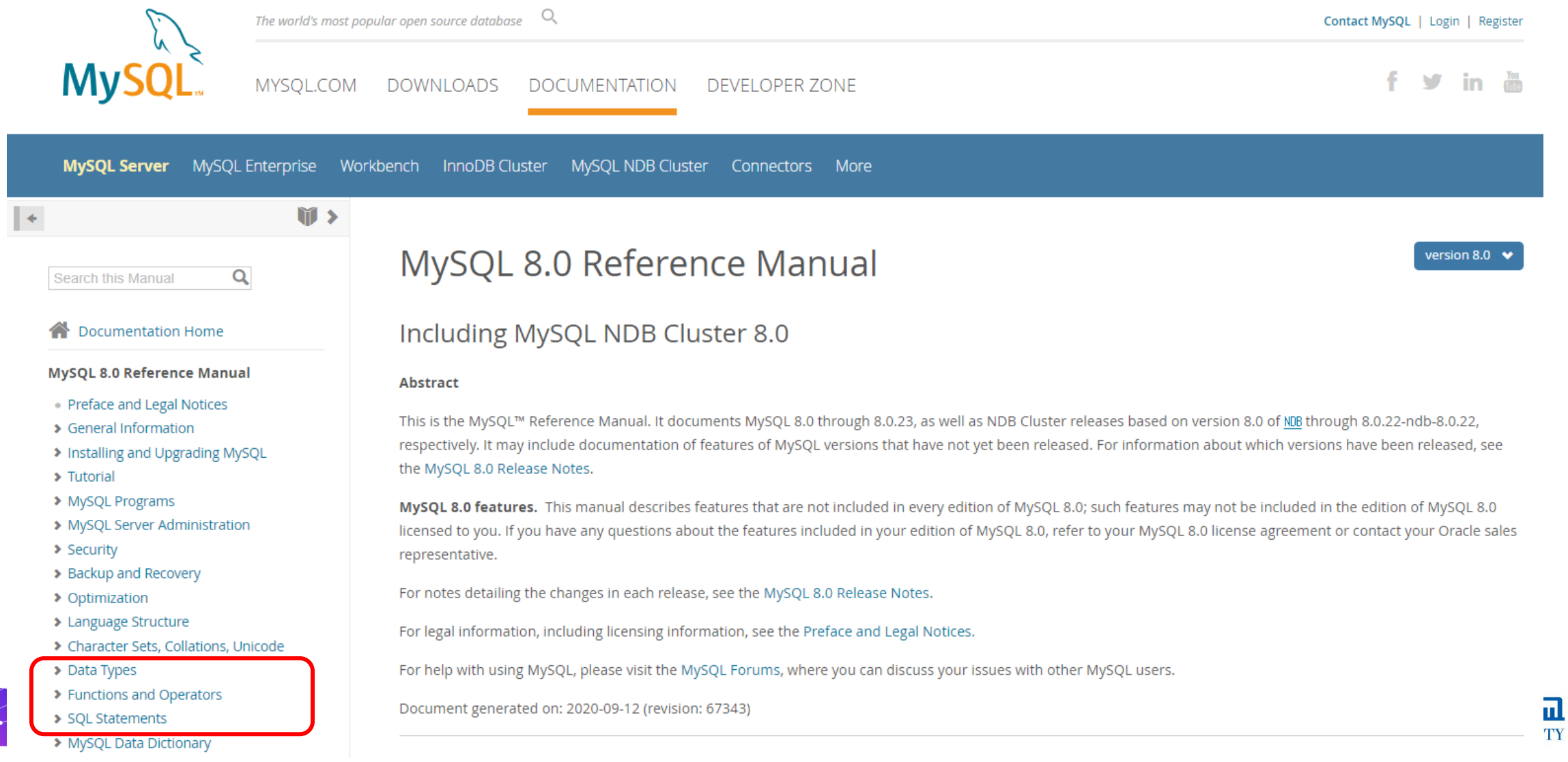


"서비스"에서 시작 및 중지



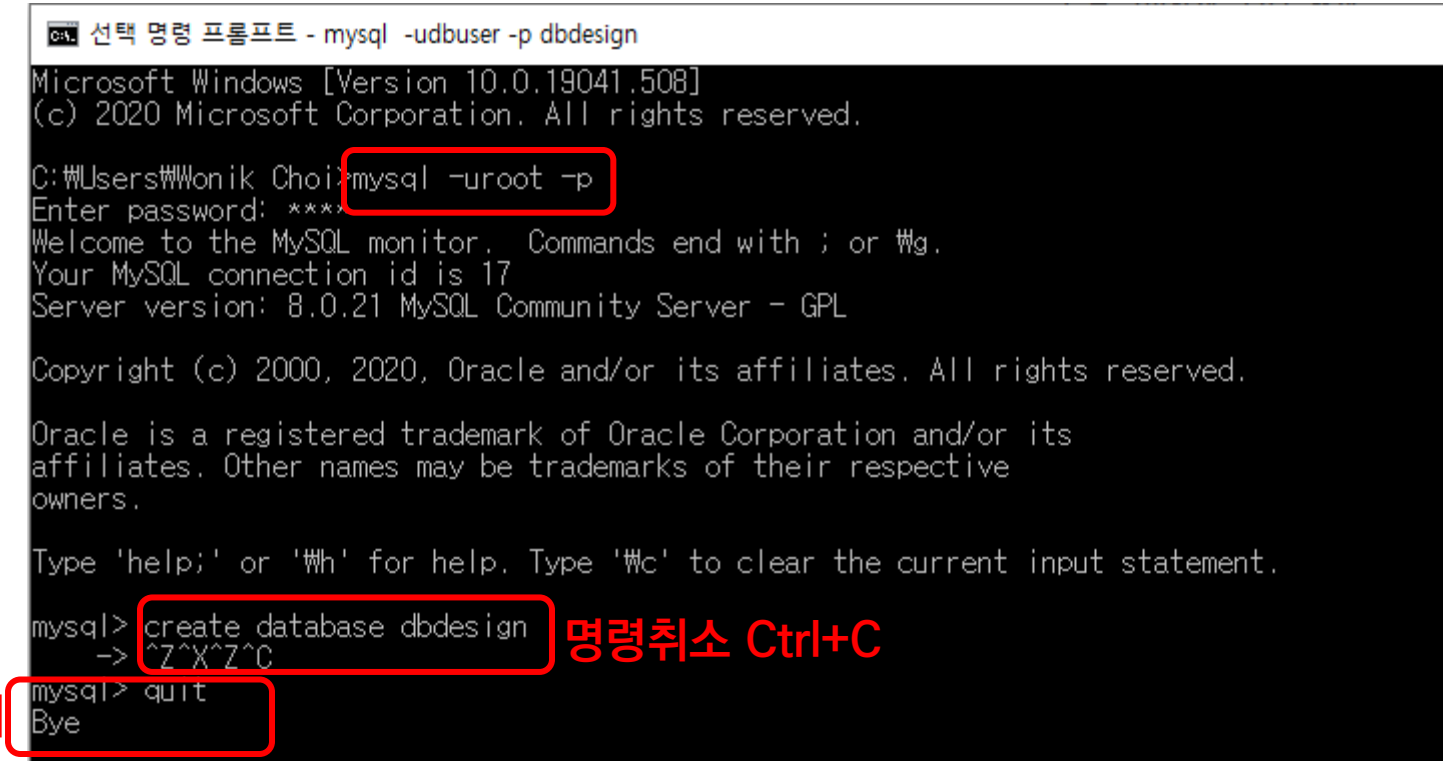
MySQL Reference Manual

- https://dev.mysql.com/doc/refman/8.0/en/



MySQL 접속 및 해제

○ mysql 접속



Database 생성하기, 사용자 생성하기

- mysql> create database DB이름;
- 예

```
mysql> create database dbdesign;
Query OK, 1 row affected (0.01 sec)
```

- mysql> CREATE USER ‘username'@'localhost' IDENTIFIED BY '*password*';
- 예

```
mysql> create user 'dbuser'@'%' identified by 'dbuser123!';
Query OK, 0 rows affected (0.01 sec)
```

권한(privilege) 주기 (GRANT, REVOKE)

- GRANT ALL ON db1.* TO ‘username'@'localhost';

```
mysql> grant all privileges on dbdesign.* to dbuser@'%';
Query OK, 0 rows affected (0.01 sec)

mysql> flush privileges;
Query OK, 0 rows affected (0.01 sec)
```

DB에 적용하라는 의미. 반드시 실행해야 함

serverTimezone 설정

- database를 생성하면 serverTimezone을 설정해주어야 함

```
mysql> SET GLOBAL time_zone = '+9:00';
Query OK, 0 rows affected (0.00 sec)

mysql> SET time_zone = '+9:00';
Query OK, 0 rows affected (0.00 sec)
```

