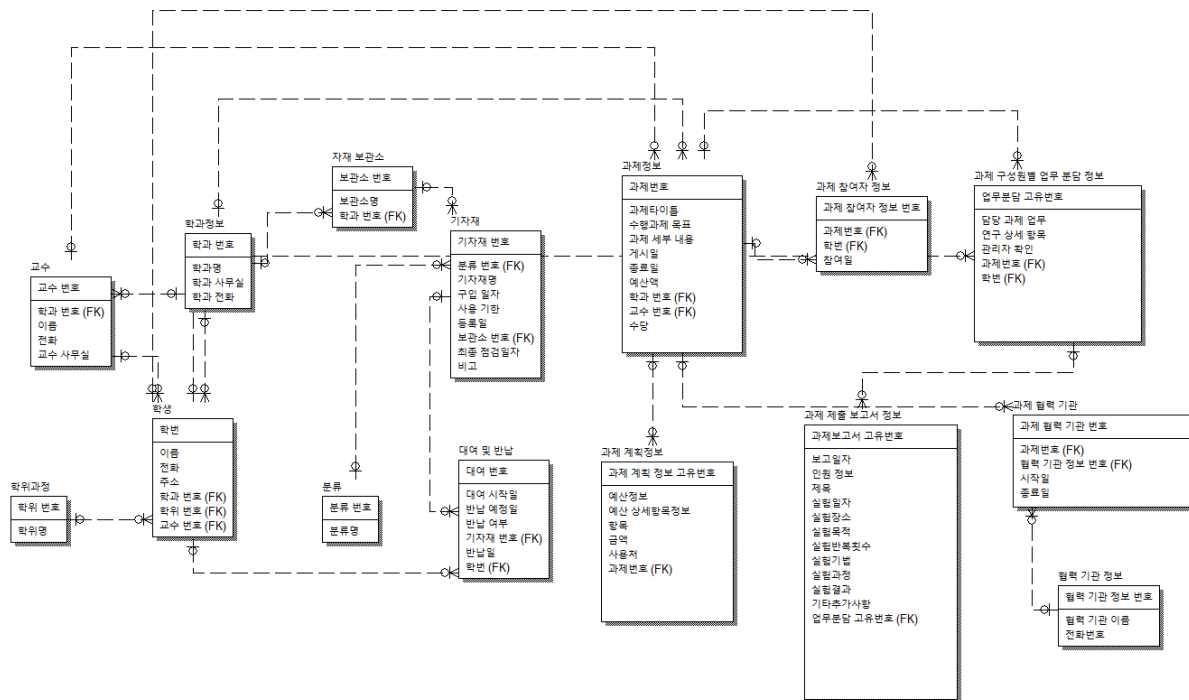


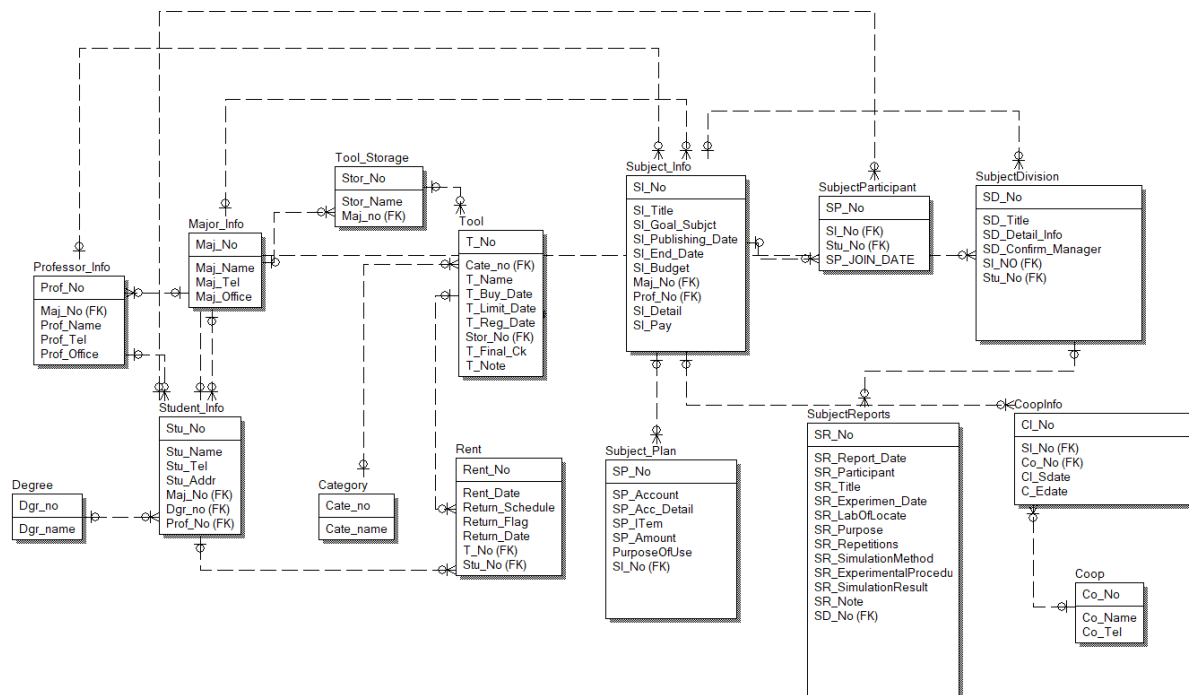
프로젝트 기술서

프로젝트 명	Javara
포트폴리오 사이트주소	https://github.com/hyeoni11i/Project.git
개발 기간	2021.08.25 ~ 2021.09.03
투입인원	5명
주요 기술	<ul style="list-style-type: none">● Eclipse 사용● Java 활용 개발● Oracle 11g 활용 기본 CRUD 구현
리뉴얼 목적	<ul style="list-style-type: none">● 같은 학부 내의 학과 간 연구 정보 공유● 수기, 디지털 작업으로 분리된 문서를 하나의 데이터 베이스에 취합● 학부에서 소유한 기자재의 체계적 관리● 예산 정보, 협력사 정보에 대한 접근 용이
본인의 역할	학부생 / 교직원 관리

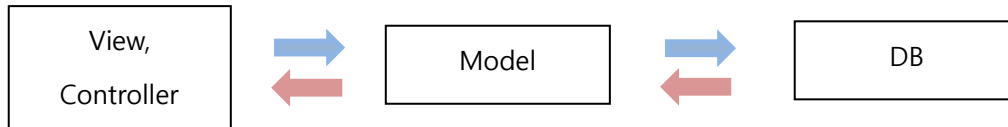
[논리적(Logical) DB 설계]



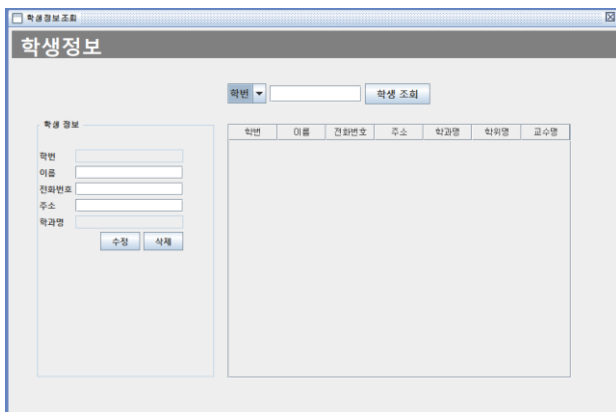
[물리적(Physical) DB 설계]



[구성 및 흐름]



➤ 학생 관리



[View , Controller]

1. 학생 조회 버튼 클릭

```

void eventProc(){
    bStuSearch.addActionListener(this);
    bModify.addActionListener(this);
    bDelete.addActionListener(this);

    stutable.addMouseListener(new MouseAdapter() {
        public void mouseClicked(MouseEvent e) {
            int row = stutable.getSelectedRow();
            int col = 0;
            int sNo = (Integer)stutable.getValueAt(row, col);
            Student stu = new Student();

            try {
                stu = stumodel.selectBySt(sNo);
            } catch (Exception e2) {
                // TODO: handle exception
                JOptionPane.showMessageDialog(null, "학생검색 실패" + e2.getMessage());
            }

            tfStuNo.setText(String.valueOf(stu.getStuno()));
            tfStuName.setText(stu.getStuname());
            tfStuTel.setText(stu.getStutel());
            tfStuAddr.setText(stu.getStuaddr());
            tfMajName.setText(stu.getMajname());
        }
    });
}
  
