

데이터 베이스 시스템

수강신청 어플리케이션 개발 과제

컴퓨터 전공

2016026026 최슬비

목차

1. 데이터베이스 구축

1) *schema* 생성

2) *table* 생성 및 *data import*

2. 사용자 어플리케이션 개발

1) 개발 환경

2) 코드 설명

3) 시연 동영상에 대한 설명

1. 데이터베이스 구축

1) schema 생성

스키마 생성 및 테이블 생성, 수정은 모두 mySQL Workbench라는 GUI 툴을 이용하였다. 워크벤치 상에서 mysql 로컬 서버 상에 university 라는 이름의 스키마를 우선 생성하였다. 이하 테이블들은 university 스키마에 생성되었다. 어플리케이션 개발 시에 활용한 서버 커넥션 주소는 다음과 같다.

`jdbc:mysql://localhost:3306/university?
useUnicode=yes&characterEncoding=UTF-8&useSSL=false`

한글 데이터를 원활히 검색, 수정하기 위해 url에 인코딩 정보를 첨부해야 했다.

2) table 생성 및 data import

사용한 table은 주어진 8개의 csv파일을 그대로 import 한 테이블과 관리자 정보를 저장하는 admin table, 프로그램 수행 도중 쿼리 수행하여 생성하는 1개의 register table 이다. 총 10개의 table을 사용하였다. register table 은 수강신청 내역을 기록하는 table이며, 관리자 로그인을 해서 initialize 명령을 함으로써 생성하게 된다(수강신청 시작). 이후 일반 학생 로그인을 통해 수강신청을 할 수 있게 된다. 다음은 10개 table 의 DDL 이다.

(a) admin

관리자 정보를 관리하기 위해 생성한 table이다. student table 과 호환할 수 있는 id 와 password 형식을 사용 하였으며, adminID는 primary key 이다.

```
CREATE TABLE `admin` (  
  `adminID` int(11) NOT NULL,  
  `adminPassword` varchar(45) NOT NULL,  
  PRIMARY KEY (`adminID`)  
) ENGINE=InnoDB DEFAULT CHARSET=utf8
```

(b) building

```
CREATE TABLE `building` (  
  `building_id` int(11) DEFAULT NULL,  
  `name` text,  
  `admin` text,  
  `rooms` int(11) DEFAULT NULL  
) ENGINE=InnoDB DEFAULT CHARSET=utf8
```

(c) class

기존의 csv파일 import한 형식에서 필드명 class_id 와 class_no 만 각각 classID, classNO 로 바꿔주었다. JDBC 상에서_(언더바)를 인식하는데에 에러가 발생했기 때문이다. 수강 신청과 강의 검색을 할 때에는 year=2014로 제약하도록 한다.

```
CREATE TABLE `class` (  
  `classID` int(11) DEFAULT NULL,  
  `classNO` int(11) DEFAULT NULL,  
  `course_id` text,  
  `name` text,  
  `major_id` int(11) DEFAULT NULL,  
  `year` int(11) DEFAULT NULL,  
  `credit` int(11) DEFAULT NULL,  
  `lecturer_id` int(11) DEFAULT NULL,
```

(d) course

```
CREATE TABLE `course` (  
  `course_id` text,  
  `name` text,  
  `credit` int(11) DEFAULT NULL  
) ENGINE=InnoDB DEFAULT CHARSET=utf8
```

(e) lecturer

```
CREATE TABLE `lecturer` (  
  `lecturer_id` int(11) DEFAULT NULL,  
  `name` text,  
  `major_id` int(11) DEFAULT NULL  
) ENGINE=InnoDB DEFAULT CHARSET=utf8
```

(f) major

```
CREATE TABLE `major` (  
  `major_id` int(11) DEFAULT NULL,  
  `name` text  
) ENGINE=InnoDB DEFAULT CHARSET=utf8
```

(g) register

수강 신청 정보를 기록하기 위해 새롭게 생성한 table이다. 자동으로 증가하는 registerID와 신청한 학생 정보를 저장하기 위해 studentID, 신청한 class를 저장하기 위해 classID를 기록하도록 한다.

```
CREATE TABLE `register` (  
  `registerID` int(11) NOT NULL AUTO_INCREMENT,  
  `studentID` int(11) NOT NULL,  
  `classID` int(11) NOT NULL,  
  PRIMARY KEY (`registerID`)  
) ENGINE=InnoDB AUTO_INCREMENT=3 DEFAULT CHARSET=utf8
```

