

Project 2: Implementing a Simple Database Application

Due: 2015/12/11 (Fri), 11:59 PM

이번 프로젝트의 목표는 데이터베이스를 이용하는 간단한 어플리케이션을 설계하고 구현하는 것이다. JDBC를 이용하여 대학교 원서 접수를 시뮬레이션하는 간단한 어플리케이션을 만들어 본다. 이번 프로젝트를 통해 학생들은 어플리케이션과 데이터베이스를 연동하는 방법을 배우게 된다.

1 요구 사항

- 데이터는 관계형 데이터베이스에 저장되어야 한다.
- 각 컬럼의 타입은 2장에 명시된 대로 정의하여야 한다.
- 학교 데이터 및 학생 데이터를 추가/삭제할 수 있어야 한다.
- 학교 데이터 및 학생 데이터를 출력할 수 있어야 한다.
- 학생이 학교에 원서 접수를 할 수 있어야 한다.
- 학교 별로 합격이 예상되는 학생의 명단을 출력할 수 있어야 한다.
- 학생 별로 합격이 예상되는 학교의 명단을 출력할 수 있어야 한다.

2 데이터 타입

1. 학교
 - 학교 ID: 정수
 - 학교명: 문자열 (최대 128자)
 - 정원: 정수 (1 이상의 값)
 - 군: 문자열 ('A', 'B', 'C' 중 1개의 값을 가져야 함)
 - 내신 성적 반영 비율: 실수 (0 이상의 값)
2. 학생
 - 학생 ID: 정수
 - 학생 이름: 문자열 (최대 20자)
 - 수능 성적: 정수 (0 이상 400 이하의 값)
 - 내신 성적: 정수 (0 이상 100 이하의 값)

3 세부 사항

구현한 프로그램은 다음과 같은 기능을 할 수 있어야 한다.

1. 모든 학교 정보 출력
 - 모든 학교의 정보(학교 ID, 이름, 정원, 군, 내신 성적 반영 비율, 지원자 수)를 출력한다.
2. 모든 학생 정보 출력
 - 모든 학생의 정보(학생 ID, 이름, 수능 성적, 내신 성적)를 출력한다.
3. 학교 추가
 - 새로운 학교를 추가한다.
 - 이름(String), 정원(int), 군(char), 내신 성적 반영 비율(float)을 입력 받음
 - 학교 ID는 자동 부여
 - 학교가 없다면 새로운 학교의 ID를 1로 부여
 - 학교가 있다면 새로운 학교의 ID를 기존 학교들 중 최대 ID값 + 1로 부여
 - 정원, 군, 내신 반영 비율이 정해진 범위를 벗어난다면 에러 메시지 출력
 - 이름이 최대 길이 (128자)보다 길다면 128자로 truncate해 추가함
4. 학교 삭제
 - 학교를 삭제한다.
 - 학교 ID(int)를 입력 받아 해당되는 학교 삭제
 - 학교를 삭제하면 관련된 원서 접수 내역도 모두 삭제되어야 함
5. 학생 추가
 - 새로운 학생을 추가한다.
 - 이름(String), 수능 성적(int), 내신 성적(int)을 입력 받음
 - 학생 ID는 자동 부여
 - 학생이 없다면 새로운 학생의 ID를 1로 부여
 - 학생이 있다면 새로운 학생의 ID를 기존 학생들 중 최대 ID값 + 1로 부여
 - 수능 성적, 내신 성적이 정해진 범위를 벗어난다면 에러 메시지 출력
 - 이름이 최대 길이 (20자)보다 길다면 20자로 truncate해 추가함

6. 학생 삭제

- 학생을 삭제한다.
 - 학생 ID(int)를 입력 받아 해당되는 학생 삭제
 - 학생을 삭제하면 관련된 원서 접수 내역도 모두 삭제되어야 함

7. 원서 접수

- 학생을 학교에 접수한다.
 - 학생 ID(int) 및 학교 ID(int)를 입력 받음
 - 학교의 정원보다 더 많은 학생이 지원할 수 있음
 - 한 학생은 각 군마다 최대 1개의 학교에 지원할 수 있음. (예: A군에 속하는 두 개의 학교에 동시에 지원할 수 없음)

8. 학교에 지원한 학생 목록 출력

- 학교에 지원한 모든 학생의 정보를 출력한다.
 - 학교 ID(int)를 입력 받음
 - 해당 학교에 지원한 학생의 정보(학생 ID, 학생 이름, 수능 성적, 내신 성적)를 출력한다.