```

2. 조회된 학생 클릭 시 해당 학생 정보 수정, 삭제 구현

```
public void actionPerformed(ActionEvent ev) {
    Object o = ev.getSource();
    if( o == bStuSearch) {
        selectTable();
        clearScreen();

    }else if (o == bModify) {

        int stuno = Integer.parseInt(tfStuNo.getText());
        String stuname = tfStuName.getText();
        String stutel = tfStuTel.getText();
        String stuaddr = tfStuAddr.getText();

        Student stu = new Student();
        stu.setStuno(stuno);
        stu.setStuname(stuname);
        stu.setStutel(stutel);
        stu.setStuaddr(stuaddr);

        try {
            stumodel.modify(stu);
            JOptionPane.showMessageDialog(null, "수정 성공");
        } catch (Exception e) {
            // TODO: handle exception
            JOptionPane.showMessageDialog(null, "수정 실패" + e.getMessage());
        }
        clearScreen();

    }else if (o == bDelete) {

        int stuno = Integer.parseInt(tfStuNo.getText());

        try {
            stumodel.stuDelete(stuno);
            JOptionPane.showMessageDialog(null, "삭제 성공");
        } catch (Exception e1) {
            // TODO: handle exception
            JOptionPane.showMessageDialog(null, "삭제 실패" + e1.getMessage());
        }
        clearScreen();
    }
}
```


[Model]

1. 학생 조회 (학번 또는 이름)

```
public ArrayList searchStu(int sel, String text) throws Exception{

    String[] selCol = {"s.stu_no = " + text, "s.stu_name = '" + text + "'"};

    String sql = "SELECT s.stu_no, s.stu_name, s.stu_tel, s.stu_addr, m.maj_name, d.dgr_name, p.prof_name "
        + "FROM student_info s, major_info m, professor_info p, degree d "
        + "where s.maj_no=m.maj_no and p.prof_no=s.prof_no and d.dgr_no=s.dgr_no "
        + "and "+ selCol[sel];

    Statement stmt = con.createStatement();
    ResultSet rs = stmt.executeQuery(sql);

    ArrayList list = new ArrayList();

    while (rs.next()) {
        ArrayList temp = new ArrayList();

        temp.add(rs.getInt("stu_no"));
        temp.add(rs.getString("stu_name"));
        temp.add(rs.getString("stu_tel"));
        temp.add(rs.getString("stu_addr"));
        temp.add(rs.getString("maj_name"));
        temp.add(rs.getString("dgr_name"));
        temp.add(rs.getString("prof_name"));

        list.add(temp);
    }

    rs.close();
    stmt.close();

    return list;
}
```

2. 조회된 학생 선택

```
public Student selectBySt( int sNo ) throws Exception{
    Student stu = new Student();
    String sql = "SELECT s.stu_no, s.stu_name, s.stu_tel, s.stu_addr, m.maj_name FROM student_info s, major_info m "
        + "where s.maj_no=m.maj_no and stu_no=" + sNo;

    Statement stmt = con.createStatement();
    ResultSet rs = stmt.executeQuery(sql);

    if(rs.next()) {
        stu.setStuno(rs.getInt("STU_NO"));
        stu.setStuname(rs.getString("STU_NAME"));
        stu.setStutel(rs.getString("STU_TEL"));
        stu.setStuaddr(rs.getString("STU_ADDR"));
        stu.setMajname(rs.getString("MAJ_NAME"));
    }
    rs.close();
    stmt.close();

    return stu;
}
```

[Model]

1. 학생 삭제

```
public void stuDelete(int stuno) throws Exception{
    String sql = "delete student_info where stu_no=" + stuno;
    Statement st = con.createStatement();

    st.executeUpdate(sql);
    st.close();
}
```

2. 조회된 학생 정보 수정

```
public void modify(Object obj) {
    // TODO Auto-generated method stub
    Student stu = (Student) obj;

    try {
        String sql = "UPDATE student_info SET stu_name=?, stu_tel=?, stu_addr=? where stu_no=?";

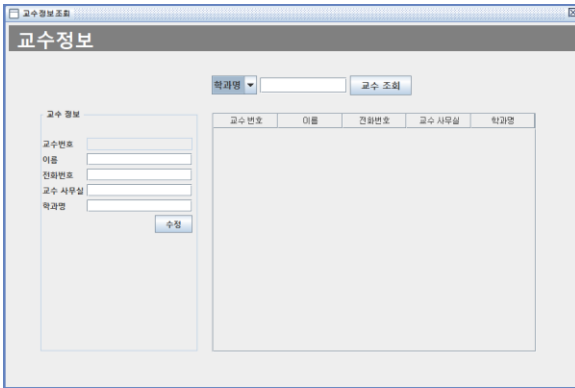
        PreparedStatement ps = con.prepareStatement(sql);

        ps.setString(1, stu.getStuname());
        ps.setString(2, stu.getStutel());
        ps.setString(3, stu.getStuaddr());
        ps.setInt(4, stu.getStuno());

        ps.execute();
        ps.close();
    } catch (Exception e) {
        // TODO: handle exception
    }
}
```

