객체지향프로그래밍응용 과제13 컴퓨터소프트웨어공학과 2-YA 20202296 전채린

학생 관리 프로젝트

[소스코드]

import java.awt.\*; // 폰트 등 그래픽 처리를 위한 클래스들의 경로명

import java.awt.event.\*; // 이벤트 처리에 필요한 기본 클래스들의 경로명

import javax.swing.\*; // 스윙 컴포넌트 클래스들의 경로명

import java.sql.\*;

// 학생 입력 프로젝트

public class InputEx1 extends JFrame {

JButton btnInsert, btnDelete, btnUpdate, btnSelect, btnSearch; // 입력 버튼, 삭제 버튼, 수정 버튼, 조회 버튼, 검색버튼

JTextField tfId, tfName, tfDept, tfSearch; // 학번 텍스트 필드, 이름 텍스트 필드, 학과 텍스트 필드, 검색 텍스트 필드

JTextArea ta; // 텍스트영역

JRadioButton rbId, rbName, rbDept; // 학번 라디오 버튼, 이름 라디오 버튼, 학과 라디오 버튼

Connection conn;

Statement stmt; // SQL 문을 실행하기 위해서는 Statement 클래스를 이용

ResultSet rs; // SQL 문 실행 결과를 얻어오기 위해서는 ResultSet 클래스를 이용

public InputEx1() { // 생성자

this.setTitle("학생 관리 프로젝트"); // 프레임 타이틀 달기

this.setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE); // 프레임 윈도우를 닫으면 프로그램을 종료하도록 설정

createGUI(); // GUI 생성 메소드

btnInsert.addActionListener(new ActionListener() { // 입력 버튼 액션 이벤트 처리

@Override

public void actionPerformed(ActionEvent e) {

dbInsert(); // 데이터 삽입 메소드

}

});

btnDelete.addActionListener(new ActionListener() { // 삭제 버튼 액션 이벤트 처리

@Override

public void actionPerformed(ActionEvent e) {

dbDelete(); // 데이터 삭제 메소드

}

});

btnUpdate.addActionListener(new ActionListener() { // 수정 버튼 액션 이벤트 처리

@Override

public void actionPerformed(ActionEvent e) {

dbUpdate(); // 데이터 수정 메소드

}

});

btnSelect.addActionListener(new ActionListener() { // 조회 버튼 액션 이벤트 처리

@Override

public void actionPerformed(ActionEvent e) {

dbSelect(); // 데이터 조회 메소드

}

});

btnSearch.addActionListener(new ActionListener() { // 검색 버튼 액션 이벤트 처리

@Override

public void actionPerformed(ActionEvent e) {

dbSearch(); // 데이터 검색 메소드

}

});

this.setSize(300, 500); // 프레임 크기 300x500 설정

this.setVisible(true); // 화면에 프레임 출력

}

public void createGUI() { // GUI 생성 메소드

Container c = this.getContentPane(); // 프레임의 컨텐트팬

c.setLayout(new FlowLayout()); // FlowLayout 배치관리자 설정

c.add(new JLabel("학번")); // 학번 레이블 달기

tfId = new JTextField(20); // 학번 텍스트 필드 생성

c.add(tfId); // 학번 텍스트 필드 달기

c.add(new JLabel("이름")); // 이름 레이블 달기

tfName = new JTextField(20); // 이름 텍스트 필드 생성

c.add(tfName); // 이름 텍스트 필드 달기

c.add(new JLabel("학과")); // 학과 레이블 달기

tfDept = new JTextField(20); // 이름 텍스트 필드 생성

c.add(tfDept); // 학과 텍스트 필드 달기

btnInsert = new JButton("입력"); // 입력 버튼 생성

c.add(btnInsert); // 입력 버튼 컨텐트팬에 부착

btnUpdate = new JButton("수정"); // 수정 버튼 생성

c.add(btnUpdate); // 수정 버튼 컨텐트팬에 부착

btnDelete = new JButton("삭제"); // 삭제 버튼 생성

c.add(btnDelete); // 삭제 버튼 컨텐트팬에 부착

btnSelect = new JButton("조회"); // 조회 버튼 생성

c.add(btnSelect); // 조회 버튼 컨텐트팬에 부착

// 검색부분

tfSearch = new JTextField(18); // 검색 텍스트 필드 생성

rbId = new JRadioButton("학번", true); // 학번 라디오 버튼 생성. 초기 선택

rbName = new JRadioButton("이름"); // 이름 라디오 버튼 생성

rbDept = new JRadioButton("학과"); // 학과 라디오 버튼 생성

ButtonGroup group = new ButtonGroup(); // 버튼 그룹 객체 생성

group.add(rbId); // 학번 라디오 버튼을 버튼 그룹에 부착

group.add(rbName); // 이름 라디오 버튼을 버튼 그룹에 부착

group.add(rbDept); // 학과 라디오 버튼을 버튼 그룹에 부착

btnSearch = new JButton("검색"); // 검색 버튼 생성

JPanel pn1 = new JPanel(); // 레이블, 검색 텍스트 필드, 검색 버튼을 부착할 패널 생성

pn1.add(new JLabel(" ")); // 레이블을 패널에 부착

pn1.add(tfSearch); // 검색 텍스트 필드를 패널에 부착

pn1.add(btnSearch); // 검색 버튼을 패널에 부착

JPanel pn2 = new JPanel(); // 3개의 라디오버튼을 부착할 패널 생성

pn2.add(rbId); // 학번 라디오 버튼을 패널에 부착

pn2.add(rbName); // 이름 라디오 버튼을 패널에 부착

pn2.add(rbDept); // 학과 라디오 버튼을 패널에 부착

JPanel pMiddle = new JPanel(new BorderLayout(0, 0)); // 2개의 패널을 부착할 패널 생성

pMiddle.add(pn1, BorderLayout.NORTH); // pn1 패널을 pMiddle 패널에 부착

pMiddle.add(pn2, BorderLayout.CENTER); // pn2 패널을 pMiddle 패널에 부착

c.add(pMiddle); // 컨텐트팬에 pMiddle 패널 부착

ta = new JTextArea(15, 20); // 텍스트영역 생성

c.add(ta); // 컨텐트팬에 텍스트영역 부착

}

public void dbInsert() { // 데이터 입력 메소드

try {

conn = DBConn.dbConnection(); // DB 연결

stmt = conn.createStatement();