9. 학생의 원서 접수 목록 출력

- 학생이 지원한 모든 학교의 정보를 출력한다.
 - 학생 ID(int)를 입력 받음
 - 해당 학생이 지원한 학교의 정보(학교 ID, 학교 이름, 정원, 군, 내신 성적 반영 비율, 지원한 학생 수)를 출력한다.
 - 내신 성적 반영 비율은 소수점 셋째 자리에서 반올림하여 출력한다.

10. 학교의 예상 합격자 목록 출력

- 학교에 지원한 학생들 중, 합격이 예상되는 학생의 정보를 출력한다.
 - 학교 ID(int)를 입력 받음
 - 합격이 예상되는 학생의 정보(학생 ID, 학생 이름, 내신 성적, 수능 성적)를 출력한다.
 - 합격 여부는 종합 점수 순으로 결정된다고 가정한다. 최종 점수란 수능 성적과 내신 성적을 각 학교 별로 정해진 내신 성적 반영 비율을 고려하여 합한 값이다.
 - 예: 수능 성적이 300, 내신 성적이 100인 학생이 반영 비율 1.2인 학교에 지원한다면 종합 점수는 420이 됨
 - 종합 점수가 동점이라면 내신 성적이 높은 학생을 우선으로 한다.
 - 수능 성적과 내신 성적이 모두 동점인 지원자가 있어 정원을 초과할 경우, 정원의 10%

를 초과하지 않는다면 모두 합격 처리한다. 그렇지 않다면 모두 불합격 처리한다.

- 이 때, 소수점 단위는 올림한다. 즉, 정원이 91명인 학교에서는 10명을 초과하는 것 까지 허용된다.

11. 학생의 예상 합격 대학 목록 출력

- 학생이 지원한 학교들 중, 합격이 예상되는 학교의 정보를 출력한다.
- 학생 ID(int)를 입력 받음
- 합격이 예상되는 학교의 정보(학교 ID, 학교 이름, 정원, 군, 내신 성적 반영 비율, 지원한 학생 수)를 출력한다.

4 실행 예시

```
=====
1. print all universities
2. print all students
3. insert a new university
4. remove a university
5. insert a new student
6. remove a student
7. make an application
8. print all students who applied for a university
9. print all universities a student applied for
10. print expected successful applicants of a university
11. print universities expected to accept a student
12. exit
=====
Select your action: 1
-----
id      name                capacity  group  weight  applied
-----
1       Seoul Nat'l Univ.    100      B      0.2     41
2       Nakseong Univ.       80       A      0.1     230
-----
=====
1. print all universities
2. print all students
3. insert a new university
4. remove a university
5. insert a new student
6. remove a student
7. make an application
8. print all students who applied for a university
9. print all universities a student applied for
10. print expected successful applicants of a university
11. print universities expected to accept a student
12. exit
=====
Select your action: 2
-----
id      name                csat_score  school_score
-----
1       Park Junghyuk       330        87
2       Moon Byul           369        84
-----
```

```
=====
1. print all universities
2. print all students
3. insert a new university
4. remove a university
5. insert a new student
6. remove a student
7. make an application
8. print all students who applied for a university
9. print all universities a student applied for
10. print expected successful applicants of a university
11. print universities expected to accept a student
12. exit
=====
```

```
Select your action: 3
University name: Cheomseong Univ.
University capacity: 123
University group: C
Weight of high school records: 0.7
A university is successfully inserted.
=====
```

```
1. print all universities
2. print all students
3. insert a new university
4. remove a university
5. insert a new student
6. remove a student
7. make an application
8. print all students who applied for a university
9. print all universities a student applied for
10. print expected successful applicants of a university
11. print universities expected to accept a student
12. exit
=====
```

```
Select your action: 5
Student name: Sulley Ali Muntari
CSAT score: 288
High school record score: 100
A student is successfully inserted.
=====
```

```
1. print all universities
2. print all students
3. insert a new university
4. remove a university
5. insert a new student
6. remove a student
7. make an application
8. print all students who applied for a university
9. print all universities a student applied for
10. print expected successful applicants of a university
11. print universities expected to accept a student
12. exit
=====
```

```
Select your action: 1
```

```
-----
id      name                capacity  group  weight  applied
-----
1       Seoul Nat'l Univ.        100      B      0.2     41
2       Nakseong Univ.           80       A      0.1     230
3       Cheomseong Univ.         123      C      0.7     0
-----
```

```

-----
=====
1. print all universities
2. print all students
3. insert a new university
4. remove a university
5. insert a new student
6. remove a student
7. make an application
8. print all students who applied for a university
9. print all universities a student applied for
10. print expected successful applicants of a university
11. print universities expected to accept a student
12. exit
=====
Select your action: 2
-----
id            name            csat_score    school_score
-----
1             Park Junghyuk        330           87
2             Moon Byul            369           84
3             Sulley Ali Muntari    288           100
-----
=====
1. print all universities
2. print all students
3. insert a new university
4. remove a university
5. insert a new student
6. remove a student
7. make an application
8. print all students who applied for a university
9. print all universities a student applied for
10. print expected successful applicants of a university
11. print universities expected to accept a student
12. exit
=====
Select your action: 7
Student ID: 2
University ID: 3
Successfully made an application.
=====
1. print all universities
2. print all students
3. insert a new university
4. remove a university
5. insert a new student
6. remove a student
7. make an application
8. print all students who applied for a university
9. print all universities a student applied for
10. print expected successful applicants of a university
11. print universities expected to accept a student
12. exit
=====
Select your action: 1
-----
id      name            capacity    group    weight    applied
-----
1       Seoul Nat'l Univ.    100        B        0.2      41