(h) room

```
CREATE TABLE `room` (  
  `room_id` int(11) DEFAULT NULL,  
  `building_id` int(11) DEFAULT NULL,  
  `occupancy` int(11) DEFAULT NULL  
) ENGINE=InnoDB DEFAULT CHARSET=utf8
```

(i) student

필드명 student_id를 studentID로 변경하였다.

```
CREATE TABLE `student` (  
  `studentID` int(11) NOT NULL,  
  `password` text,  
  `name` text,  
  `sex` text,  
  `major_id` int(11) DEFAULT NULL,  
  `tutor_id` int(11) DEFAULT NULL,  
  `year` int(11) DEFAULT NULL,  
  PRIMARY KEY (`studentID`)  
) ENGINE=InnoDB DEFAULT CHARSET=utf8
```

(j) time

```
CREATE TABLE `time` (  
  `time_id` int(11) DEFAULT NULL,  
  `classID` int(11) DEFAULT NULL,  
  `period` int(11) DEFAULT NULL,  
  `begin` text,  
  `end` text  
) ENGINE=InnoDB DEFAULT CHARSET=utf8
```

2. 사용자 어플리케이션 개발

1) 개발 환경

데이터베이스는 mySQL을 통해 관리하였으며, 주로 워크벤치 툴을 사용하여 관리, 작업하였다. 사용자 어플리케이션을 만드는데 사용한 언어는 자바이다. 또한 자바 소스 코드 상에서 SQL쿼리를 수행할 수 있도록 해주는 인터페이스인 JDBC를 활용하였다. 개발 툴은 이클립스를 사용하였고 export한 jar 파일을 제출 폴더에 첨부하였다(test.jar).

2) 코드 설명

구현한 어플리케이션 자바 프로그램은 8개의 자바 파일로 이루어져 있으며, 이 중 하나는 추상 클래스이다. 아래에 각각의 코드가 하는 역할에 대해 설명한다.

(a) Login.java

main함수를 포함하고 있는 로그인 클래스이다. 실행시 String arg 인자로 ID와 password를 같이 넘겨 받도록 되어있다. DBConnection 객체를 할당한 후, DBConnection 클래스 메소드인 isAdmin() 과 isStudent()를 차례대로 수행하여 입력한 ID와 password가 관리자이거나 학생인지 확인한다.

(b) DBConnection.java

서버와 연결하는 클래스이다. Connection 객체에 서버 url을 통해 연결을 시도하고, 해당 객체는 자동커밋이 안되도록 설정한다. 이는 rollback기능을 구현하기 위함이다. 내부 메소드로 isAdmin() 과 isStudent()가 있다. 이는 각각 데이터베이스의 admin 클래스와 student 클래스에 SELECT 쿼리를 수행하여 그 결과를 result set 객체로 받아온다. 메인 함수에서 인자로 받아온 아이디와 비밀번호 정보가 데이터베이스 내에 존재 하는지 확인하여 boolean형태로 리턴한다.

(c) Program.java

추상클래스이다. ManagerProgram과 StudentProgram 은 Program을 상속받는다. 관리자와 학생 둘다 이용하게 될 변수와 메소드들을 포함하고 있다. input(), save(), rollback() 메소드는 관리자와 학생이 공통적으로 사용하게 된다. input()은 명령어를 받아 파싱하여 멤버변수 order 배열에 저장한다. 또 히스토리를 위해 파싱하지 않은 명령어 전체를 commands 배열에 저장한다. save()는 setSavepoint() 메소드를 호출하여 Savepoint 객체 배열에 세이프포인트를 저장한다. rollBack()은 바로 이전 세이프 인덱스로 세션을 되돌

리도록 rollback()을 호출한다. 이 외에도 Program 클래스는 help(), process()의 추상 메소드를 선언한다.