String id = tfId.getText().toString(); // 학번 텍스트 필드에 입력된 데이터 가져오기

String name = tfName.getText().toString(); // 이름 텍스트 필드에 입력된 데이터 가져오기

String dept = tfDept.getText().toString(); // 학과 텍스트 필드에 입력된 데이터 가져오기

stmt.executeUpdate("insert into student values('" + id + "', '" + name + "', '" + dept + "');"); // insert 문을 이용하여 데이터 삽입 명령

System.out.println(name + " 입력 완료"); // 콘솔에 결과 출력

tfId.setText(""); // 학번 텍스트 필드 공백으로 설정

tfName.setText(""); // 이름 텍스트 필드 공백으로 설정

tfDept.setText(""); // 학과 텍스트 필드 공백으로 설정

stmt.close(); // stmt 닫기

conn.close(); // conn 닫기

} catch (Exception e) {

e.printStackTrace();

}

}

public void dbDelete() { // 데이터 삭제 메소드

try {

conn = DBConn.dbConnection(); // DB 연결

stmt = conn.createStatement();

String inputid = tfId.getText().toString(); // 학번 텍스트 필드에 입력된 데이터 가져오기

stmt.executeUpdate("delete from student where id = '" + inputid + "'"); // delete 문을 이용하여 데이터 삭제 명령

System.out.println(inputid +" 삭제 완료"); // 콘솔에 결과 출력

tfId.setText(""); // 학번 텍스트 필드 공백으로 설정

tfName.setText(""); // 이름 텍스트 필드 공백으로 설정

tfDept.setText(""); // 학과 텍스트 필드 공백으로 설정

stmt.close(); // stmt 닫기

conn.close(); // conn 닫기

} catch (Exception e) {

e.printStackTrace();

}

}

public void dbSearch() { // 검색 메소드

try {

conn = DBConn.dbConnection(); // DB 연결

stmt = conn.createStatement();

String searchText = tfSearch.getText().toString(); // 검색 텍스트 필드에 입력된 데이터 가져오기

String searchSql = "";

if(rbId.isSelected()) // 학번 라디오 버튼이 선택되었을 경우

searchSql = "select \* from student where id = '" + searchText + "';"; // select 문을 이용하여 데이터 조회 명령

else if(rbName.isSelected()) // 이름 라디오 버튼이 선택되었을 경우

searchSql = "select \* from student where name = '" + searchText + "';"; // select 문을 이용하여 데이터 조회 명령

else // 학과 라디오 버튼이 선택되었을 경우

searchSql = "select \* from student where dept = '" + searchText + "';"; // select 문을 이용하여 데이터 조회 명령

rs = stmt.executeQuery(searchSql); // Statement의 excuteQuery()는 SQL 문을 실행하여 실행 결과를 넘겨줌. 모든 행의 모든 열을 읽어 결과를 rs에 저장

String line = "";

ta.setText("id name dept\n");

ta.append("--------------------------------------------------------------\n");

while(rs.next()) { // 다음 행이 있으면 true

String name = rs.getString("name");

String dept = rs.getString("dept");

String id = rs.getString("id");

line = " | " + id + " | " + name + " | " + dept + "\n";

System.out.println("rs => " + line); // 콘솔에 결과 출력

ta.append(line); // update 문을 이용하여 데이터 수정 명령

tfSearch.setText(""); // 검색 텍스트 필드 공백으로 설정

}

stmt.close(); // stmt 닫기

conn.close(); // conn 닫기

} catch(Exception e) {

e.printStackTrace();

}

}

public void dbSelect() { // 데이터 조회 메소드

try {

conn = DBConn.dbConnection(); // DB 연결

stmt = conn.createStatement();

rs = stmt.executeQuery("select \* from student;"); // Statement의 excuteQuery()는 SQL 문을 실행하여 실행 결과를 넘겨줌. 모든 행의 모든 열을 읽어 결과를 rs에 저장. select 문을 이용하여 데이터 조회 명령

String line = "";

ta.setText("id name dept\n");

ta.append("--------------------------------------------------------------\n");

while(rs.next()) { // 다음 행이 있으면 True

String name = rs.getString("name");

String dept = rs.getString("dept");

String id = rs.getString("id");

line = " | " + id + " | " + name + " | " + dept + "\n";

System.out.println("rs => " + line); // 콘솔에 결과 출력

ta.append(line); // 텍스트영역에 추가

tfId.setText(""); // 학번 텍스트 필드 공백으로 설정

tfName.setText(""); // 이름 텍스트 필드 공백으로 설정

tfDept.setText(""); // 학과 텍스트 필드 공백으로 설정

}

stmt.close(); // stmt 닫기

conn.close(); // conn 닫기

} catch(Exception e) {

e.printStackTrace();

}

}

public void dbUpdate() { // 데이터 수정 메소드

try {

conn = DBConn.dbConnection(); // DB 연결

stmt = conn.createStatement();

String in\_id = tfId.getText().toString(); // 학번 텍스트 필드에 입력된 데이터 가져오기

String in\_name = tfName.getText().toString(); // 이름 텍스트 필드에 입력된 데이터 가져오기

String in\_dept = tfDept.getText().toString(); // 학과 텍스트 필드에 입력된 데이터 가져오기

stmt.executeUpdate("update student set dept = '" + in\_dept + "', name = '" + in\_name + "' where id = '" + in\_id + "'"); // update 문을 이용하여 데이터 수정 명령

