W02. MySQL Practice

 # Create Database

- Error 'No database selected' appears unless you create database at the first place.

```
CREATE database w02;
USE w02;
```

 # Remove Database

```
drop database w02;
```

 # DDL: Creating Relations in SQL

```
CREATE TABLE Students
        (sid CHAR(20),
         sname CHAR(20),
         age INTEGER,
         gpa REAL);

CREATE TABLE Enrolled
        (sid CHAR(20),
         cid CHAR(20),
         grade CHAR(2));
```

 # DML: Modifying Relations in SQL

```
INSERT INTO Students (sid, sname, age, gpa) VALUES (101, 'Alice', 21, 3.2);
```

[CAUTION]
Error occurs if you use italic type for single quotation marks(' ').

```
mysql>  INSERT INTO Students (sid, sname, age, gpa)
    ->  VALUES (101, 'Alice' 21, 3.2);
ERROR 1054 (42S22): Unknown column ' 'Alice' ' in 'field list'
mysql> INSERT INTO Students (sid, sname, age, gpa)
    ->  VALUES (101, 'Alice', 21, 3.2);
Query OK, 1 row affected (0.09 sec)
```

```
DELETE FROM Students
WHERE sname='Alice';
```

[CAUTION]
Error occurs if you write as below.

```
mysql> delete from students s where s.sname='Alice';
ERROR 1064 (42000): You have an error in your SQL syntax; check the manual
 that corresponds to your MySQL server version for the right syntax to use
 near 's where s.sname='Alice'' at line 1
```

```
UPDATE Students
SET gpa = gpa - 0.1
WHERE gpa >= 3.3;
```

```
DROP TABLE Enrolled;
DROP TABLE Students;
```

 # Retrieval Queries in SQL

```
create table Courses (
  cid CHAR(20) primary key,
  cname varchar(32),
  credit integer,
  dept varchar(8)
);

insert into Courses (cid, cname, credit, dept) values ('101', 'C_programming', 3, 'CS');
insert into Courses (cid, cname, credit, dept) values ('102', 'Discrete_Math', 2, 'Math');
insert into Courses (cid, cname, credit, dept) values ('301', 'Databases', 4, 'CS');
insert into Courses (cid, cname, credit, dept) values ('302', 'Artificial_Intelligence', 3, 'CS');
insert into Courses (cid, cname, credit, dept) values ('405', 'Data_Mining', 3, 'CS');
```

```
SELECT  *
FROM  Courses
WHERE  credit = 3;
```

[RESULT]

```
+-----+-------------------------+--------+------+
| cid | cname                   | credit | dept |
+-----+-------------------------+--------+------+
| 101 | C_programming           |      3 | CS   |
| 302 | Artificial_Intelligence |      3 | CS   |
| 405 | Data_Mining             |      3 | CS   |
+-----+-------------------------+--------+------+
3 rows in set (0.02 sec)
```

```
SELECT  cname
FROM  Courses
WHERE  dept = 'CS';
```

[RESULT]

```
+-------------------------+
| cname                   |
+-------------------------+
| C_programming           |
| Databases               |
| Artificial_Intelligence |
| Data_Mining             |
+-------------------------+
4 rows in set (0.00 sec)
```

 # Retrieval Queries in SQL

```
SELECT  cid, cname
FROM  Courses
WHERE  credit > 2 and dept = 'CS';
```

[RESULT]

```
+-----+-------------------------+
| cid | cname                   |
+-----+-------------------------+
| 101 | C_programming           |
| 301 | Databases               |
| 302 | Artificial_Intelligence |
| 405 | Data_Mining             |
+-----+-------------------------+
4 rows in set (0.00 sec)
```

```
SELECT  DISTINCT credit
FROM  Courses
WHERE  dept = 'CS';
```

[RESULT]

```
+--------+
| credit |
+--------+
|      3 |
|      4 |
+--------+
2 rows in set (0.00 sec)
```

```
DROP TABLE Courses;
```

