

REPORT



제출일자 :

2023/11/06

담당교수 :

황수찬 교수님

과목 :

데이터베이스 기초

학부(과) :

소프트웨어학과

팀번호 :

팀 130

과제번호 :

HW4 (JDBC)

데이터베이스기초 JDBC 실습 보고서

130팀

소프트웨어학과 2019125021 남서아

물류전공 2019123097 허진혁

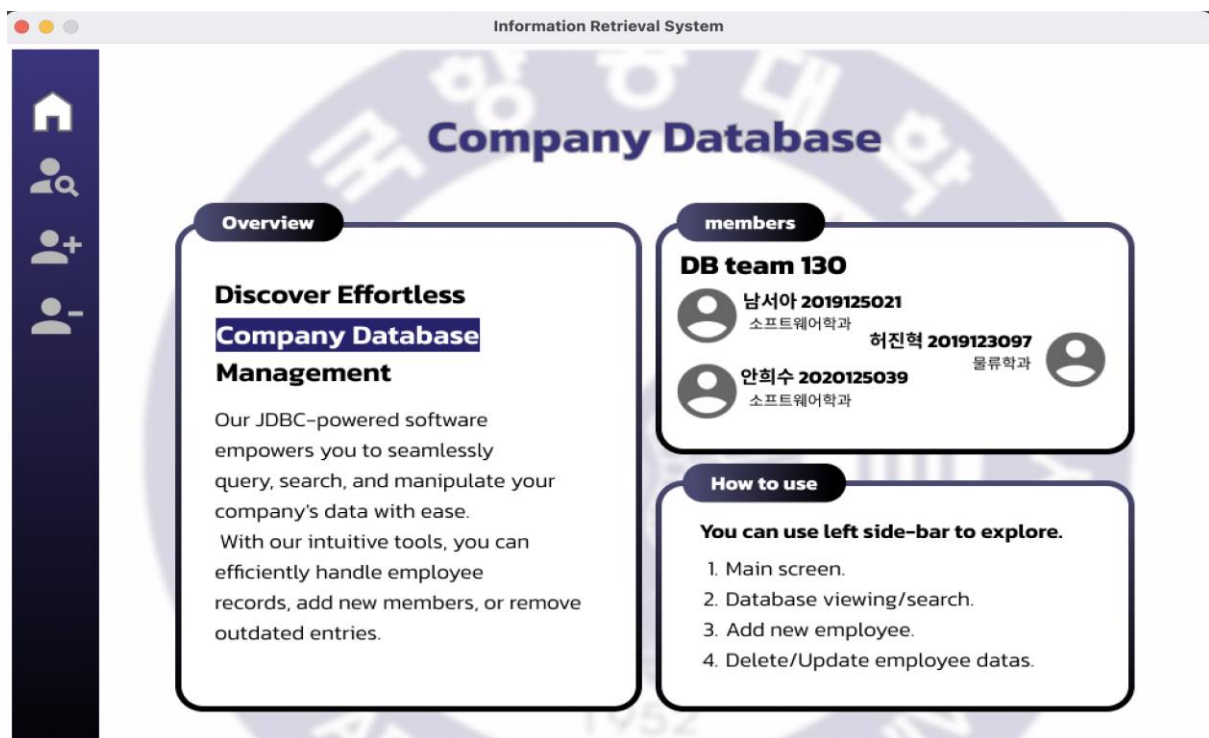
소프트웨어학과 2020125039 안희수

1. 문제 및 프로그램 설명

프로그램은 총 4개의 네비게이션 바로 구성되어있다.

1) Main screen

첫번째 사이드바에 해당하는 페이지는 우리가 만든 직원 검색 시스템의 기본 정보를 나타낸다. Overview에는 시스템에서 할 수 있는 기능과 이점에 대한 내용이 나와있다. Members 항목에는 각 팀원들의 정보가 기재되어 있다. How to use에는 각 사이드바에서 할 수 있는 기능을 나타냈다. 이는 과제에서 제시된 기능은 아니나, 처음 시작 페이지를 보여준다.



2) Database viewing / search

두번째 사이드바에 해당하는 내용은 보고서 출력과 조건 검색을 할 수 있는 창이다. 해당 탭에서는 과제의 1번과 2번 항목에 해당하는 기능이 구현되어 있다.

Range에 default 값으로 All을 넣어두고, 검색 조건 체크박스는 모두 True로 만들어두었다. 기본으로 설정된 화면 그대로 Search 버튼을 누르면 EMPLOYEE 테이블의 모든 직원의 정보를 출력할 수 있다.