System.out.println(in\_name + " 수정완료"); // 콘솔에 결과 출력

tfId.setText(""); // 학번 텍스트 필드에 입력된 데이터를 지우기

tfName.setText(""); // 이름 텍스트 필드에 입력된 데이터를 지우기

tfDept.setText(""); // 학과 텍스트 필드에 입력된 데이터를 지우기

stmt.close(); // stmt 닫기

conn.close(); // conn 닫기

} catch(Exception e) {

e.printStackTrace();

System.err.println("수정 에러");

}

}

public static void main(String[] args) {

new InputEx1(); // InputEx1 객체 생성

}

}

[SQL 소스코드]

-- (InputEx1-sql.sql)

-- student(id, name, dept)

-- 데이터베이스 생성

create database sampledb;

-- 사용할 데이터베이스 선택

use sampledb;

-- 테이블 삭제

drop table student;

-- 테이블 생성

-- student(id, name, dept)

create table student (

id char(7) primary key ,

name varchar(10) ,

dept varchar(20)

);

-- 데이터 입력

-- student(id, name, dept)

insert into student values('2020001', '김철수', '컴퓨터공학과');

insert into student values('2020002', '이영희', '정보통신학과');

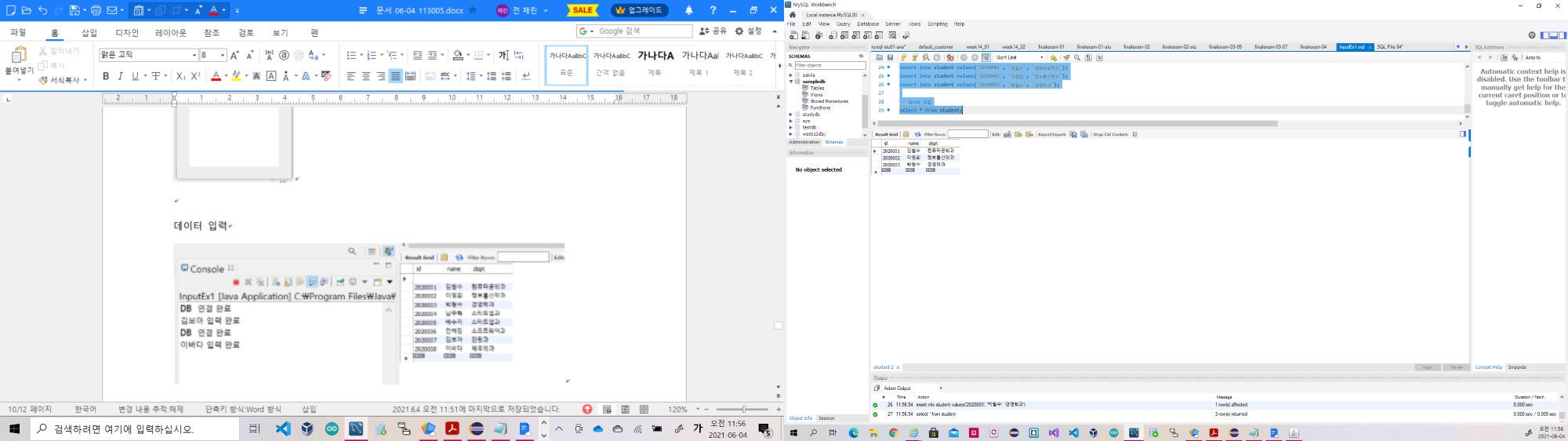
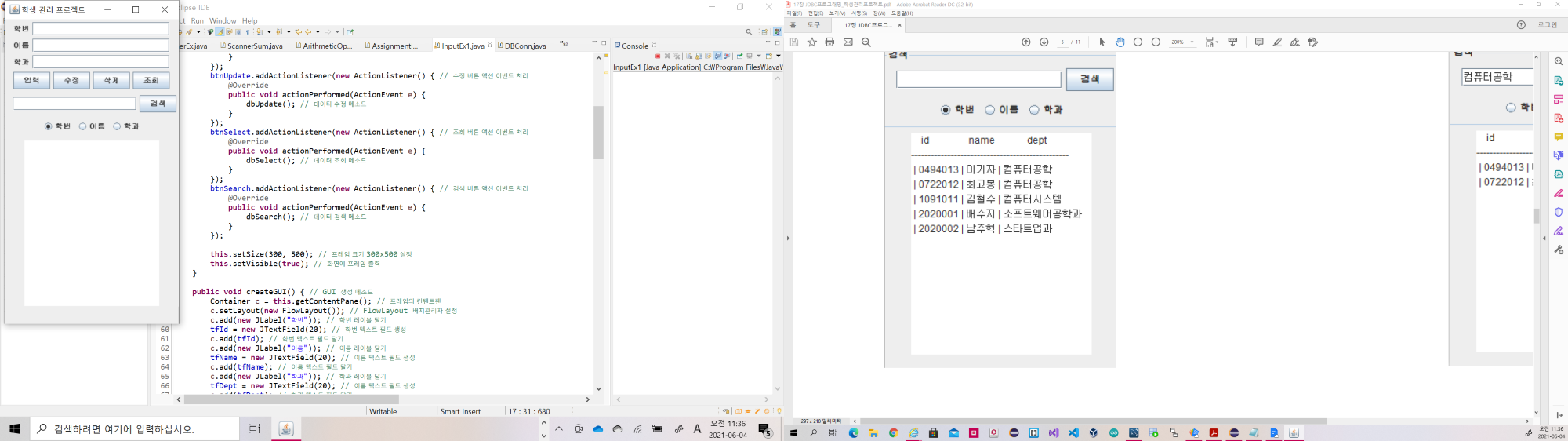
insert into student values('2020003', '박철수', '경영학과');

-- 데이터 조회

select \* from student;

[실행결과]