- 또는 my.ini 에 설정

```
# SERVER SECTION
# ----------------------------------------------------------------------
#
# The following options will be read by the MySQL Server. Make sure that
# you have installed the server correctly (see above) so it reads this
# file.
#
# server_type=3
[mysqld]
# default-time-zone='+9:00'»    # KST
```

- 또는 connection string에 설정
  - "jdbc:mysql://localhost/dbdesign?serverTimezone=Asia/Seoul"

새 사용자로 접속 및 실습예

```
C:₩Users₩Wonik Choi>mysql -udbuser -p dbdesign
Enter password: **********
Welcome to the MySQL monitor.  Commands end with ; or ₩g.
Your MySQL connection id is 20
Server version: 8.0.21 MySQL Community Server - GPL

Copyright (c) 2000, 2020, 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 version(), current_date;
+-----------+--------------+
| version() | current_date |
+-----------+--------------+
| 8.0.21    | 2020-09-13   |
+-----------+--------------+
1 row in set (0.00 sec)

mysql> SELECT SIN(PI()/4), (4+1)*5;
+------------------+---------+
| SIN(PI()/4)      | (4+1)*5 |
+------------------+---------+
| 0.7071067811865476 |      25 |
+------------------+---------+
1 row in set (0.00 sec)
```

내장 함수 사용

내장 함수 사용

```
mysql> SELECT VERSION(); SELECT NOW();
+-----------+
| VERSION() |
+-----------+
| 8.0.21    |
+-----------+
1 row in set (0.00 sec)

+---------------------+
| NOW()               |
+---------------------+
| 2020-09-13 12:04:09 |
+---------------------+
1 row in set (0.00 sec)
```

내장 함수 사용

```
mysql> select user(),
    -> current_date;
+-------------------+--------------+
| user()            | current_date |
+-------------------+--------------+
| dbuser@localhost | 2020-09-13   |
+-------------------+--------------+
1 row in set (0.00 sec)
```

여러 라인으로 입력해도 됨

Table 만들기 및 기본적인 SQL문

```
mysql> show databases;
+--------------------+
| Database           |
+--------------------+
| dbdesign           |
| information_schema |
+--------------------+
2 rows in set (0.00 sec)
```

database 목록보기

```
mysql> show tables;
Empty set (0.01 sec)
```

table 목록보기

```
mysql> CREATE TABLE STUDENT ( Name VARCHAR(30) NOT NULL,
    -> StudentNumber INTEGER NOT NULL,
    -> Class CHAR NOT NULL,
    -> Major CHAR(4),
    -> PRIMARY KEY (StudentNumber) );
Query OK, 0 rows affected (0.09 sec)
```

student 릴레이션 생성

```
mysql> desc student;
+---------------+------------+------+-----+---------+-------+
| Field         | Type       | Null | Key | Default | Extra |
+---------------+------------+------+-----+---------+-------+
| Name          | varchar(30)| NO   |     | NULL    |       |
| StudentNumber | int        | NO   | PRI | NULL    |       |
| Class         | char(1)    | NO   |     | NULL    |       |
| Major         | char(4)    | YES  |     | NULL    |       |
+---------------+------------+------+-----+---------+-------+
4 rows in set (0.00 sec)
```

table 스키마 보기(describe)

```
mysql> select name from student;
Empty set (0.00 sec)
```

select문 실행

```
mysql> select name1 from student
ERROR 1054 (42S22): Unknown column 'name1' in 'field list'
mysql> select NAME from student;
Empty set (0.00 sec)
```

속성이름 오류

SQL은 case insensitive

```
mysql> insert into student values ('choi', 1, 1,'com');
Query OK, 1 row affected (0.01 sec)
```

insert 문 실행

```
mysql> select NAME from student;
+------+
| NAME |
+------+
| choi |
+------+
1 row in set (0.00 sec)
```

```
mysql> select nAmE from student;
+------+
| nAmE |
+------+
| choi |
+------+
1 row in set (0.00 sec)
```

SQL은 case insensitive

```
mysql> insert into student values ('choi', 1, 1,'com');
ERROR 1062 (23000): Duplicate entry '1' for key 'student.PRIMARY'
mysql> select * from student;
+------+---------------+-------+-------+
| Name | StudentNumber | Class | Major |
+------+---------------+-------+-------+
| choi |             1 | 1     | com   |
+------+---------------+-------+-------+
1 row in set (0.00 sec)
```

참조 무결성 오류

```
mysql> select name from student where studentnumber=1;
+------+
| name |
+------+
| choi |
+------+
1 row in set (0.00 sec)
```

predicate사용

SQL (Structured Query Language)