The screenshot shows a web application window titled "Information Retrieval System". On the left is a dark blue sidebar with three icons: a house, a person with a magnifying glass, and a person with a plus sign. The main area has a dark blue header with a "Range" dropdown set to "All". Below the header is a row of checkboxes for search criteria: Name, Ssn, Bdate, Address, Sex, Salary, Supervisor, and Department, all of which are checked. A "Search" button is on the right. The main content area is a large, empty white box.

Range는 아래와 같이 콤보박스에서 선택할 수 있다.

This image shows a close-up of the "Range" dropdown menu. The menu is open, displaying a list of items: Name, Ssn, Bdate, Address, Sex, Salary, Supervisor, and Department. The "Sex" item is currently selected and highlighted. To the right of the dropdown is a separate box showing the selected value "M" and a list of available options: "M" and "F".

3) Add new employee

세번째 사이드바는 새로운 직원을 추가할 수 있는 창이다. 해당하는 값을 입력하고 Confirm 창을 누르면 새로운 직원이 추가된다. 해당 탭에서는 과제의 4번 항목에 해당하는 기능이 구현되어 있다.

Information Retrieval System

새로운 직원 정보 추가

First Name:	<input type="text"/>	Address:	<input type="text"/>
Middle Initial:	<input type="text"/>	Sex:	<input type="text" value="female"/>
Last Name:	<input type="text"/>	Salary:	<input type="text"/>
Ssn:	<input type="text"/>	Super_ssn:	<input type="text"/>
Birthdate:	<input type="text"/>	Dno:	<input type="text"/>

Confirm

4) Delete / Update employee datas

네번째 사이드바는 선택된 직원을 삭제하거나 입력받은 값으로 업데이트하는 기능을 수행하는 창이다. 해당 탭에서는 과제의 3번 항목에 해당하는 기능이 구현되어 있다.

직원 정보를 삭제하고 싶을 때는 삭제를 원하는 직원을 check 해주고 오른쪽 아래의 빨간색 삭제 버튼을 눌러주면 된다. 직원 정보를 업데이트할 때는 업데이트 하고싶은 직원을 체크하고 오른쪽 아래의 수정 항목에 값을 입력하여 초록색 업데이트 버튼을 누르면 업데이트 된다. 해당 버튼들에는 update와 delete를 상징하는 아이콘(위쪽 화살표, 휴지통)을 사용하였다.

delete 기능만이 원래 과제에 제시된 요구사항이나, update 기능을 추가로 구현하였다.

Information Retrieval System

Home

Search

+

-

Range

All

Search

Item

☒ Name

☒ Ssn

☒ Bdate

☒ Address

☒ Sex

☒ Salary

☒ Supervisor

☒ Department

Name	Ssn	Bdate	Address	Sex	Salary	Supervisor	Dname	check
John B Smith	123456789	1965-01-09	731 Fondren, Ho...	M	30000.00	Franklin T Wong	Research	<input type="checkbox"/>
Franklin T Wong	333445555	1955-12-08	638 Voss, Hous...	M	40000.00	James E Borg	Research	<input type="checkbox"/>
Joyce A English	453453453	1972-07-31	5631 Rice, Hous...	F	25000.00	Franklin T Wong	Research	<input type="checkbox"/>
Ramesh K Nara...	666884444	1962-09-15	975 Fire Oak, H...	M	38000.00	Franklin T Wong	Research	<input type="checkbox"/>
James E Borg	888665555	1937-11-10	450 Stone, Hous...	M	55000.00	NULL	Headquarters	<input type="checkbox"/>
Jennifer S Walla...	987654321	1941-06-20	291 Berry, Bellai...	F	43000.00	James E Borg	Administration	<input type="checkbox"/>
Ahmad V Jabbar	987987987	1969-03-29	980 Dallas, Hous...	M	25000.00	Jennifer S Walla...	Administration	<input type="checkbox"/>
Alicia J Zelaya	999887777	1968-01-19	3321 Castle, Sp...	F	25000.00	Jennifer S Walla...	Administration	<input type="checkbox"/>

선택한 직원 :

수정 : Name

↑

↓

인원 수 :

2. 알고리즘 및 분석

※ mysql connection 부분

```
public static void connectDB() throws SQLException{
    try
    {
        String user = "root";
        String pwd = "root"; // 비밀번호 입력
        String dbname = "company"; // 스키마 이름 입력
        String url = "jdbc:mysql://localhost:3306/" + dbname +
"?serverTimezone=UTC";
        conn = DriverManager.getConnection(url, user, pwd);
        System.out.println("정상적으로 연결되었습니다.");
    }
    catch (SQLException e1) {
        System.err.println("연결할 수 없습니다.");
        e1.printStackTrace();
        // SQL 예외가 발생했을 때 예외 처리
        String errorMessage = "SQL 오류:" + e1.getMessage();
        JOptionPane.showMessageDialog(null, errorMessage, "SQL Error",
JOptionPane.ERROR_MESSAGE);
    }
}
```

해당 코드는 mysql과 jdbc를 연결하기 위한 부분이다. user 변수에는 설정해둔 user 명을, pwd 변수에는 설정해둔 비밀번호를, dbname에는 만들어둔 스키마의 변수명을 넣어야한다.