초기화면



데이터 입력

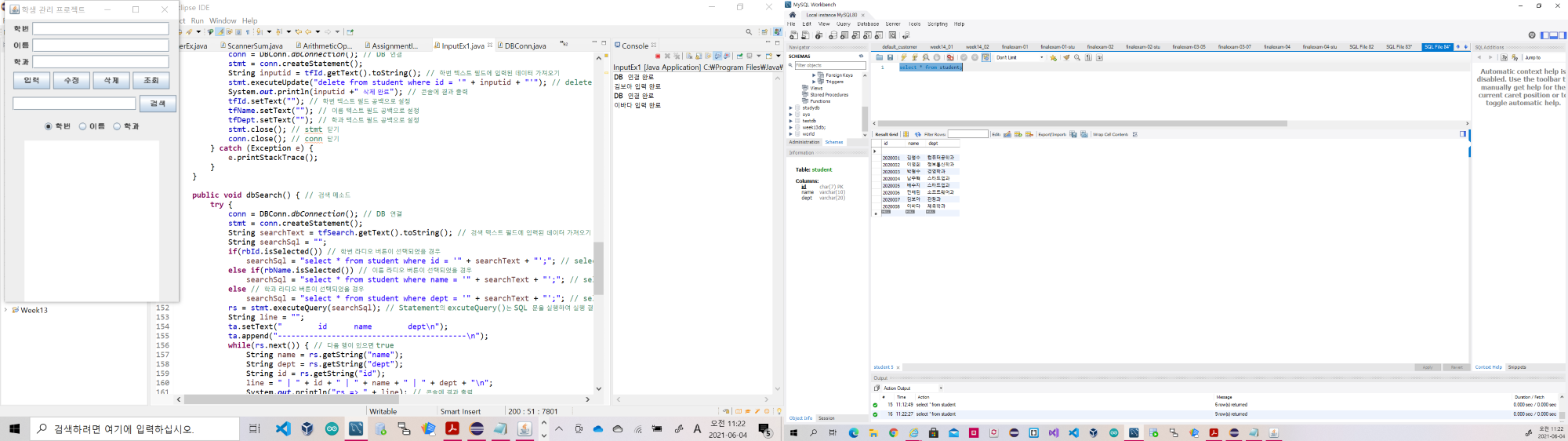
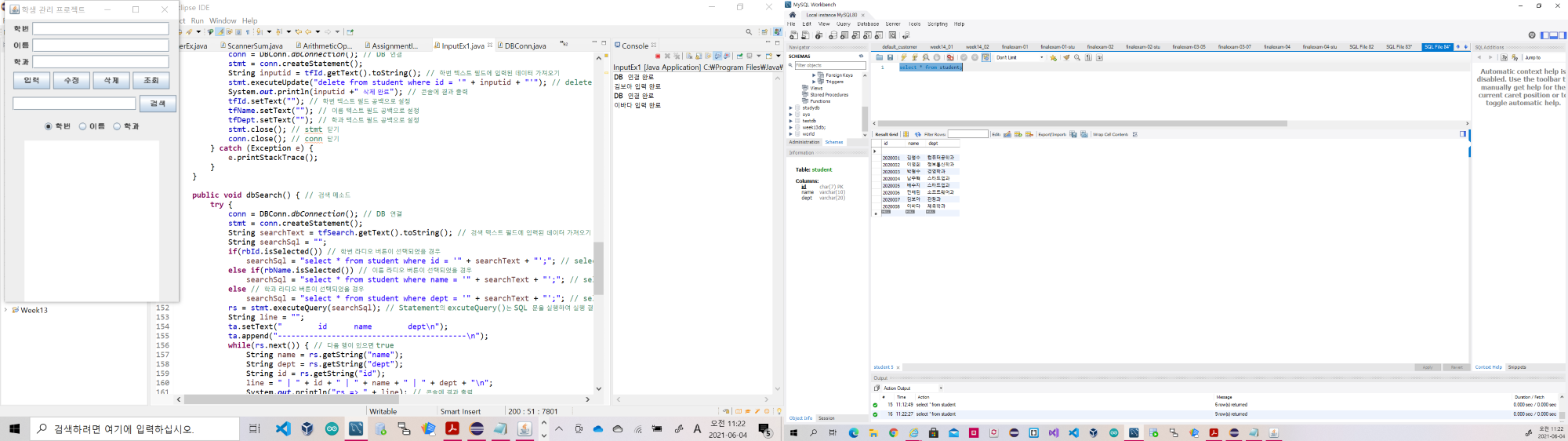
학번이 2020004, 이름이 남주혁, 학과가 스타트업과인 데이터 삽입

