

과제 3 레포트

(1) MySQL의 대화식 SQL 도구를 이용하여 자신의 설계과제의 응용에서 필요하다고 예상되는 테이블 1 개를 선택하여 create하시오.

한정판 거래 플랫폼 DB에서 회원정보를 담은 테이블을 선택하여서 ID를 primary key로 하여 회원정보를 담은 테이블을 Create하였습니다.

```
CREATE TABLE `SOLDOUT`.`account_info` (
  `ID` VARCHAR(10) NOT NULL,
  `email` VARCHAR(20) NULL,
  `password` VARCHAR(20) NULL,
  `nickname` VARCHAR(15) NULL,
  `tel` VARCHAR(15) NULL,
  `point` INT NULL,
  `membership_grade` VARCHAR(15) NULL,
  PRIMARY KEY (`ID`));
```

(2) MySQL의 대화식 SQL 도구를 이용하여, R에 튜플을 5개 내외 삽입한 후 튜플들이 잘 삽입되었는지 select 문으로 확인하시오.

총 6개의 input을 진행하였습니다. 그 후 select * from account_info를 실행하여서 workbench를 이용하여 잘 삽입됨을 확인할 수 있습니다.

The screenshot displays the MySQL Workbench interface. On the left, the 'SCHEMAS' pane shows the 'assignment2_Database' with a table named 'account_info' under the 'SOLDOUT' schema. The main area shows the 'Query 1' window with the SQL query: `select * from account_info`. Below the query, the 'Result Grid' shows the data for the 'account_info' table. The table has columns: ID, email, password, nickname, tel, point, and membership_grade. The data rows are as follows:

ID	email	password	nickname	tel	point	membership_grade
bobbyco	bobbyci101@naver.com	1332243333	bobbyck	010-1111-3905	1000	bronze
donguk242	donguk2000@naver.com	131324333	ukpeace	010-2320-3405	0	bronze
dukkk	dukkk10@naver.com	132445333	duckduck	010-1321-4445	5000	gold
jin01020	jun01020@naver.com	1234444	sjking	010-2304-2211	0	bronze
mgmg0612	mgmg0612@naver.com	123456789	realishoman	010-1234-5678	3000	bronze
sven777	sven770@naver.com	12344133	heheka	010-2322-2101	10000	silver
NULL	NULL	NULL	NULL	NULL	NULL	NULL

Below the result grid, the 'Action Output' pane shows the execution history of the queries. The queries are as follows:

Time	Action	Response	Duration / Fetch Time
23:24:21	CREATE TABLE `SOLDOUT`.`account_info` (`ID` VARCHAR(10) NOT NULL,...	0 row(s) affected	0.036 sec
23:28:27	insert into account_info values("mgmg0612","mgmg0612@naver.com","123456...	1 row(s) affected	0.012 sec
23:28:52	select * from account_info LIMIT 0, 1000	1 row(s) returned	0.00046 sec / 0.000...
23:31:51	insert into account_info values("jin01020","jun01020@naver.com","1234444","s...	1 row(s) affected	0.0063 sec
23:32:44	insert into account_info values("sven77","sven770@naver.com","12344133","he...	1 row(s) affected	0.00077 sec
23:37:23	insert into account_info values("donguk242","donguk2000@naver.com","13132...	1 row(s) affected	0.0058 sec
23:38:09	insert into account_info values("bobbyco","bobbyci101@naver.com","13322433...	1 row(s) affected	0.0013 sec
23:38:47	insert into account_info values("dukkk","dukkk10@naver.com","132445333","du...	1 row(s) affected	0.0010 sec
23:39:14	select * from account_info LIMIT 0, 1000	6 row(s) returned	0.00045 sec / 0.000...

Handwritten red text in the image includes: '결과값입니다.' (This is the result) and '6개의 tuple query로 입력하였습니다.' (I entered it with 6 tuple queries). Another red text note says: '그 후 select * from account_info를 실행하였습니다.' (Then I executed select * from account_info).

(3) (1)과 (2)의 연산이 수행된 후, R이 MySQL의 DB에 그대로 저장되어 있다고 하자. MySQL의 이 DB에 connect하여 아래 작업을 수행하는 Java/JDBC 프로그램을 작성하시오.