이때, 만일 해당 정보가 올바르지 않아 sql exception이 발생할 경우, 그에 맞는 에러 창을 띄워서 보여 준다.

1) 보고서 출력

2) 조건 검색

```
searchButton.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        if (e.getSource() == searchButton) {
            String stmt = "SELECT ";
            if (nameCheckBox.isSelected()) {
                stmt += "concat(e.fname, ' ', e.minit, ' ', e.lname, ' ')
as Name";
            }
            if (ssnCheckBox.isSelected()) {
                if (!stmt.equals("SELECT ")) {
                    stmt += ", ";
                }
                stmt += "e.ssn";
            }
            if (bdateCheckBox.isSelected()) {
                if (!stmt.equals("SELECT ")) {
                    stmt += ", ";
                }
                stmt += "e.bdate";
            }
            if (addressCheckBox.isSelected()) {
                if (!stmt.equals("SELECT ")) {
                    stmt += ", ";
                }
                stmt += "e.address";
            }
            if (sexCheckBox.isSelected()) {
                if (!stmt.equals("SELECT ")) {
                    stmt += ", ";
                }
                stmt += "e.sex";
            }
            if (salaryCheckBox.isSelected()) {
                if (!stmt.equals("SELECT ")) {
                    stmt += ", ";
                }
                stmt += "e.salary";
            }
            if (supervisorCheckBox.isSelected()) {
                if (!stmt.equals("SELECT ")) {
                    stmt += ", ";
                }
                stmt += "concat(s.fname, ' ', s.minit, ' ', s.lname, ' ')
```

```

as Supervisor";
    }
    if (departmentCheckBox.isSelected()) {
        if (!stmt.equals("SELECT ")) {
            stmt += ", ";
        }
        stmt += "dname";
    }
    stmt += " FROM employee e LEFT OUTER JOIN employee s ON
e.super_ssn = s.ssn, department WHERE e.dno = dnumber";
    if
(searchItemComboBox.getSelectedItem().toString().equals("Name")) {
        String fname = searchTextField.getText();
        String[] nameParts = fname.split(" ");
        if (nameParts.length == 1) {
            // 단일 이름으로 검색
            stmt += " and (e.fname = '" + nameParts[0] + "' OR
e.lname = '" + nameParts[0] + "' OR e.minit = '" + nameParts[0] + "')";
        } else if (nameParts.length == 2) {
            // 이름과 성으로 검색
            stmt += " and (e.fname = '" + nameParts[0] + "' AND
e.minit = '" + nameParts[1] + "')";
        } else if (nameParts.length == 3) {
            // 이름, 중간 이름, 성으로 검색
            stmt += " and (e.fname = '" + nameParts[0] + "' AND
e.minit = '" + nameParts[1] + "' AND e.lname = '" + nameParts[2] + "')";
        }
    }
    if (searchItemComboBox.getSelectedItem().toString() ==
("Supervisor (Ssn)")) {
        String fssn = searchTextField.getText();
        stmt += " and e.ssn = '" + fssn + "'";
    }
    if (searchItemComboBox.getSelectedItem().toString() ==
("Bdate")) {
        String fbdate = searchTextField.getText();
        //비워져있으면 NoResult
        if (fbdate.isEmpty()) {
            fbdate = "error";
        }
        stmt += " and e.bdate like '%" + fbdate + "%'";
    }
    if (searchItemComboBox.getSelectedItem().toString() ==
("Address")) {
        String faddress = searchTextField.getText();
        //비워져있으면 NoResult
        if (faddress.isEmpty()) {
            faddress = "error";
        }
        stmt += " and e.address like '%" + faddress + "%'";
    }
    if (searchItemComboBox.getSelectedItem().toString() == ("Sex"))
{
        if (searchComboBox.getSelectedItem().toString() == ("M")) {
            String fsex = "M";
            stmt += " and e.sex = '" + fsex + "'";
        }
        else if (searchComboBox.getSelectedItem().toString() ==
("F")) {
            String fsex = "F";