학번이 2020005, 이름이 배수지, 학과가 스타트업과인 데이터 삽입

학번이 2020006, 이름이 전채린, 학과가 소프트웨어과인 데이터 삽입

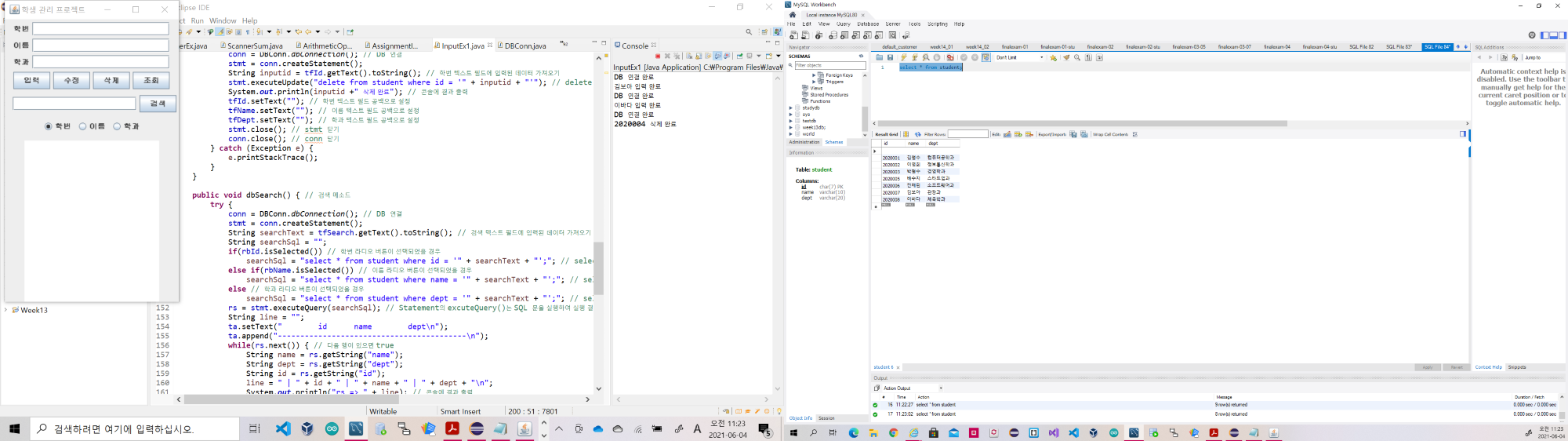
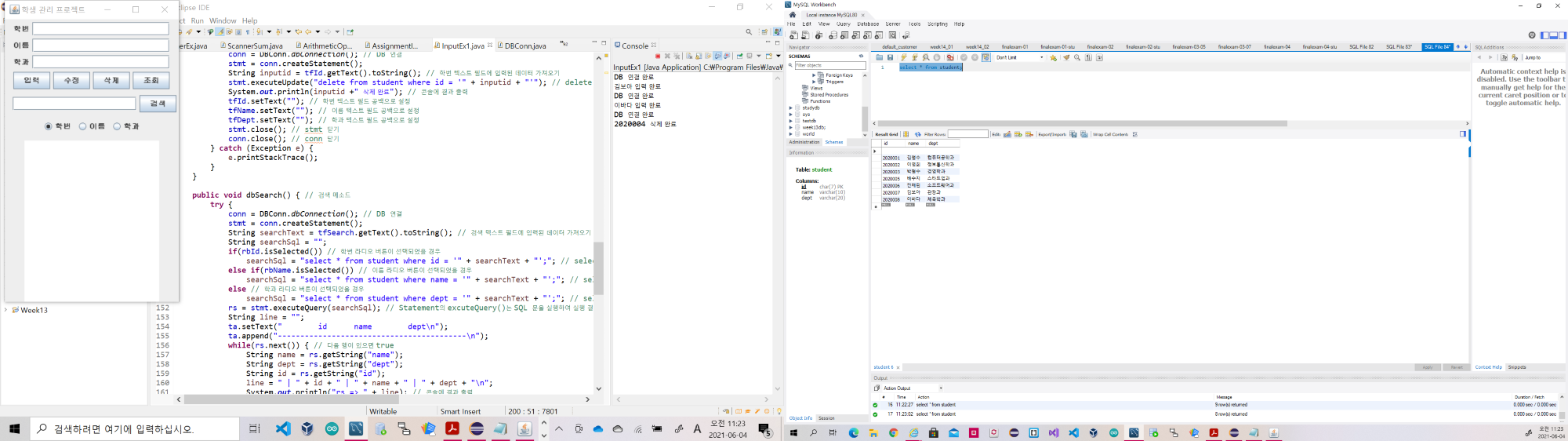
학번이 2020007, 이름이 김보아, 학과가 관광과인 데이터 삽입

학번이 2020008, 이름이 이바다, 학과가 체육학과인 데이터 삽입



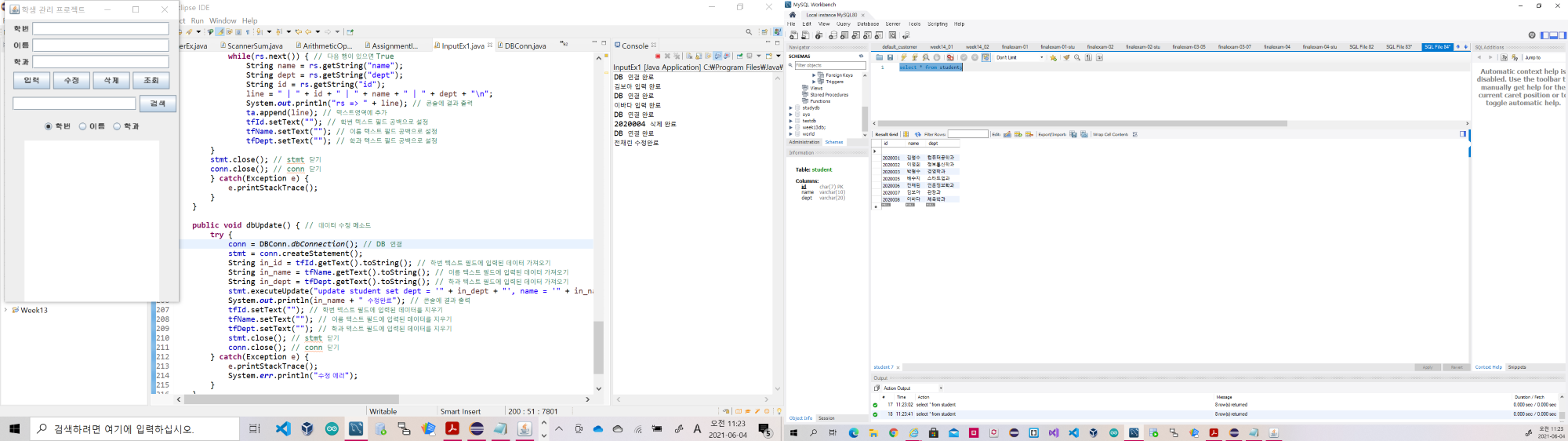
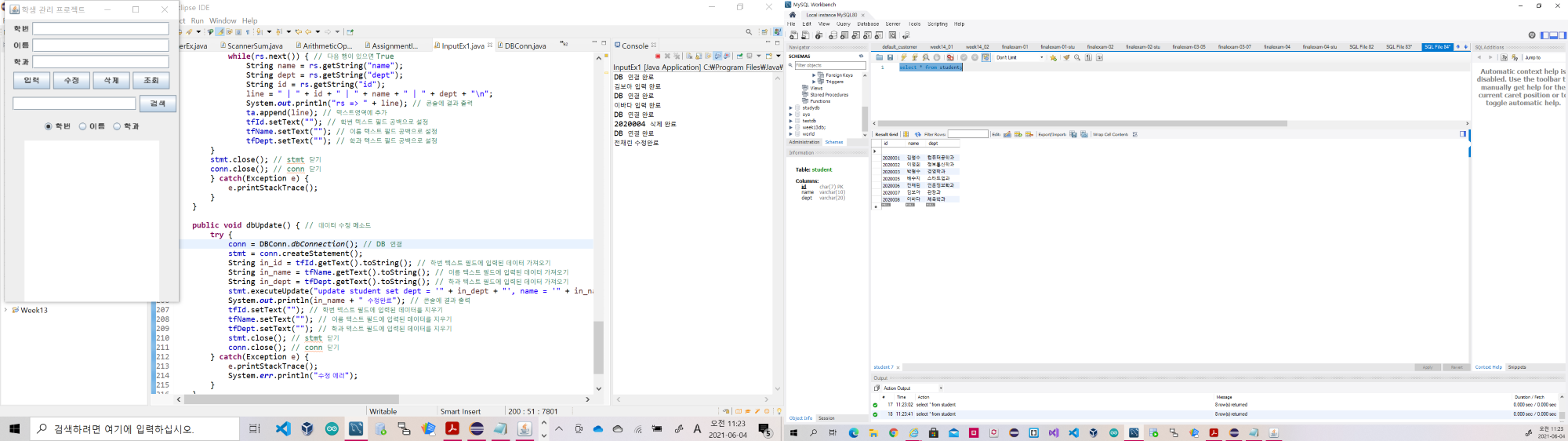
데이터 삭제