```

2	Nakseong Univ.	80	A	0.1	230
3	Cheomseong Univ.	123	C	0.7	1

```
-----
=====
1. print all universities
2. print all students
3. insert a new university
4. remove a university
5. insert a new student
6. remove a student
7. make an application
8. print all students who applied for a university
9. print all universities a student applied for
10. print expected successful applicants of a university
11. print universities expected to accept a student
12. exit
=====
Select your action: 12
Bye!
```

5 개발 환경

- Java
- Eclipse
- MariaDB 10.1
 - ◆ DB 접속 정보
 - Host: 147.46.15.238
 - Port: 3306
 - ID: DB-[학번] (예: DB-2015-12345)
 - Password: DB-[학번]

6 제출

1. Runnable jar 파일
 - 파일명: PRJ2_학번.jar (예: PRJ2_2015-12345.jar)
 - jar 파일 생성 방법은 강의 홈페이지의 공지사항을 참조
2. Eclipse 프로젝트
 - 소스 파일은 반드시 적절한 주석을 포함하여야 함
3. DB

- 프로젝트 제출시 자신의 DB 계정 안의 테이블 스키마만 남기고 레코드는 모두 삭제해야 함
- **이를 지키지 않았을 시 채점이 제대로 되지 않아 0점 처리 될 수 있으니 주의!**

4. 리포트

- 파일명: PRJ2_학번.pdf (예: PRJ2_2015-12345.pdf)
- 반드시 pdf 포맷으로 제출
- 포함되어야 하는 내용
 - ◆ 핵심 모듈과 알고리즘에 대한 설명
 - ◆ 구현한 내용에 대한 간략한 설명
 - ◆ (제시된 요구사항 중 구현하지 못한 부분이 있다면) 구현하지 못한 내용
 - ◆ 가정한 것들
 - ◆ 컴파일과 실행 방법
 - ◆ 프로젝트를 하면서 느낀 점
- 위의 3가지 파일을 압축하여 lecture@europa.snu.ac.kr로 제출
 - 파일명: PRJ2_학번.zip (예: PRJ2_2015-12345.zip)
 - 메일 제목: [DB Project2] 학번, 이름 (예: [DB Project2] 2015-12345, 박정혁)
- 리포트의 hard copy는 12월 14일 오후 6시까지 301동 420호로 제출
 - Hard copy와 soft copy를 모두 제출하여야 함

7 성적 관련 사항

- 제출 기한 이후 24시간 이내 제출시 10% 감점
- 제출 기한 이후 24시간 이후 48시간 이내 제출시 20% 감점
- 제출 기한 48시간 이후에는 점수 없음
- 부정 행위는 0점 처리
 - ◆ 다른 사람의 코드를 참조하는 행위
 - ◆ 이전에 수강한 사람의 코드를 참조하는 행위

8 References

- MariaDB

- <https://mariadb.org/>

프로젝트 관련 질문은 eTL 게시판을 이용!