```

```

        stmt += " and e.sex = '" + fsex + "'";
    }
}
if      (searchItemComboBox.getSelectedItem().toString() ==
("Salary")) {
    String fsalary = searchTextField.getText();
    //숫자로 이뤄져있는지 판단
    if (!fsalary.matches("^\\d+$")) {
        fsalary = "9999999999999999";
    }
    stmt += " and e.salary >= '" + fsalary + "'";
}
if      (searchItemComboBox.getSelectedItem().toString() ==
("Supervisor")) {
    String fsupervisor_ssn = searchTextField.getText();
    stmt += " and e.super_ssn = '" + fsupervisor_ssn + "'";
}
if      (searchItemComboBox.getSelectedItem().toString() ==
("Department")) {
    if (searchComboBox.getSelectedItem().toString() ==
("Research")) {
        String fdepartment = "Research";
        stmt += " and dname = '" + fdepartment + "'";
    }
    else if (searchComboBox.getSelectedItem().toString() ==
("Administration")) {
        String fdepartment = "Administration";
        stmt += " and dname = '" + fdepartment + "'";
    }
    else if (searchComboBox.getSelectedItem().toString() ==
("Headquarters")) {
        String fdepartment = "Headquarters";
        stmt += " and dname = '" + fdepartment + "'";
    }
}
try {
    connectDB();
    ResultSet rs = rs(conn, stmt);
    ResultSetMetaData rsmd = rs.getMetaData();
    int cols = rsmd.getColumnCount();
    noresults.setVisible(false);
    tableModel.setRowCount(0); // Clear existing data
    tableModel.setColumnCount(0); // Clear existing headers
    // Add the column names to the tableModel
    for (int i = 1; i <= rsmd.getColumnCount(); i++) {
        tableModel.addColumn(rsmd.getColumnName(i));
    }
    // Populate the tableModel with data
    while (rs.next()) {
        String[] rowData = new String[cols];
        for (int i = 1; i <= cols; i++) {
            Object columnValue = rs.getObject(i);
            rowData[i - 1] = columnValue != null ?
columnValue.toString() : "NULL";
            if (columnValue != null) {
                System.out.printf(rsmd.getColumnName(i) + " : "+ "%-20s\t", columnValue.toString());
            } else {
                System.out.printf("%-20s\t", "NULL");
            }
        }
    }
}

```



```

        System.out.println();
        tableModel.addRow(rowData);
    }
    tableModel.addColumn("check");
    tableModel.fireTableDataChanged();
    tableModel.fireTableStructureChanged();
    TableColumn checkboxColumn =
employeeTable.getColumnModel().getColumn(cols); // 0번째 열 (왼쪽 열)에 체크박스 추가
checkboxColumn.setCellRenderer(new CheckboxRenderer());
checkboxColumn.setCellEditor(new CheckboxEditor());
// Create the JTable using the tableModel
//employeeTable.setAutoResizeMode(JTable.AUTO_RESIZE_OFF);
if (tableModel.getRowCount() <= 0 && noresults.isVisible()
== false) {
    noresults.setVisible(true);
} else {
    noresults.setVisible(false);
}
} catch (SQLException e1) {
    e1.printStackTrace();
}
System.out.println(stmt);
    }
}
});

```

searchButton이 눌리면 sql문이 SELECT로 고정되고 후에 어떤 CheckBox가 선택되는지 여부에 따라 sql문이 완성된다. name이나 supervisor가 선택되면 db가 가지고 있는 fname, minit, lname을 모두 합쳐서 하나의 이름으로 보여준다.

Range옆의 콤보박스 searchItemComboBox로 검색하고 싶은 attribute를 고르고, searchComboBox에 검색하고자 하는 특정 값을 입력하면 조건에 맞는 값이 출력된다. 이때 name을 고르고 fname, minit, lname 중 하나의 값을 넣어도 검색될 수 있도록 했다.

모든 조건들은 해당하는 값이 없거나 db에 저장된 데이터 형식과 다르면 jdbc에서 값이 없다고 반환 해주어 no results 창이 뜨도록 했다.

