

부스트코스 백엔드

▼ SQL & JDBC 프로그래밍

▼ MySQL

- 데이터베이스 (Database)

정의: 데이터의 집합

→ 여러 프로그램들의 통합된 정보들을 저장하여 운영할 수 있는 공용 데이터의 집합

특성:

- 실시간 접근성
- 지속적인 변화
- 동시 공유성
- 내용 참조

- 데이터베이스 매니지먼트 시스템 (DBMS)

정의: 데이터베이스를 관리하는 소프트웨어

기능:

- 정의기능
- 조작기능 (검색, 삭제, 갱신, 삽입, 삭제)
- 제어기능 (정확성, 안전성)

장점:

- 데이터 중복 최소화
- 데이터의 일관성 및 무결성 유지
- 데이터 보안 보장

단점:

- 운영비가 비쌈
- 백업 및 복구에 대한 관리가 복잡
- 부분적 데이터베이스 손실이 전체 시스템을 정지

▼ SQL

- SQL

정의: 데이터를 보다 쉽게 검색하고 추가, 삭제, 수정 등의 조작을 할 수 있도록 고안된 컴퓨터 언어

- DML

데이터 조작

- DDL

데이터베이스의 스키마 정의 및 조작

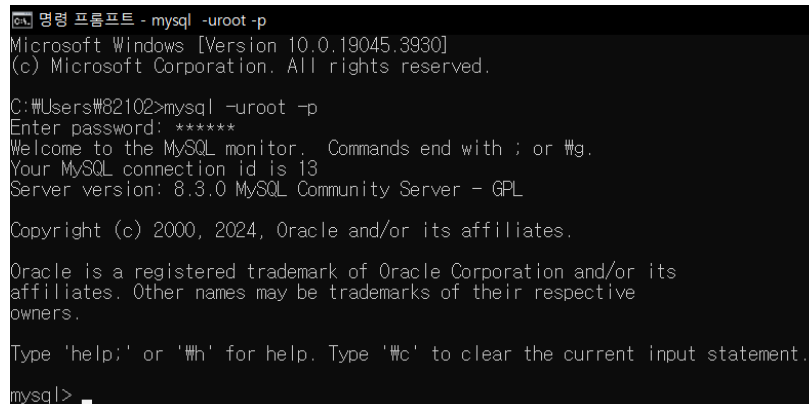
- DCL

데이터를 제어하는 언어 → 권한 관리, 데이터 보안, 무결성 등 정의

- 데이터베이스 생성

- mysql -uroot -p

MySQL 관리자 계정인 root로 데이터베이스 관리 시스템에 접속



```
명령 프롬프트 - mysql -uroot -p
Microsoft Windows [Version 10.0.19045.3930]
(c) Microsoft Corporation. All rights reserved.

C:\Users\#82102>mysql -uroot -p
Enter password: *****
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 13
Server version: 8.3.0 MySQL Community Server - GPL

Copyright (c) 2000, 2024, Oracle and/or its affiliates.

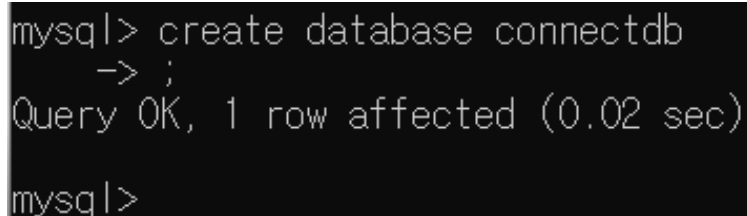
Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

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

mysql>
```

- mysql> create database DB이름;

DB이름으로 데이터베이스 생성



```
mysql> create database connectdb
-> ;
Query OK, 1 row affected (0.02 sec)

mysql>
```

- 데이터베이스 사용자 생성과 권한 주기

```
grant all privileges on db이름.* to 계정이름@'%' identified by ' 암호';  
grant all privileges on db이름.* to 계정이름@'localhost' identified by ' 암호';  
flush privileges;
```

```
grant all privileges on connectdb.* to connectuser@'%' identified by 'connect123!@#';  
grant all privileges on connectdb.* to connectuser@'localhost' identified by 'connect123!@#';  
flush privileges;
```

- *mysql 버전 8에서는 아래와 같이 변경

```
mysql -u root -p
```

위와 같이 root 계정으로 접속을 합니다. 암호는 설치시 입력한 암호를 사용합니다.

```
CREATE DATABASE connectdb;
```

```
CREATE USER connectuser@localhost IDENTIFIED BY 'connect123!@#';
```

```
GRANT ALL PRIVILEGES ON connectdb.* TO 'connectuser'@'localhost';
```

```
FLUSH PRIVILEGES;
```

- 생성한 데이터베이스에 접속하기

- mysql -h호스트명 -uDB계정명 -p 데이터베이스이름

```
C:\Users\#82102>mysql -h127.0.0.1 -uconnectuser -p connectdb  
Enter password: *****  
Welcome to the MySQL monitor. Commands end with ; or \g.  
Your MySQL connection id is 14  
Server version: 8.3.0 MySQL Community Server - GPL  
  
Copyright (c) 2000, 2024, Oracle and/or its affiliates.  
  
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> _
```