학번이 2020004인 데이터 삭제

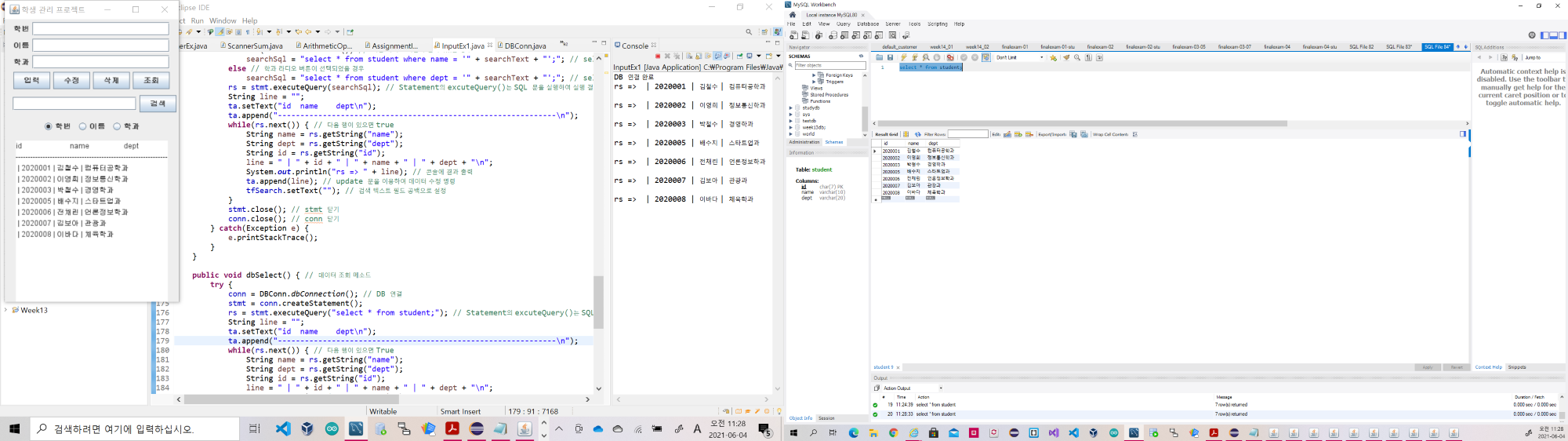
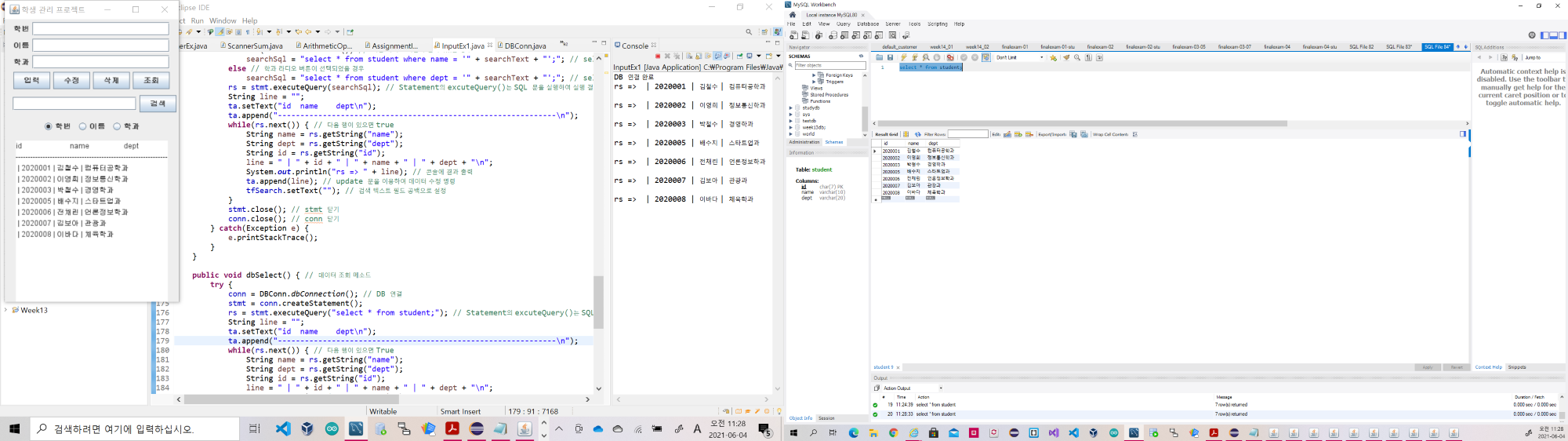
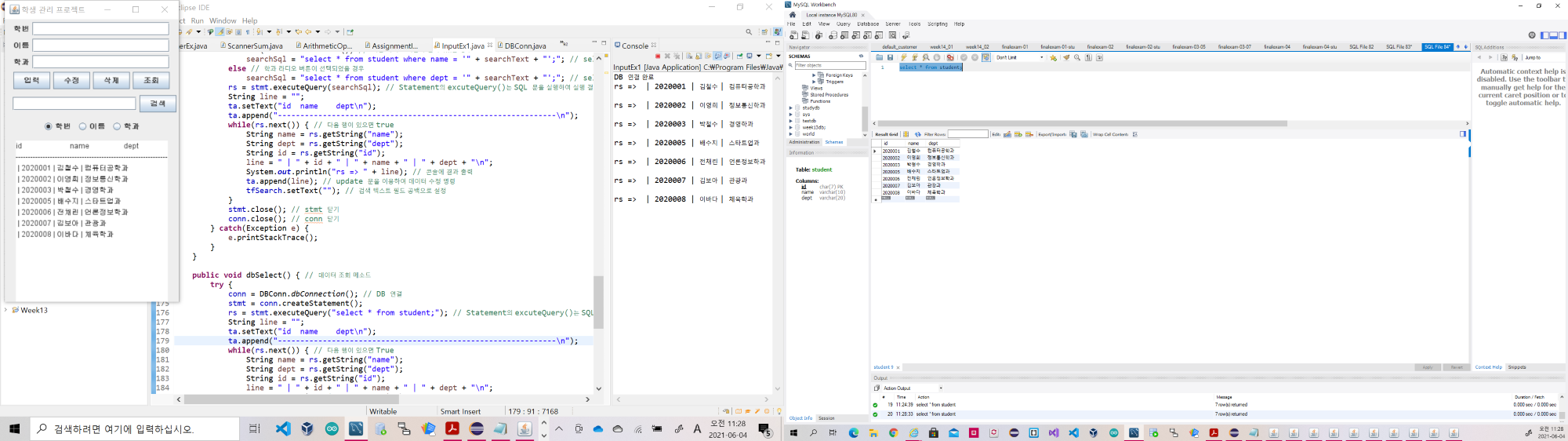