➤ 교직원 관리

[View, Controller]



1. 교직원 정보 조회

```
void eventProc(){
    bProfSearch.addActionListener(this);
    bModify.addActionListener(this);

    tableProf.addMouseListener(new MouseAdapter() {
        public void mouseClicked(MouseEvent e) {
            int row = tableProf.getSelectedRow();
            int col = 0;
            int pNo = (Integer)tableProf.getValueAt(row, col);
            Professor prof = new Professor();

            try {
                prof = profmodel.selectByPf(pNo);
            } catch (Exception e2) {
                // TODO: handle exception
                JOptionPane.showMessageDialog(null, "교수검색 실패" + e2.getMessage());
            }

            tfProfNo.setText(String.valueOf(prof.getProfno()));
            tfProfName.setText(prof.getProfname());
            tfProfTel.setText(prof.getProftel());
            tfProfOffice.setText(prof.getProfoffice());
            tfMajName.setText(prof.getMajname());

            tfMajName.setEditable(false);
        }
    });
}
```


2. 교직원 정보 수정

```
public void actionPerformed(ActionEvent ev) {
    Object o = ev.getSource();
    if( o == bProfSearch) {
        selectTable();
        clearScreen();
    }else if (o == bModify) {

        int profno = Integer.parseInt(tfProfNo.getText());
        String profname = tfProfName.getText();
        String proftel = tfProfTel.getText();
        String profoffice = tfProfOffice.getText();
        String majname = tfMajName.getText();

        Professor prof = new Professor();
        prof.setProfno(profno);
        prof.setProfname(profname);
        prof.setProftel(proftel);
        prof.setProfoffice(profoffice);
        prof.setMajname(majname);

        try {
            profmodel.modify(prof);
            JOptionPane.showMessageDialog(null, "수정 성공");
        } catch (Exception e) {
            // TODO: handle exception
            JOptionPane.showMessageDialog(null, "수정 실패" + e.getMessage());
        }
        clearScreen();
    }
}
```

[Model]

1. 교직원 조회

```
public ArrayList searchProf(int sel, String text) throws Exception{

    String[] selcol = {"m.maj_name like '%" + text + "%'", "p.prof_name = '" + text + "'"};

    String sql = "SELECT p.prof_no, p.prof_name, p.prof_tel, p.prof_office, m.maj_name FROM professor_info p, major_info m "
        + "where p.maj_no=m.maj_no "
        + "and " + selcol[sel];
    PreparedStatement ps = con.prepareStatement(sql);
    ResultSet rs = ps.executeQuery(sql);

    ArrayList list = new ArrayList();

    while (rs.next()) {
        ArrayList temp = new ArrayList();

        temp.add(rs.getInt("prof_no"));
        temp.add(rs.getString("prof_name"));
        temp.add(rs.getString("prof_tel"));
        temp.add(rs.getString("prof_office"));
        temp.add(rs.getString("maj_name"));

        list.add(temp);
    }
    rs.close();
    ps.close();

    return list;
}
```

[Model]

2. 조회된 교직원 선택

```
public Professor selectByPf( int pNo ) throws Exception{
    Professor prof = new Professor();
    String sql = "SELECT p.prof_no, p.prof_name, p.prof_tel, p.prof_office, m.maj_name FROM professor_info p, major_info m "
        + "where p.maj_no=m.maj_no and prof_no=" + pNo;
    Statement stmt = con.createStatement();
    ResultSet rs = stmt.executeQuery(sql);
    if(rs.next()) {
        prof.setProfno(rs.getInt("PROF_NO"));
        prof.setProfname(rs.getString("PROF_NAME"));
        prof.setProftel(rs.getString("PROF_TEL"));
        prof.setProfoffice(rs.getString("PROF_OFFICE"));
        prof.setMajname(rs.getString("MAJ_NAME"));
    }
    rs.close();
    stmt.close();

    return prof;
}
```

3. 교직원 정보 수정

```
public void modify(Object obj) {
    // TODO Auto-generated method stub
    Professor prof = (Professor) obj;

    try {
        String sql = "UPDATE professor_info SET prof_name=?, prof_tel=?, prof_office=? where prof_no=?";

        PreparedStatement ps = con.prepareStatement(sql);

        ps.setString(1, prof.getProfname());
        ps.setString(2, prof.getProftel());
        ps.setString(3, prof.getProfoffice());
        ps.setInt(4, prof.getProfno());
        ps.execute();
        ps.close();
    }catch (Exception e) {
        // TODO: handle exception
    }
}
```