- Select * from account_info를 쿼리문으로 하여 (2)에서 입력한 값을 JDBC에서 출력할 수 있도록 아래와 같이 코드를 만들었습니다. 이 코드에 resultset 데이터를 변수로 넣으면 select * from account_info를 실행하여 출력해주는 printaccountinfo를 만들었습니다. 이를 통해 JDBC와 연결이 잘 되어있는지 확인할 수 있었습니다.

```
DBC_EX.java X
public class JDBC_EX {
    static final String DB_URL= "jdbc:mysql://localhost/soldout?useUnicode=true&useJDBCCompliantTimezoneShift=true&useLegacyDat
    static final String USER = "root";
    static final String PASS = "12345678";
    static final String QUERY = "select * from account_info";

    public static void printaccountinfo(ResultSet r) throws SQLException {
        int accountNum = 1;
        while(r.next()) {
            System.out.println("account" + accountNum + "\n" + "ID = " + r.getString("ID") + " email = " + r.getString("email") +
            accountNum++;
        }
    }

    public static void main(String[] args) {
        Connection conn = null;
        Statement stmt = null;
        ResultSet rs = null;
        PreparedStatement pstmt = null;
        ResultSet aftrs = null;
        try {
            conn = DriverManager.getConnection(DB_URL,USER,PASS);
            stmt = conn.createStatement();
            rs = stmt.executeQuery(QUERY);
            printaccountinfo(rs);
        }
    }
}
```

-또한 동적 구문으로 tuple을 하나 추가하기 위해 아래와 같이 코드를 짰습니다. 그 후 잘 입력됐는지 확인하기 먼저 아래와 같이 코드를 작성하여 동적으로 tuple을 추가했습니다.

```
String sql = "insert into account_info values (?,?,?,?,?,?,?)";
pstmt = conn.prepareStatement(sql);
pstmt.setString(1,"butcherjo");
pstmt.setString(2,"butche20@naver.com");
pstmt.setString(3,"12341201@");
pstmt.setString(4,"buchern");
pstmt.setString(5,"010-9323-1392");
pstmt.setInt(6,1000);
pstmt.setString(7,"bronze");
pstmt.executeUpdate();
```

그 후 select * from account_info query를 실행한 resultset을 매개변수로 넣은 printaccountinfo함수를 실행하였습니다.

아래는 결과값입니다. 동적으로 넣은 tuple(Id = butcherjo)가 잘 입력됨을 확인할 수 있습니다.

```
terminated> JDBC_EX [Java Application] /Library/Java/JavaVirtualMachines/jdk-17.0.5.jdk/Contents/Home/bin/java (2022. 11. 11. 오전 12:55:15 - 오전 12:55:15) [pid: 20993]
account1 동적 쿼리로 insert 전 select * from account_info 출력 구문입니다.
ID = bobbyco email = bobbyci101@naver.com password = 1332243333 ninckname = bobbyck tel = 010-1111-3305 point = 1000 membership_grade = bronze
account2
ID = donguk242 email = donguk2000@naver.com password = 1313243333 ninckname = ukpeace tel = 010-2320-3405 point = 0 membership_grade = bronze
account3
ID = dukkk email = dukkk10@naver.com password = 1324453333 ninckname = duckduck tel = 010-1321-4445 point = 5000 membership_grade = gold
account4
ID = jin01020 email = jun01020@naver.com password = 1234444 ninckname = sjking tel = 010-2304-2211 point = 0 membership_grade = bronze
account5
ID = mgmg0612 email = mgmg0612@naver.com password = 123456789 ninckname = realissoman tel = 010-1234-5678 point = 3000 membership_grade = bronze
account6
ID = sven77 email = sven770@naver.com password = 12344133 ninckname = heheka tel = 010-2322-2101 point = 10000 membership_grade = silver
after input
동적 쿼리로 insert 후 select * from account_info 출력 구문입니다.
account1
ID = bobbyco email = bobbyci101@naver.com password = 1332243333 ninckname = bobbyck tel = 010-1111-3305 point = 1000 membership_grade = bronze
account2
ID = butcherjo email = butche20@naver.com password = 12341201@ ninckname = buchern tel = 010-9323-1392 point = 1000 membership_grade = bronze
account3 동적으로 추가 됨을 확인할 수 있습니다.
ID = donguk242 email = donguk2000@naver.com password = 1313243333 ninckname = ukpeace tel = 010-2320-3405 point = 0 membership_grade = bronze
account4
ID = dukkk email = dukkk10@naver.com password = 1324453333 ninckname = duckduck tel = 010-1321-4445 point = 5000 membership_grade = gold
account5
ID = jin01020 email = jun01020@naver.com password = 1234444 ninckname = sjking tel = 010-2304-2211 point = 0 membership_grade = bronze
account6
ID = mgmg0612 email = mgmg0612@naver.com password = 123456789 ninckname = realissoman tel = 010-1234-5678 point = 3000 membership_grade = bronze
account7
ID = sven77 email = sven770@naver.com password = 12344133 ninckname = heheka tel = 010-2322-2101 point = 10000 membership_grade = silver
```