데이터 수정

학번이 2020006인 전채린의 학과를 언론정보학과로 수정

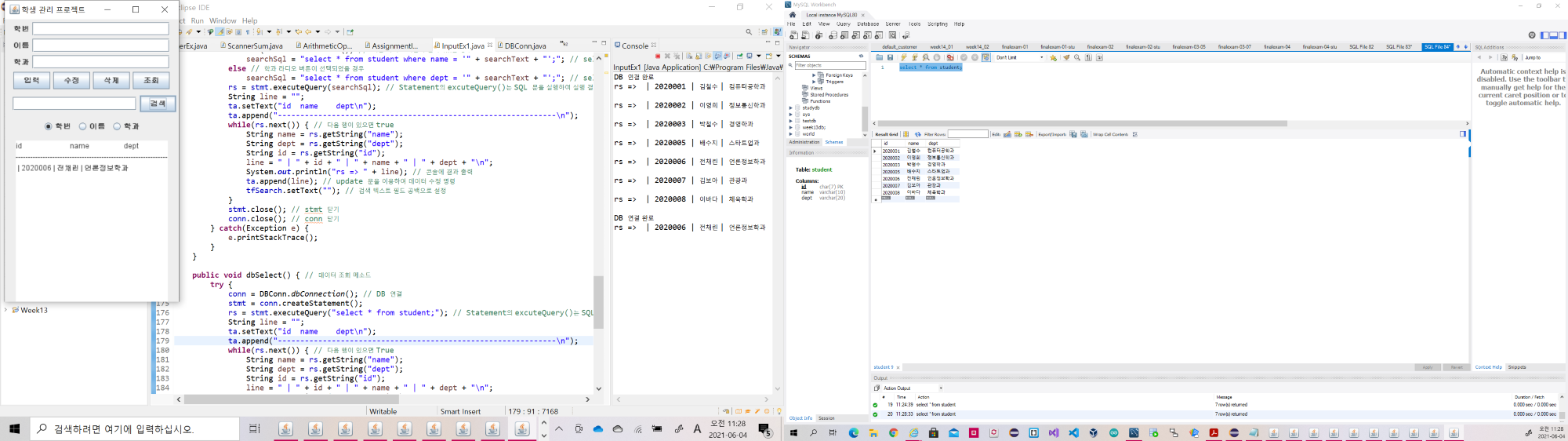
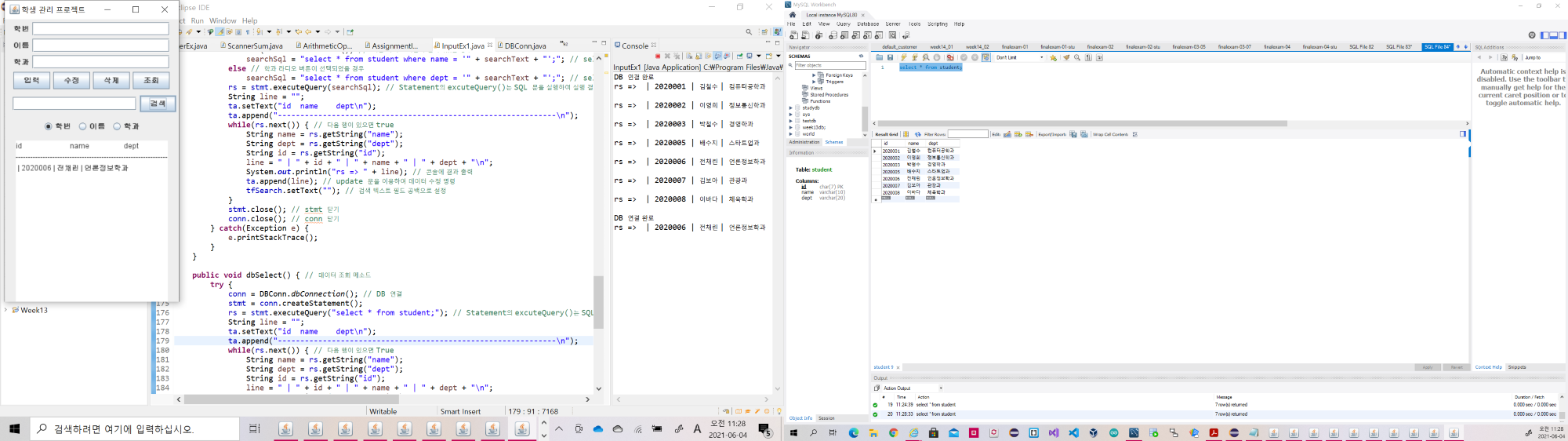