3) 조건 삭제

```
deleteBtn.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        Vector<String> delete_ssn = new Vector<String>();
        if (e.getSource() == deleteBtn) {
            try {
                String ColumnName1 = employeeTable.getColumnName(0);
                String ColumnName2 = employeeTable.getColumnName(1);
                if ("Ssn".equals(ColumnName1) || "Ssn".equals(ColumnName2))
                {
                    int rowCount = employeeTable.getRowCount();
                    int colCount = employeeTable.getColumnCount();
                    System.out.println(rowCount);
                    System.out.println(colCount);
                    for (int i = 0; i < rowCount ; i++) {
                        if (Boolean.TRUE.equals(employeeTable.getValueAt(i,
colCount-1))) {
                            if("Ssn".equals(ColumnName2)) {
                                delete_ssn.add((String)
employeeTable.getValueAt(i, 1)); // Ssn 열
System.out.println(employeeTable.getValueAt(i, colCount-1));
                            }else {
                                delete_ssn.add((String)
employeeTable.getValueAt(i, 0)); // Ssn 열
                            }
                        }
                    }
                    System.out.println(delete_ssn);
                    for (int i = 0; i< delete_ssn.size(); i++) {
                        for(int k = 0; k< employeeTable.getRowCount(); k++)
                        {
                            if(employeeTable.getValueAt(k, colCount-1 )==
Boolean.TRUE)
                                tableModel.removeRow(k);
                        }
                    }
                    // 데이터베이스에서도 삭제
                    for (String ssn : delete_ssn) {
                        String deleteStmt = "DELETE FROM EMPLOYEE WHERE
Ssn=?";
                        PreparedStatement p =
conn.prepareStatement(deleteStmt);
                        p.clearParameters();
                        p.setString(1, ssn);
                        p.executeUpdate();
                    }
                    searchButton.doClick();
                }else {
                    System.out.println("삭제가 불가능 합니다. Ssn을 반드시 선택해 주
세요");
                    JOptionPane.showMessageDialog(null, "삭제가 불가능 합니다.
Ssn을 반드시 선택해 주세요.", "Unspecified Behavior", JOptionPane.ERROR_MESSAGE);
                }
            } catch (SQLException e1) {
                e1.printStackTrace();
                String errorMessage = "SQL 오류 : 입력값을 다시 확인해주세요.Wn" +
e1.getMessage();
            }
        }
    }
});
```

```

        JOptionPane.showMessageDialog(null, errorMessage, "SQL
Error", JOptionPane.ERROR_MESSAGE);
    }
}
});
deleteBtn.setDefaultCapable(false);
deleteBtn.setFocusable(false);
deleteBtn.setFocusTraversalKeysEnabled(false);
deleteBtn.setFocusPainted(false);
deleteBtn.setContentAreaFilled(false);
deleteBtn.setBorderPainted(false);
deleteBtn.setSelectedIcon(new ImageIcon("images/delet_hovered.png"));
deleteBtn.setRolloverIcon(new ImageIcon("images/delet_hovered.png"));
deleteBtn.setRequestFocusEnabled(false);
deleteBtn.setIcon(new ImageIcon("images/delete_idle.png"));
deleteBtn.setToolTipText("delete");
updatecomboBox.addItemListener(new ItemListener() {
    public void itemStateChanged(ItemEvent itemEvent) {
        String selected = updatecomboBox.getSelectedItem().toString();
        if (selected.equals("Department") || selected.equals("Sex")) {
            update_input_text.setVisible(false);
            update_input_combo.setVisible(true);
            if (selected.equals("Sex")) {
                update_input_combo.setModel(new
DefaultComboBoxModel<String>(new String[]{"M", "F"}));
            }
            else if (selected.equals("Department")) {
                String[] departmentOptions = {"Research", "Administration",
"Headquarters"};
                update_input_combo.setModel(new
DefaultComboBoxModel<String>(departmentOptions));
            }
        } else {
            update_input_text.setVisible(true);
            update_input_combo.setVisible(false);
        }
    }
});

```

삭제는 primary key인 ssn을 기준으로 진행한다. 사용자가 조건 검색을 ssn 없이 하면 삭제될 수 없도록 구현해놓았다. ssn이 선택되지 않은 상황에서 삭제를 하면 해당 직원에 대한 정보가 충분하지 않은 상태라 문제가 생길 수 있겠다고 생각했기 때문이다. employeeTable의 모든 행을 돌며 체크박스가 true인지를 확인하고 delete_ssn 문자열 벡터에 삭제할 ssn 값을 저장한다. 해당 ssn에 해당하는 직원을 데이터베이스에서 삭제한다.