(d) ManagerProgram.java

Program 클래스를 상속한 클래스이다. 관리자 모드에서 할 수 있는 기능들을 담고 있다. Program에서 상속받은 메소드 이외에도 init(), modify(), classControl() 메소드가 있는데 이들의 기능은 각각 다음과 같다. init()은 register table을 생성하여 학생 계정에서 수강신청이 가능한 환경을 구축한다. modify()는 옵션과 ID를 인자로 받아 해당 아이디의 비밀번호를 바꿀 수 있도록 한다. 관리자 모드에서 정보수정이 가능한 ID는 학생 ID와 관리자 ID 모두에 해당된다. classControl()은 옵션과 classID를 인자로 받아 강좌 개설과 폐강을 가능하게 한다. class close 옵션을 입력했을 시에 class table에서 해당 classID를 검색하여 opened 컬럼(2014년 개설)을 -1로 수정한다(open은 2014로 수정).

상속받은 help() 메소드에서는 관리자가 사용할 수 있는 명령어를 명시 하였으며, process()메소드에서는 while 루프를 이용하여 매 루프 input()메소드를 호출하고 입력값에 대응하는 액션을 취하도록 하였다. 또한 데이터베이스 테이블에 영향을 미치는 SQL 쿼리 수행시 마다 save()메소드를 호출하여 세이프 포인트를 기록하고, 명령어 cancel 입력시에 rollBack()메소드를 호출하여 이전 세션으로 돌아가도록 하였다.

(e) StudentProgram.java

이 클래스도 Program 클래스를 상속하였다. 학생 모드에서 수행할 기능들을 담고 있다. 우선 modify(), check(), classRD() 메소드가 있다. modify() 메소드는 password를 인자로 받아 학생 자기 자신의 비밀번호를 수정할 수 있다. 수정된 정보는 student table에 저장된다. check() 메소드는 로그인 한 학생이 수강신청한 강의들을 select문을 실행하여 보여준다. select 쿼리를 통해서 데이터베이스의 register table에서 studentID와 동일한 행들을 검색하여 출력하게 된다. classRD() 메소드는 강의 신청과 삭제를 구현한 메소드이다. 옵션과 classID를 인자로 받아 insert into와 delete문을 이용하여 데이터베이스의 register table에 정보를 저장, 삭제한다.

관리자 프로그램과 마찬가지로 Program.java로부터 help()와 process()를 상속받게 된다. help()는 학생이 사용할 수 있는 명령어를 명시하였다. process()도 관리자 프로그램 메소드와 유사한 역할을 수행하고 있다.

(f) SearchClass.java

강의 검색 기능을 구현하기 위해 작성한 코드이다. 5가지 옵션을 통해서 기본 select SQL 문을 조작하여 과제에서 명시한 검색 기능을 제공하였다. 이는 manipulate()에 담았으며, 가공된 쿼리를 실행하여 출력하는 것은 print()가 수행하고 있다.

(g) Statistics.java

관리자 모드에서 필요한 기능인 통계처리를 위해 작성한 클래스이다.

(h) TimeTable.java

학생 모드에서 필요한 기능인 시간표 출력을 위해 작성한 클래스이다.

3) 시연 동영상에 대한 설명

제출한 과제파일 가운데 2016026026최솔비.mov 는 test.jar을 실행하는 것을 기록한 영상이다. (구현한 기능에 대한 사용을 연속적으로 시연하기 위해 UI스크린샷 대신 제출하였습니다) 우선 관리자 모드로 접속하여 실행가능한 명령어들을 하나하나 동작해보았다. init()명령어를 입력하였을 때 위의 코드 항목에서 설명한 것과 같이 register class가 생성된 것을 확인할 수 있다. 그 다음 2016026026(학생 본인의 ID: student table 에 미리 삽입해놓았음)의 비밀번호를 solbi2로 수정하였다. 이후 수정된 비밀번호로 접속하는 것을 확인할 수 있다. 이 후 all 명령어를 사용하여 2014년도 강의를 모두 출력한다. stats 명령어를 이용하여 컴퓨터전공의 통계 또한 출력 됨을 확인할 수 있다. class close를 통해 opened state를 수정하였고, 이후 cancel 명령어를 이용하여 액션을 취소하였다. 검색 기능 테스트 후 quit을 입력하여 로그아웃 한후 학생모드로 다시 로그인 하였다. 학생 모드에서는 자기 자신의 비밀번호만 수정할 수 있다. 이후 class register을 하여 check 와 tt(time table)명령어로 확인하였으며 cancel 명령어가 역시 잘 동작하는 것을 확인하였다. 유의할 점은 일련의 명령어 실행후 quit 명령어를 통해 정상종료 하여야 데이터베이스 상에 정상적으로 모두 커밋된다는 것이다.