- MySQL 연결 끊기

프롬프트에서 quit 혹은 exit 입력

```
mysql> exit
Bye
```

- MySQL 버전과 현재 날짜 구하기

```
mysql> SELECT VERSION(), CURRENT_DATE;
+-----+-----+
| VERSION() | CURRENT_DATE |
+-----+-----+
| 8.3.0      | 2024-01-30    |
+-----+-----+
1 row in set (0.00 sec)

mysql> _
```

** 키워드는 대소문자를 구별하지 않음

ex) select version() = SeLeCt VeRsIoN()

- SQL 입력 도중 취소 가능

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

- DBMS에 존재하는 데이터베이스 확인하기

```
mysql> show databases;
+-----+
| Database |
+-----+
| connectdb |
| information_schema |
| performance_schema |
+-----+
3 rows in set (0.00 sec)

mysql>
```

- 사용중인 데이터베이스 전환하기

“use” command 사용 → use 데이터베이스명

```
C:\Users\#82102>mysql -uroot -p
Enter password: *****
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 20
Server version: 8.3.0 MySQL Community Server - GPL

Copyright (c) 2000, 2024, Oracle and/or its affiliates.

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> use connectdb
Database changed
mysql>
```

▼ SQL2

- 테이블(table)
 - RDBMS의 기본적 저장구조
 - 한 개 이상의 column과 0개 이상의 row로 구성
- 테이블(table)의 구성요소
 - 열(Column)
 - 테이블 상에서의 단일 종류의 데이터를 나타냄
 - 특정 데이터 타입 및 크기를 가지고 있음
 - 행(Row)
 - Column들의 값의 조합
 - 레코드라고 불림
 - 기본키에 의해 구분되며 기본키는 중복을 허용하지 않으며 없어서는 안 됨

- Field

- Row와 Column의 교차점
- 데이터를 포함할 수 있고 데이터가 없을 때는 NULL 값을 가짐

- 데이터베이스의 테이블 확인하기

“show tables;”

- 테이블 생성 및 값의 저장

터미널에서 examples.sql이 있는 폴더로 이동한 후, 다음과 같이 명령을 수행

```
mysql -uconnectuser -p connectdb < examples.sql
```

```
mysql> show tables;
+-----+
| Tables_in_connectdb |
+-----+
| bonus                |
| department           |
| employee             |
| project              |
| project_participation |
| role                 |
| salarygrade          |
+-----+
7 rows in set (0.00 sec)

mysql>
```

- 테이블 구조 확인

ex) EMPLOYEE의 구조 확인

```
mysql> desc EMPLOYEE;
```

▼ DML

- 데이터 조작어의 종류

- SELECT - 검색

SELECT(DISTINCT) 컬럼명(ALIAS)
FROM 테이블명;

SELECT	검색하고자 하는 데이터(컬럼)를 나열 한다
DISTINCT	중복행을 제거
ALIAS	나타날 컬럼에 대한 다른 이름 부여
FROM	선택한 컬럼이 있는 테이블을 명시한다.

** SELECT 뒤에 *를 붙이면 전체 데이터 검색 가능

- 특정 컬럼 검색

ex) employee 테이블에서 직원의 사번(empno), 이름(name), 직업(job)을 출력

```
select empno, name, job from employee;
```

```
mysql> select empno, name, job from employee;
+-----+-----+-----+
| empno | name  | job      |
+-----+-----+-----+
| 7369   | SMITH | CLERK    |
| 7499   | ALLEN | SALESMAN |
| 7521   | WARD  | SALESMAN |
| 7566   | JONES | MANAGER  |
| 7654   | MARTIN | SALESMAN |
| 7698   | BLAKE | MANAGER  |
| 7782   | CLARK | MANAGER  |
| 7788   | SCOTT | ANALYST  |
| 7839   | KING  | PRESIDENT |
| 7844   | TURNER | SALESMAN |
| 7876   | ADAMS | CLERK    |
| 7900   | JAMES | CLERK    |
| 7902   | FORD  | ANALYST  |
| 7934   | MILLER | CLERK    |
+-----+-----+-----+
14 rows in set (0.00 sec)

mysql>
```

- 컬럼에 Alias(별칭) 부여

```
select empno as 사번, name as 이름, job as 직업 from employee;
```

```
mysql> select empno as 사번, name as 이름, job as 직업 from employee;
```

사번	이름	직업
7369	SMITH	CLERK
7499	ALLEN	SALESMAN
7521	WARD	SALESMAN
7566	JONES	MANAGER
7654	MARTIN	SALESMAN
7698	BLAKE	MANAGER
7782	CLARK	MANAGER
7788	SCOTT	ANALYST
7839	KING	PRESIDENT
7844	TURNER	SALESMAN
7876	ADAMS	CLERK
7900	JAMES	CLERK
7902	FORD	ANALYST
7934	MILLER	CLERK

```
14 rows in set (0.00 sec)

mysql>
```