4) 직원 추가

```
confirmBtn.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        String input_fname = fname_input.getText() , input_minit =
minit_input.getText() , input_lname = lname_input.getText() ,
        input_ssn = ssn_input.getText() , input_bdate =
bdate_input.getText() , input_address = add_input.getText() ,
        input_salary = salary_input.getText() , input_superssn =
superSsn_input.getText() , input_dno = dno_input.getText() ,
        input_sex =
sex_input.getSelectedItem().toString().equals("female") ? "F" : "M";
        Timestamp timestamp = new Timestamp(System.currentTimeMillis());
        SimpleDateFormat sdf = new SimpleDateFormat ("yyyy-MM-dd hh:mm:ss");
        String stmt1 = "Insert into employee values
(?,?,?,?,?,?,?,?,?,?,?)";
        try {
            connectDB();
            PreparedStatement p = conn.prepareStatement(stmt1);
            p.clearParameters();
            p.setString(1, input_fname);
            p.setString(2, input_minit);
            p.setString(3, input_lname);
            p.setString(4, input_ssn);
            p.setString(5, input_bdate);
            p.setString(6, input_address);
            p.setString(7, input_sex);
            p.setString(8, input_salary);
            p.setString(9, input_superssn);
            p.setString(10, input_dno);
            p.setString(11, sdf.format(timestamp));
            p.setString(12, sdf.format(timestamp));
            p.executeUpdate();
            // clear inputs
            fname_input.setText("");
            minit_input.setText("");
            lname_input.setText("");
            ssn_input.setText("");
            bdate_input.setText("");
            add_input.setText("");
            salary_input.setText("");
            superSsn_input.setText("");
            dno_input.setText("");
            sex_input.setSelectedIndex(-1);
            p.close();
            JOptionPane.showMessageDialog(null, "성공적으로 데이터가 추가되었습니다!", "message", JOptionPane.INFORMATION_MESSAGE);
        } catch (SQLException ex) {
            ex.printStackTrace();
            String errorMessage = "SQL 오류 : 입력값을 다시 확인해주세요.\n" +
ex.getMessage();
            JOptionPane.showMessageDialog(null, errorMessage, "SQL Error",
JOptionPane.ERROR_MESSAGE);
        }
    }
});
```

직원 추가 탭에 있는 버튼의 이벤트 핸들러에 sql 쿼리 실행 기능을 추가하였다.

해당 버튼을 누를 시, 먼저 현재 input textfield에 설정된 값들을 가져온다. 그리고 sql statement의 각 변수 자리에 해당 변수들을 mapping 시킨다. 그리고 Insert를 실행한다. 이때, 입력 데이터의 문제로 인해 sql exception이 발생할 경우, 에러창을 띄워 해당 exception의 에러 메시지를 보여준다. 반면, 성공적으로 insert가 이뤄질 경우 성공적으로 데이터가 추가되었다는 창을 띄워준다. 동시에 현재 Input field들의 값을 clear 해준다.

