

데이터베이스 설계과제 1

20222663 권기영

1. Create table문

```
1 • CREATE TABLE `takes` (  
2   `ID` varchar(5) NOT NULL,  
3   `course_id` varchar(8) NOT NULL,  
4   `sec_id` varchar(8) NOT NULL,  
5   `semester` varchar(6) NOT NULL,  
6   `year` decimal(4,0) NOT NULL,  
7   `grade` varchar(2) DEFAULT NULL,  
8   PRIMARY KEY (`ID`,`course_id`,`sec_id`,`semester`,`year`),  
9   CONSTRAINT `ID` FOREIGN KEY (`ID`) REFERENCES `student` (`ID`)  
10  );  
11  
12 • CREATE TABLE `student` (  
13   `ID` varchar(5) NOT NULL,  
14   `name` varchar(20) NOT NULL,  
15   `dept_name` varchar(20) DEFAULT NULL,  
16   `tot_cred` decimal(3,0) DEFAULT NULL,  
17   PRIMARY KEY (`ID`)  
18  );  
19  
20 • CREATE TABLE `course` (  
21   `course_id` varchar(8) NOT NULL,  
22   `title` varchar(50) DEFAULT NULL,  
23   `dept_name` varchar(20) DEFAULT NULL,  
24   `credits` decimal(2,0) DEFAULT NULL,  
25   PRIMARY KEY (`course_id`)  
26  );
```

위의 코드와 같이, 총 3개의 테이블(takes, student, course)을 만들었고, PK 설정과 FK 설정을 해주었다.

```
1 • insert into course values ('CSE-123', 'Data Structure', 'CSE', '3');  
2 • insert into course values ('EE-321', 'Logic Curcuit', 'EE', '3');  
3 • insert into course values ('SAT-456', 'Elementary Statistics', 'Statistics', '2');  
4  
5 • insert into student values ('12345', 'Minseok', 'CSE', '8');  
6 • insert into student values ('13579', 'Hoon', 'Statistics', '5');  
7 • insert into student values ('13579', 'Hoon', 'Statistics', '5');  
8 • insert into student values ('97531', 'Suhyeon', 'CSE', '3');  
9  
10 • insert into takes values ('12345', 'CSE-123', '1', 'Spring', '2023', 'A+');  
11 • insert into takes values ('12345', 'EE-321', '1', 'Fall', '2021', 'C+');  
12 • insert into takes values ('12345', 'SAT-456', '2', 'Fall', '2022', 'B+');  
13 • insert into takes values ('13579', 'CSE-123', '2', 'Spring', '2023', 'A+');  
14 • insert into takes values ('13579', 'SAT-456', '1', 'Spring', '2022', 'B+');  
15 • insert into takes values ('54321', 'EE-321', '2', 'Fall', '2018', 'A+');  
16 • insert into takes values ('97531', 'CSE-123', '1', 'Fall', '2023', 'A+');
```

만든 3개의 테이블에 대해서, course 테이블에는 총 3개의 튜플을, student 테이블에는 총 4개의 튜플을, takes 테이블에는 총 7개의 튜플을 삽입하였다.

2. SQL문

```
static final String QUERY1 = "SELECT * FROM student WHERE ID = ?";
static final String QUERY2 = "SELECT T.course_id, C.title, T.sec_id, T.semester, T.year, T.grade " +
    "FROM course as C, takes as T WHERE ID = ? and T.course_id = C.course_id";
```

QUERY1은 어떤 ID에 대해서, 그 ID인 student 테이블의 튜플 전체를 선택하는 질의문이고, QUERY2는 어떤 ID에 대해서, 그 ID를 가진 학생이 수강한 과목의 과목번호, 과목명, 분반번호, 년도, 학기, 성적 리스트를 선택하는 질의문이다.

3. 프로그램 로직

- 1) 2개의 쿼리 문을 선언한다. QUERY1은 입력 받은 ID와 맞는 학생의 학생정보를 SELECT 하는 쿼리 문이고, QUERY2는 입력받은 ID와 맞는 학생의 수강과목들의 정보를 SELECT 하는 쿼리문이다.
- 2) while 문으로 계속 학번(ID)을 입력 받고, 만약 사용자가 아무것도 입력하지 않고 그냥 엔터를 누르면 프로그램이 종료된다.
- 3) 두 개의 쿼리에 입력 값인 input을 넣는다.
- 4) 두 개의 쿼리에 대해서 결과셋을 출력한다. 결과셋은 re.next()가 있을 때까지 출력한다. 만약 첫 번째 쿼리문에서 input의 ID를 가진 학생이 없다면, input 학번을 가진 학생이 없다는 출력만 나오고, input ID를 가진 학생이 있다면, 그 학생의 학생정보와 수강과목 정보가 모두 나오게 된다.