- SELECT : 데이터베이스에서 데이터를 검색할 때 사용되며, 데이터를 조회하는 데 사용됨
- INSERT : 데이터베이스 테이블에 새로운 데이터를 추가하는 데 사용됨
- UPDATE : 데이터베이스 테이블의 기존 데이터를 수정하는 데 사용됨
- DELETE : 데이터베이스 테이블에서 데이터를 삭제하는 데 사용됨
- CREATE : 데이터베이스 객체(테이블, 뷰, 인덱스 등)를 생성하는 데 사용됨
- ALTER : 데이터베이스 객체의 구조를 수정하는 데 사용됨
- DROP : 데이터베이스 객체(테이블, 뷰, 인덱스 등)를 삭제하는 데 사용됨

SQL (Structured Query Language)

- SELECT first_name, last_name FROM employee WHERE salary >= 50000;
- SELECT name, salary FROM employee ORDER BY salary ASC/DESC;
- INSERT INTO employee (first_name, last_name, salary) VALUES ('John', 'Doe', 60000);
- UPDATE employee SET salary = 65000 WHERE first_name = 'John' AND last_name = 'Doe';
- DELETE FROM employee WHERE first_name = 'John' AND last_name = 'Doe';

SQL (Structured Query Language)

- SELECT e.first_name, e.last_name, c.client_name
  FROM employee e
  INNER JOIN works_with ww ON e.emp_id = ww.emp_id
  INNER JOIN client c ON ww.client_id = c.client_id;

- SELECT *
  FROM employee
  WHERE birth_day BETWEEN '1990-01-01' AND '2000-12-31';

- SELECT COUNT(*) AS total_customers FROM customer;

- SELECT department, AVG(salary) AS avg_salary FROM employees
  GROUP BY department;

COMPANY Database

Employee

emp_id	first_name	last_name	birth_date	sex	salary	super_id	branch_id
100	David	Wallace	1967-11-17	M	250,000	NULL	1
101	Jan	Levinson	1961-05-11	F	110,000	100	1
102	Michael	Scott	1964-03-15	M	75,000	100	2
103	Angela	Martin	1971-06-25	F	63,000	102	2
104	Kelly	Kapoor	1980-02-05	F	55,000	102	2
105	Stanley	Hudson	1958-02-19	M	69,000	102	2
106	Josh	Porter	1969-09-05	M	78,000	100	3
107	Andy	Bernard	1973-07-22	M	65,000	106	3
108	Jim	Halpert	1978-10-01	M	71,000	106	3

Branch

branch_id	branch_name	mgr_id	mgr_start_date
1	Corporate	100	2006-02-09
2	Scranton	102	1992-04-06
3	Stamford	106	1998-02-13

Client

client_id	client_name	branch_id
400	Dunmore Highschool	2
401	Lackawana Country	2
402	FedEx	3
403	John Daly Law, LLC	3
404	Scranton Whitepages	2
405	Times Newspaper	3
406	FedEx	2

Works_With

emp_id	client_id	total_sales
105	400	55,000
102	401	267,000
108	402	22,500
107	403	5,000
108	403	12,000
105	404	33,000
107	405	26,000
102	406	15,000
105	406	130,000

Branch Supplier

branch_id	supplier_name	supply_type
2	Hammer Mill	Paper
2	Uni-ball	Writing Utensils
3	Patriot Paper	Paper
2	J.T. Forms & Labels	Custom Forms
3	Uni-ball	Writing Utensils
3	Hammer Mill	Paper
3	Stamford Lables	Custom Forms