5) 조건 업데이트 (추가기능)

```
updateBtn.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        if (e.getSource() == updateBtn) {
            String ColumnName1 = employeeTable.getColumnName(0);
            String ColumnName2 = employeeTable.getColumnName(1);
            if ("Ssn".equals(ColumnName1) || "Ssn".equals(ColumnName2)) {
                try {
                    int rowCount = employeeTable.getRowCount();
                    int colCount = employeeTable.getColumnCount();
                    int checkboxColumn = colCount - 1;
                    int ColumnName = 0, ColumnSsn = 0, ColumnBdate = 0,
                    ColumnAddress = 0, ColumnSex = 0, ColumnSalary = 0, ColumnDepartment = 0;
                    for (int i = 0; i < colCount; i++) {
                        if ("Name".equals(employeeTable.getColumnName(i)))
                        {
                            ColumnName = i;
                        }
                        else if ("Ssn".equals(employeeTable.getColumnName(i))) {
                            ColumnSsn = i;
                        }
                        else if ("Bdate".equals(employeeTable.getColumnName(i))) {
                            ColumnBdate = i;
                        }
                        else if ("Address".equals(employeeTable.getColumnName(i))) {
                            ColumnAddress = i;
                        }
                        else if ("Sex".equals(employeeTable.getColumnName(i))) {
                            ColumnSex = i;
                        }
                        else if ("Salary".equals(employeeTable.getColumnName(i))) {
                            ColumnSalary = i;
                        }
                        else if ("Dname".equals(employeeTable.getColumnName(i))) {
                            ColumnDepartment = i;
                        }
                    }
                    // 콤보박스과 업데이트 값 가져오기
                    String selectedUpdateColumn = (String)
                    updatecomboBox.getSelectedItemAt();
                    String selectedUpdateInput = (String)
                    update_input_combo.getSelectedItemAt();
                    String updatedValue = update_input_text.getText();
                    if ("Name".equals(selectedUpdateColumn)) {
                        for (int i = 0; i < rowCount; i++) {
```

```

        if
        (Boolean.TRUE.equals(employeeTable.getValueAt(i, checkboxColumn))) {
            // Ssn 값을 문자열로 가져옴
            String ssn = (String)
employeeTable.getValueAt(i, ColumnSsn);
            // Name을 Fname, Minit, Lname으로 분리
            String[] nameParts = updatedValue.split("
");
            if (nameParts.length >= 3) {
                String fname = nameParts[0];
                String minit = nameParts[1];
                String lname = nameParts[2];
                employeeTable.setValueAt(updatedValue,
i, ColumnName);

                // 데이터베이스 업데이트
                String updateSQL = "UPDATE EMPLOYEE SET
Fname = ?, Minit = ?, Lname = ? WHERE Ssn = ?";
                PreparedStatement p =
conn.prepareStatement(updateSQL);

                p.setString(1, fname);
                p.setString(2, minit);
                p.setString(3, lname);
                p.setString(4, ssn);
                p.executeUpdate();
                JOptionPane.showMessageDialog(null, "성
공적으로 값이 업데이트 되었습니다", "message", JOptionPane.INFORMATION_MESSAGE);
                System.out.println("db수정 완료");
            }
            else {
                System.out.println("Fname, Minit, Lname
을 모두 입력해주세요.");
                JOptionPane.showMessageDialog(null,
"Fname, Minit, Lname 을 모두 입력해주세요.", "Unspecified Behavior",
JOptionPane.ERROR_MESSAGE);
            }
        }
        searchButton.doClick();
    } else if ("Ssn".equals(selectedUpdateColumn)) {
        for (int i = 0; i < rowCount; i++) {
            if
            (Boolean.TRUE.equals(employeeTable.getValueAt(i, checkboxColumn))) {
                // Ssn 값을 문자열로 가져옴
                String ssn = (String)
employeeTable.getValueAt(i, ColumnSsn);
                try {
                    // 데이터베이스 업데이트
                    String updateSQL = "UPDATE EMPLOYEE SET
Ssn = ? WHERE Ssn = ?";
                    PreparedStatement p =
conn.prepareStatement(updateSQL);

                    p.setString(1, updatedValue);
                    p.setString(2, ssn);
                    p.executeUpdate();
                    JOptionPane.showMessageDialog(null, "성
공적으로 값이 업데이트 되었습니다", "message", JOptionPane.INFORMATION_MESSAGE);
                    System.out.println("db수정 완료");
                    employeeTable.setValueAt(updatedValue,
i, ColumnSsn);

```

```

        } catch (SQLException e1) {
            e1.printStackTrace();
            String errorMessage = "SQL 오류 : " +
e1.getMessage();
            JOptionPane.showMessageDialog(null,
errorMessage, "SQL Error", JOptionPane.ERROR_MESSAGE);
        }
    }
    searchButton.doClick();
} else if ("Bdate".equals(selectedUpdateColumn)) {
    for (int i = 0; i < rowCount; i++) {
        if
(Boolean.TRUE.equals(employeeTable.getValueAt(i, checkboxColumn))) {
            // Ssn 값을 문자열로 가져옴
            String ssn = (String)
employeeTable.getValueAt(i, ColumnSsn);
            // Bdate 값을 업데이트
            employeeTable.setValueAt(updatedValue, i,
ColumnBdate);
            // 데이터베이스 업데이트
            String updateSQL = "UPDATE EMPLOYEE SET
Bdate = ? WHERE Ssn = ?";
            PreparedStatement p =
conn.prepareStatement(updateSQL);
            p.setString(1, updatedValue);
            p.setString(2, ssn);
            p.executeUpdate();
            JOptionPane.showMessageDialog(null, "성공적
으로 값이 업데이트 되었습니다", "message", JOptionPane.INFORMATION_MESSAGE);
            System.out.println("db수정 완료");
        }
    }
    searchButton.doClick();
} else if ("Address".equals(selectedUpdateColumn)) {
    for (int i = 0; i < rowCount; i++) {
        if
(Boolean.TRUE.equals(employeeTable.getValueAt(i, checkboxColumn))) {
            // Ssn 값을 문자열로 가져옴
            String ssn = (String)
employeeTable.getValueAt(i, ColumnSsn);
            // Address 값을 업데이트
            employeeTable.setValueAt(updatedValue, i,
ColumnAddress);
            // 데이터베이스 업데이트
            String updateSQL = "UPDATE EMPLOYEE SET
Address = ? WHERE Ssn = ?";
            PreparedStatement p =
conn.prepareStatement(updateSQL);
            p.setString(1, updatedValue);
            p.setString(2, ssn);
            p.executeUpdate();
            JOptionPane.showMessageDialog(null, "성공적
으로 값이 업데이트 되었습니다", "message", JOptionPane.INFORMATION_MESSAGE);
            System.out.println("db수정 완료");
        }
    }
    searchButton.doClick();
} else if ("Sex".equals(selectedUpdateColumn)) {