4. DB 데이터

<course> 테이블의 DB 데이터이다.

	course_id	title	dept_name	credits
▶	CSE-123	Data Structure	CSE	3
	EE-321	Logic Circuit	EE	3
	SAT-456	Elementary Statistics	Statistics	2
*	NULL	NULL	NULL	NULL

<student> 테이블의 DB 데이터이다.

	ID	name	dept_name	tot_cred
▶	12345	Minseok	CSE	8
	13579	Hoon	Statistics	5
	54321	Minsik	EE	3
	97531	Suhyeon	CSE	3
✱	NULL	NULL	NULL	NULL

<takes> 테이블의 DB 데이터이다.

	ID	course_id	sec_id	semester	year	grade
▶	12345	CSE-123	1	Spring	2023	A+
	12345	EE-321	1	Fall	2021	C+
	12345	SAT-456	2	Fall	2022	B+
	13579	CSE-123	2	Spring	2023	A+
	13579	SAT-456	1	Spring	2022	B+
	54321	EE-321	2	Fall	2018	A+
	97531	CSE-123	1	Fall	2023	A+
✱	NULL	NULL	NULL	NULL	NULL	NULL

5. Java 프로그램 실행 결과

```

학번(ID) >> 12345
<학생정보>
학번 : 12345, 이름 : Minseok, 학과: CSE, 총 수강학점 : 8
<수강과목 정보>
과목번호 : CSE-123, 과목명 : Data Structure, 분반번호 : 1, 년도 : 2023, 학기 : Spring, 성적 : A+
과목번호 : EE-321, 과목명 : Logic Curcuit, 분반번호 : 1, 년도 : 2021, 학기 : Fall, 성적 : C+
과목번호 : SAT-456, 과목명 : Elementary Statistics, 분반번호 : 2, 년도 : 2022, 학기 : Fall, 성적 : B+

학번(ID) >> 13579
<학생정보>
학번 : 13579, 이름 : Hoon, 학과: Statistics, 총 수강학점 : 5
<수강과목 정보>
과목번호 : CSE-123, 과목명 : Data Structure, 분반번호 : 2, 년도 : 2023, 학기 : Spring, 성적 : A+
과목번호 : SAT-456, 과목명 : Elementary Statistics, 분반번호 : 1, 년도 : 2022, 학기 : Spring, 성적 : B+

학번(ID) >> 54321
<학생정보>
학번 : 54321, 이름 : Minsik, 학과: EE, 총 수강학점 : 3
<수강과목 정보>
과목번호 : EE-321, 과목명 : Logic Curcuit, 분반번호 : 2, 년도 : 2018, 학기 : Fall, 성적 : A+

학번(ID) >> 97531
<학생정보>
학번 : 97531, 이름 : Suhyeon, 학과: CSE, 총 수강학점 : 3
<수강과목 정보>
과목번호 : CSE-123, 과목명 : Data Structure, 분반번호 : 1, 년도 : 2023, 학기 : Fall, 성적 : A+

학번(ID) >> 13131
13131학번을 가진 학생은 없습니다.

학번(ID) >>
프로그램 종료!!

```

4명의 학생에 대한 학생 정보와 수강과목 정보가 제대로 출력된다. 또한 학생 테이블에 없는 ID를 입력하면 없는 학생이라고 출력되고, 그냥 엔터를 누르면 프로그램이 종료된다.

6. 정확성 검증

<course>

	course_id	title	dept_name	credits
▶	CSE-123	Data Structure	CSE	3
	EE-321	Logic Curcuit	EE	3
	SAT-456	Elementary Statistics	Statistics	2
*	NULL	NULL	NULL	NULL

<student>

	ID	name	dept_name	tot_cred
▶	12345	Minseok	CSE	8
	13579	Hoon	Statistics	5
	54321	Minsik	EE	3
	97531	Suhyeon	CSE	3
*	NULL	NULL	NULL	NULL

<takes>

	ID	course_id	sec_id	semester	year	grade
▶	12345	CSE-123	1	Spring	2023	A+
	12345	EE-321	1	Fall	2021	C+
	12345	SAT-456	2	Fall	2022	B+
	13579	CSE-123	2	Spring	2023	A+
	13579	SAT-456	1	Spring	2022	B+
	54321	EE-321	2	Fall	2018	A+
	97531	CSE-123	1	Fall	2023	A+
*	NULL	NULL	NULL	NULL	NULL	NULL