Labels

	Primary Key
	Foreign Key
	Attribute

https://www.giraffeacademy.com/databases/sql/company-database-intro/

COMPANY_Database_Schema.sql

```sql
CREATE TABLE employee (
  emp_id INT PRIMARY KEY,
  first_name VARCHAR(40),
  last_name VARCHAR(40),
  birth_day DATE,
  sex VARCHAR(1),
  salary INT,
  super_id INT,
  branch_id INT
);
CREATE TABLE branch (
  branch_id INT PRIMARY KEY,
  branch_name VARCHAR(40),
  mgr_id INT,
  mgr_start_date DATE,
  FOREIGN KEY(mgr_id) REFERENCES employee(emp_id) ON DELETE SET NULL
);
ALTER TABLE employee ADD FOREIGN KEY(branch_id) REFERENCES branch(branch_id) ON DELETE SET NULL;
ALTER TABLE employee ADD FOREIGN KEY(super_id) REFERENCES employee(emp_id) ON DELETE SET NULL;
CREATE TABLE client (
  client_id INT PRIMARY KEY,
  client_name VARCHAR(40),
  branch_id INT,
  FOREIGN KEY(branch_id) REFERENCES branch(branch_id) ON DELETE SET NULL
);
CREATE TABLE works_with (
  emp_id INT,
  client_id INT,
  total_sales INT,
  PRIMARY KEY(emp_id, client_id),
  FOREIGN KEY(emp_id) REFERENCES employee(emp_id) ON DELETE CASCADE,
  FOREIGN KEY(client_id) REFERENCES client(client_id) ON DELETE CASCADE
);
CREATE TABLE branch_supplier (
  branch_id INT,
  supplier_name VARCHAR(40),
  supply_type VARCHAR(40),
  PRIMARY KEY(branch_id, supplier_name),
  FOREIGN KEY(branch_id) REFERENCES branch(branch_id) ON DELETE CASCADE
);
```

```sql
-- Corporate
INSERT INTO employee VALUES(100, 'David', 'Wallace', '1967-11-17', 'M', 250000, NULL, NULL);
INSERT INTO branch VALUES(1, 'Corporate', 100, '2006-02-09');
UPDATE employee SET branch_id = 1 WHERE emp_id = 100;
INSERT INTO employee VALUES(101, 'Jan', 'Levinson', '1961-05-11', 'F', 110000, 100, 1);
-- Scranton
INSERT INTO employee VALUES(102, 'Michael', 'Scott', '1964-03-15', 'M', 75000, 100, NULL);
INSERT INTO branch VALUES(2, 'Scranton', 102, '1992-04-06');
UPDATE employee SET branch_id = 2 WHERE emp_id = 102;
INSERT INTO employee VALUES(103, 'Angela', 'Martin', '1971-06-25', 'F', 63000, 102, 2);
INSERT INTO employee VALUES(104, 'Kelly', 'Kapoor', '1980-02-05', 'F', 55000, 102, 2);
INSERT INTO employee VALUES(105, 'Stanley', 'Hudson', '1958-02-19', 'M', 69000, 102, 2);
-- Stamford
INSERT INTO employee VALUES(106, 'Josh', 'Porter', '1969-09-05', 'M', 78000, 100, NULL);
INSERT INTO branch VALUES(3, 'Stamford', 106, '1998-02-13');
UPDATE employee SET branch_id = 3 WHERE emp_id = 106;
INSERT INTO employee VALUES(107, 'Andy', 'Bernard', '1973-07-22', 'M', 65000, 106, 3);
INSERT INTO employee VALUES(108, 'Jim', 'Halpert', '1978-10-01', 'M', 71000, 106, 3);
-- BRANCH SUPPLIER
INSERT INTO branch_supplier VALUES(2, 'Hammer Mill', 'Paper');
INSERT INTO branch_supplier VALUES(2, 'Uni-ball', 'Writing Utensils');
INSERT INTO branch_supplier VALUES(3, 'Patriot Paper', 'Paper');
INSERT INTO branch_supplier VALUES(2, 'J.T. Forms & Labels', 'Custom Forms');
INSERT INTO branch_supplier VALUES(3, 'Uni-ball', 'Writing Utensils');
INSERT INTO branch_supplier VALUES(3, 'Hammer Mill', 'Paper');
INSERT INTO branch_supplier VALUES(3, 'Stamford Lables', 'Custom Forms');
-- CLIENT
INSERT INTO client VALUES(400, 'Dunmore Highschool', 2);
INSERT INTO client VALUES(401, 'Lackawana Country', 2);
INSERT INTO client VALUES(402, 'FedEx', 3);
INSERT INTO client VALUES(403, 'John Daly Law, LLC', 3);
INSERT INTO client VALUES(404, 'Scranton Whitepages', 2);
INSERT INTO client VALUES(405, 'Times Newspaper', 3);
INSERT INTO client VALUES(406, 'FedEx', 2);
-- WORKS_WITH
INSERT INTO works_with VALUES(105, 400, 55000);
INSERT INTO works_with VALUES(102, 401, 267000);
INSERT INTO works_with VALUES(108, 402, 22500);
INSERT INTO works_with VALUES(107, 403, 5000);
INSERT INTO works_with VALUES(108, 403, 12000);
INSERT INTO works_with VALUES(105, 404, 33000);
INSERT INTO works_with VALUES(107, 405, 26000);
INSERT INTO works_with VALUES(102, 406, 15000);
INSERT INTO works_with VALUES(105, 406, 130000);
```


https://www.giraffeacademy.com/databases/sql/creating-company-database/