데이터 조회



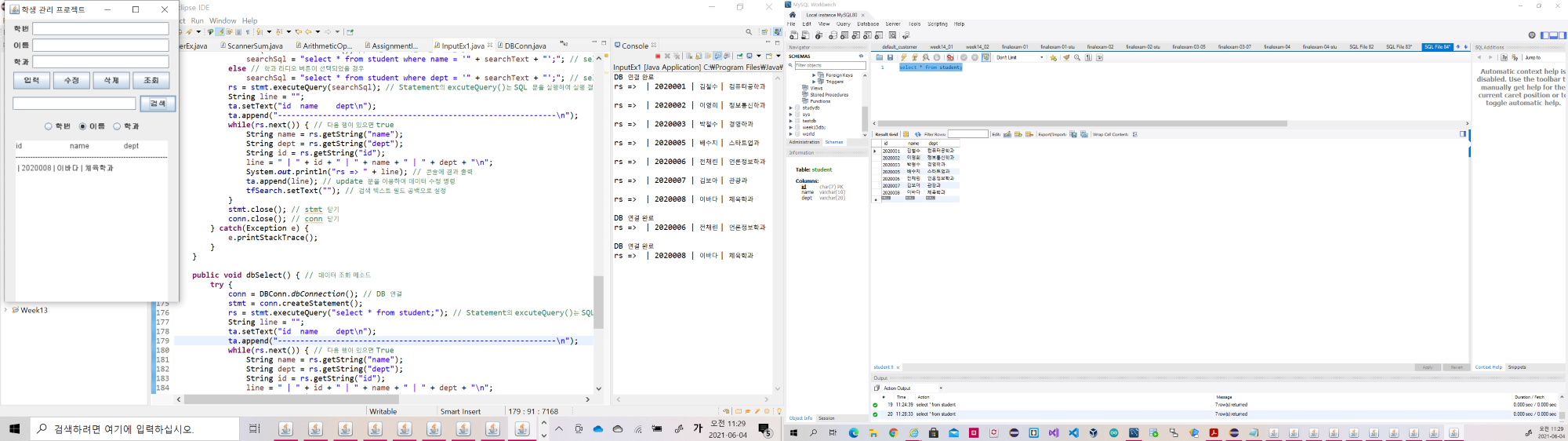
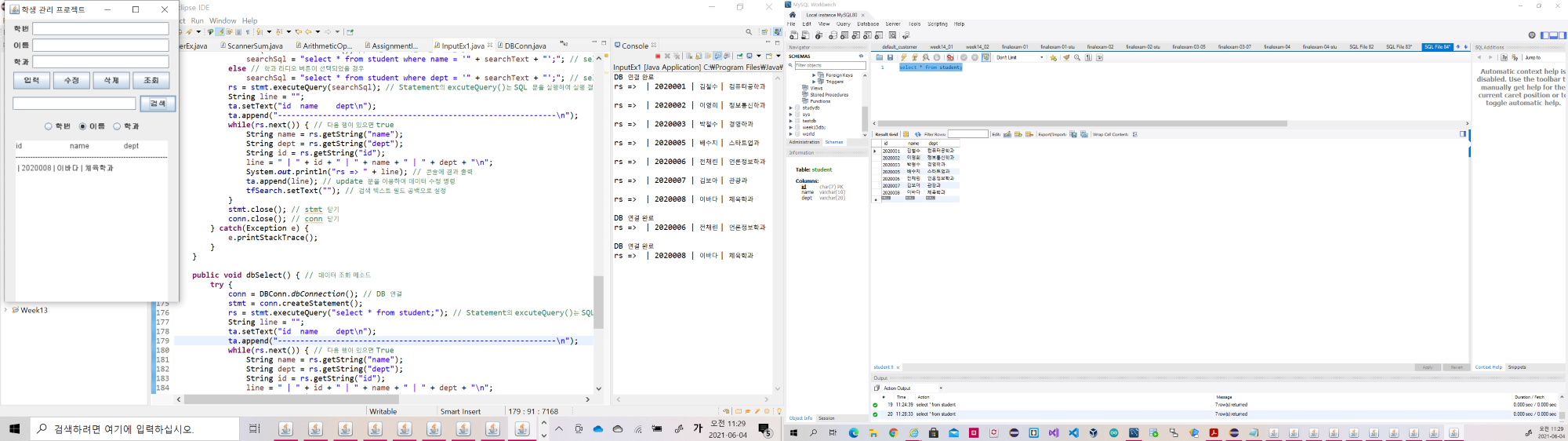
학번으로 데이터 검색

학번이 2020005인 데이터 검색



이름으로 데이터 검색

이름이 이바다인 데이터 검색



학과로 데이터 조회

학과가 스타트업과인 데이터 검색