 # Primary and Candidate Keys in SQL

```
CREATE TABLE Enrolled
        (sid CHAR(20),
         cid CHAR(20),
         grade CHAR(2),
         PRIMARY KEY(sid, cid));
```

```
DROP TABLE Enrolled;
```

```
CREATE TABLE Enrolled
        (sid CHAR(20),
         cid CHAR(20),
         grade CHAR(2),
         PRIMARY KEY(sid),
         UNIQUE (cid, grade));
```

```
DROP TABLE Enrolled;
```

 # UNIQUE and NOT NULL

```
CREATE TABLE Emp (
  idx int,
  no int NULL,
  name varchar(32),
  PRIMARY KEY (idx),
  UNIQUE (no));

INSERT INTO Emp (idx, no, name) VALUES (1, 100, 'Kee');
INSERT INTO Emp (idx, no, name) VALUES (2, NULL, 'Lee');
INSERT INTO Emp (idx, no, name) VALUES (3, NULL, 'Yoo');
```

 # CHECK Constraints

```
CREATE TABLE Students
        (sid CHAR(20),
         sname CHAR(20),
         age INTEGER,
         gpa REAL);
```

```
DROP TABLE Students;
```

```
CREATE TABLE Students
        (sid CHAR(20) NOT NULL,
         sname CHAR(20) NOT NULL,
         age INTEGER CHECK (age>=18),
         gpa REAL  CHECK (gpa >= 0.0 ),
        PRIMARY KEY (sid));
```

```
DROP TABLE Students;
```

 # Foreign Keys in SQL

```
CREATE TABLE Courses (
  cid varchar(20) primary key,
  cname varchar(32),
  credit integer,
  dept varchar(8)
);
```

```
CREATE TABLE Students
        (sid VARCHAR(20) NOT NULL,
         sname CHAR(20) NOT NULL,
         age INTEGER CHECK (age>=18),
         gpa REAL  CHECK (gpa >= 0.0 ),
        PRIMARY KEY (sid));
```

```
CREATE TABLE Enrolled
   (sid CHAR(20),  cid CHAR(20),  grade CHAR(2),
     PRIMARY KEY  (sid,cid),
     FOREIGN KEY (sid) REFERENCES Students(sid),
     FOREIGN KEY (cid) REFERENCES Courses(cid));
```

```
DROP TABLE Enrolled;
DROP TABLE Students;
DROP TABLE Courses;
```

 # Referential Integrity in SQL

```
CREATE TABLE Students
        (sid CHAR(20) NOT NULL,
         sname CHAR(20) NOT NULL,
         age INTEGER CHECK (age>=18),
         gpa REAL  CHECK (gpa >= 0.0 ),
        PRIMARY KEY (sid));
```

```
CREATE TABLE Enrolled
   (sid CHAR(20),
    cid CHAR(20),
    grade CHAR(2),
    PRIMARY KEY  (sid,cid),
    FOREIGN KEY (sid) REFERENCES Students(sid)
         ON DELETE CASCADE
         ON UPDATE CASCADE);
```

```
DROP TABLE Enrolled;
DROP TABLE Students;
```

 # Views and Base Tables

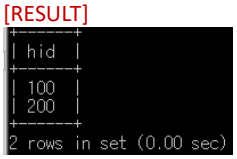
```
CREATE TABLE Students
        (sid CHAR(20),
         sname CHAR(20),
         age INTEGER
         );

insert into Students (sid, sname, age) values ('100', 'Alice', 22);
insert into Students (sid, sname, age) values ('200', 'Bob', 23);
insert into Students (sid, sname, age) values ('300', 'David', 15);
insert into Students (sid, sname, age) values ('400', 'Eva', 12);
```

```
CREATE  VIEW  HighStudents (hid, name)
        AS  SELECT   S.sid, S.sname
               FROM  Students S
               WHERE  S.age > 21;
```

 # Queries on Views

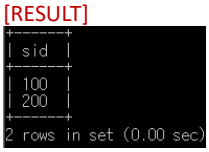
```
SELECT   S.hid
 FROM  HighStudents S;
```

[RESULT]




```
SELECT   S.sid
 FROM  Students S
 WHERE  S.age > 21;
```

[RESULT]




```
DROP TABLE Students;
DROP VIEW HighStudents;
```