1) ID = 12345

<출력>

```
학번(ID) >> 12345
<학생정보>
학번 : 12345, 이름 : Minseok, 학과: CSE, 총 수강학점 : 8
<수강과목 정보>
과목번호 : CSE-123, 과목명 : Data Structure, 분반번호 : 1, 년도 : 2023, 학기 : Spring, 성적 : A+
과목번호 : EE-321, 과목명 : Logic Curcuit, 분반번호 : 1, 년도 : 2021, 학기 : Fall, 성적 : C+
과목번호 : SAT-456, 과목명 : Elementary Statistics, 분반번호 : 2, 년도 : 2022, 학기 : Fall, 성적 : B+
```

학생 테이블과 출력을 비교해보면, ID = 12345인 학생의 이름, 학과, 총 수강학점이 제대로 출력되고 있다. 또한 ID = 12345인 학생이 수강하는 과목은 총 3개이고, 수강하는 과목들에 대한 정보도 제대로 출력되고 있는 것을 확인할 수 있다.

2) ID = 13579

<출력>

```
학번(ID) >> 13579
<학생정보>
학번 : 13579, 이름 : Hoon, 학과: Statistics, 총 수강학점 : 5
<수강과목 정보>
과목번호 : CSE-123, 과목명 : Data Structure, 분반번호 : 2, 년도 : 2023, 학기 : Spring, 성적 : A+
과목번호 : SAT-456, 과목명 : Elementary Statistics, 분반번호 : 1, 년도 : 2022, 학기 : Spring, 성적 : B+
```

이 또한 학생 테이블과 출력을 비교해 봤을 때, 학생 정보가 제대로 출력된다. 또한 ID = 13579인 학생이 수강하는 과목은 총 2개이고, 정보가 제대로 출력되는 것을 확인할 수 있다.

7. Java 프로그램 소스코드

```
package assignment1;

import java.sql.*;
import java.util.Scanner;

public class JDBC_print_seclist
{
    static final String DB_URL = "jdbc:mysql://localhost/assignment1_schema?useUnicode=true&useJDBCCompliantTimezoneShift=true&useLegacyDatetimeCode=false&serverTimezone=UTC";
    static final String USER = "root"; // user name
    static final String PASS = "k6385112"; // user password

    static final String QUERY1 = "SELECT * FROM student WHERE ID = ?";
    static final String QUERY2 = "SELECT T.course_id, C.title, T.sec_id, T.semester, T.year, T.grade " +
        "FROM course as C, takes as T WHERE ID = ? and T.course_id = C.course_id";

    public static void main(String[] args)
    {
        Scanner sc = new Scanner(System.in);
        Connection conn = null;
        PreparedStatement pstmt1 = null;
        PreparedStatement pstmt2 = null;

        try {
            conn = DriverManager.getConnection(DB_URL, USER, PASS);
            pstmt1 = conn.prepareStatement(QUERY1);
            pstmt2 = conn.prepareStatement(QUERY2);

            while (true)
            {
                System.out.print("학번(ID) >> ");
                String input = sc.nextLine();

                if (input == "") // exit condition
                {
                    System.out.println("프로그램 종료!!");
                    break;
                }

                pstmt1.setString(1, input); // 학리문의 ? 자리에 입력 넣기
                pstmt2.setString(1, input);

                try (ResultSet rs = pstmt1.executeQuery())
                {
                    if (!rs.next())
                        System.out.println(input + "학번을 가진 학생은 없습니다.");
                    else
                    {
                        System.out.println("<<학생정보>>");
                        System.out.print("학번 : " + rs.getString("ID") + ", ");
                        System.out.print("이름 : " + rs.getString("name") + ", ");
                        System.out.print("학과 : " + rs.getString("dept_name") + ", ");
                        System.out.println("총 수강학점 : " + rs.getString("tot_cred"));
                        System.out.println("<<수강과목 정보>>");
                    }
                }
            }
        }
    }
}
```

```
        }
    }

    try (ResultSet rs = pstmt2.executeQuery())
    {
        while (rs.next())
        {
            System.out.print("과목번호 : " + rs.getString("T.course_id") + ", ");
            System.out.print("과목명 : " + rs.getString("C.title") + ", ");
            System.out.print("분반번호 : " + rs.getString("T.sec_id") + ", ");
            System.out.print("년도 : " + rs.getString("T.year") + ", ");
            System.out.print("학기 : " + rs.getString("T.semester") + ", ");
            System.out.println("성적 : " + rs.getString("T.grade"));
        }
        System.out.println("");
    }
}

} catch (SQLException e) {
    System.out.println("SQLException : " + e);
} finally {
    try {
        pstmt1.close();
        pstmt2.close();
        conn.close();
    } catch (SQLException e) {
        e.printStackTrace();
    }
}
}
```