■ 컬럼의 합성(Concatenation)

문자열 결합함수 concat 사용

```
select concat(empno, '-', deptno) as '사번-부서번호' from employee;
```

```
mysql> select concat(empno, '-', deptno) as '사번-부서번호' from employee;
```

사번-부서번호
7782-10
7839-10
7934-10
7369-20
7566-20
7788-20
7876-20
7902-20
7499-30
7521-30
7654-30
7698-30
7844-30
7900-30

```
14 rows in set (0.01 sec)

mysql>
```

■ 중복행의 제거

ex) 부서번호 중복 제거하고 출력

```
select distinct deptno from employee;
```



```
mysql> select deptno from employee;
+-----+
| deptno |
+-----+
|      10 |
|      10 |
|      10 |
|      20 |
|      20 |
|      20 |
|      20 |
|      20 |
|      30 |
|      30 |
|      30 |
|      30 |
|      30 |
|      30 |
+-----+
14 rows in set (0.00 sec)

mysql> select distinct deptno from employee;
+-----+
| deptno |
+-----+
|      10 |
|      20 |
|      30 |
+-----+
```

- 정렬하기

ORDER BY 절

```
SELECT(DISTINCT) 컬럼명(ALIAS)
FROM 테이블명
ORDER BY 컬럼이나 표현식 (ASC 또는 DESC);
```

ASC	오름차순 정렬, 기본값
DESC	내림차순

ex) 이름을 기준으로 오름차순 정렬

ex) 사번을 기준으로 내림차순 정렬

```
select empno, name, job from employee order by name;
```

```
select empno as 사번, name as 이름, job as 직업 from employee order by
사번 desc;
```

```
mysql> select empno as 사번, name as 이름, job as 직업 from employee order by name;
+-----+-----+-----+
| 사번 | 이름 | 직업 |
+-----+-----+-----+
| 7876 | ADAMS | CLERK |
| 7499 | ALLEN | SALESMAN |
| 7698 | BLAKE | MANAGER |
| 7782 | CLARK | MANAGER |
| 7902 | FORD | ANALYST |
| 7900 | JAMES | CLERK |
| 7566 | JONES | MANAGER |
| 7839 | KING | PRESIDENT |
| 7654 | MARTIN | SALESMAN |
| 7934 | MILLER | CLERK |
| 7788 | SCOTT | ANALYST |
| 7369 | SMITH | CLERK |
| 7844 | TURNER | SALESMAN |
| 7521 | WARD | SALESMAN |
+-----+-----+-----+
14 rows in set (0.00 sec)

mysql> select empno as 사번, name as 이름, job as 직업 from employee order by 사번 desc;
+-----+-----+-----+
| 사번 | 이름 | 직업 |
+-----+-----+-----+
| 7934 | MILLER | CLERK |
| 7902 | FORD | ANALYST |
| 7900 | JAMES | CLERK |
| 7876 | ADAMS | CLERK |
| 7844 | TURNER | SALESMAN |
| 7839 | KING | PRESIDENT |
| 7788 | SCOTT | ANALYST |
| 7782 | CLARK | MANAGER |
| 7698 | BLAKE | MANAGER |
| 7654 | MARTIN | SALESMAN |
| 7566 | JONES | MANAGER |
| 7521 | WARD | SALESMAN |
| 7499 | ALLEN | SALESMAN |
| 7369 | SMITH | CLERK |
+-----+-----+-----+
14 rows in set (0.00 sec)
```

■ 특정 행 검색

SELECT(DISTINCT) 컬럼명(ALIAS)
FROM 테이블명
WHERE 조건식
ORDER BY 컬럼이나 표현식 (ASC 또는 DESC)

조건식	컬럼이름이나 표현식의 상수, 연산자로 구성
-----	-------------------------

WHERE 형식 연산자 값

ex)
WHERE title = 'Staff'
WHERE salary BETWEEN 1000 AND 2000

• 논리연산자

- IN 키워드
- LIKE 키워드
- 와일드 카드를 사용하여 특정 문자를 포함한 값에 대한 조건 처리
- %는 0부터 여러 개의 문자열을 나타냄
- _는 단 하나의 문자를 나타내는 와일드 카드
- 함수의 사용
 - lower, lcase
 - substring

“select substring(문자열, 시작 인덱스, 끊을 문자 개수)”
 - lpad, rpad

문자열의 공백을 특정 문자로 채워서 출력

lpad는 왼쪽 공백, rpad는 오른쪽 공백

“select lpad(문자열, 출력할 문자 개수, 공백을 채울 문자)”
 - trim, ltrim, rtrim
 - ABS(x) : x의 절댓값 출력
 - MOD(n,m) : n을 m으로 나눈 나머지 값 출력
- INSERT - 등록
- UPDATE - 수정
- DELETE - 삭제