```

```

        for (int i = 0; i < rowCount; i++) {
            if
(Boolean.TRUE.equals(employeeTable.getValueAt(i, checkBoxColumn))) {
                // Ssn 값을 문자열로 가져옴
                String ssn = (String)
employeeTable.getValueAt(i, ColumnSsn);
                String sex;
                switch (selectedUpdateInput) {
                    case "F":
                        sex = "F";
                        break;
                    case "M":
                        sex = "M";
                        break;
                }
                // Sex 값을 업데이트

employeeTable.setValueAt(selectedUpdateInput, i, ColumnSex);
                // 데이터베이스 업데이트
                String updateSQL = "UPDATE EMPLOYEE SET Sex
= ? WHERE Ssn = ?";
                PreparedStatement p =
conn.prepareStatement(updateSQL);
                p.setString(1, selectedUpdateInput);
                p.setString(2, ssn);
                p.executeUpdate();
                JOptionPane.showMessageDialog(null, "성공적
으로 값이 업데이트 되었습니다", "message", JOptionPane.INFORMATION_MESSAGE);
                System.out.println("db수정 완료");
            }
        }
        searchButton.doClick();
    } else if ("Salary".equals(selectedUpdateColumn)) {
        for (int i = 0; i < rowCount; i++) {
            if
(Boolean.TRUE.equals(employeeTable.getValueAt(i, checkBoxColumn))) {
                // Ssn 값을 문자열로 가져옴
                String ssn = (String)
employeeTable.getValueAt(i, ColumnSsn);
                // Salary 값을 업데이트

employeeTable.setValueAt(Double.parseDouble(updatedValue), i, ColumnSalary);
                // 데이터베이스 업데이트
                String updateSQL = "UPDATE EMPLOYEE SET
Salary = ? WHERE Ssn = ?";
                PreparedStatement p =
conn.prepareStatement(updateSQL);
                p.setDouble(1,
Double.parseDouble(updatedValue));
                p.setString(2, ssn);
                p.executeUpdate();
                JOptionPane.showMessageDialog(null, "성공적
으로 값이 업데이트 되었습니다", "message", JOptionPane.INFORMATION_MESSAGE);
                System.out.println("db수정 완료");
            }
        }
        searchButton.doClick();
    } else if ("Supervisor
(Ssn)".equals(selectedUpdateColumn)) {

```



```

        for (int i = 0; i < rowCount; i++) {
            if
(Boolean.TRUE.equals(employeeTable.getValueAt(i, checkBoxColumn))) {
                // Ssn 값을 문자열로 가져옴
                String ssn = (String)
employeeTable.getValueAt(i, ColumnSsn);
                // 데이터베이스에서 Supervisor의 Ssn을 찾음
                String query = "SELECT Ssn FROM EMPLOYEE
WHERE Ssn = ?";
                PreparedStatement queryStatement =
conn.prepareStatement(query);
                queryStatement.setString(1, updatedValue);
                ResultSet resultSet =
queryStatement.executeQuery();
                if (resultSet.next()) {
                    // 데이터베이스 업데이트
                    String updateSQL = "UPDATE EMPLOYEE SET
Super_ssn = ? WHERE Ssn = ?";
                    PreparedStatement updateStatement =
conn.prepareStatement(updateSQL);
                    updateStatement.setString(1,
updatedValue); // 새로운 Supervisor의 Ssn으로 업데이트
                    updateStatement.setString(2, ssn); // 선택
                    updateStatement.executeUpdate();
                    JOptionPane.showMessageDialog(null, "성
공적으로 값이 업데이트 되었습니다", "message", JOptionPane.INFORMATION_MESSAGE);
                    searchButton.doClick();
                    System.out.println("db수정 완료");
                } else {
                    JOptionPane.showMessageDialog(null, "입
력된 Ssn을 가진 직원을 찾을 수 없습니다. 변경할 수 없습니다.", "Unspecified Behavior",
JOptionPane.ERROR_MESSAGE);
                    System.out.println("입력된 Ssn을 가진 직원
을 찾을 수 없습니다. 변경할 수 없습니다.");
                }
            }
        }
    } else if ("Department".equals(selectedUpdateColumn)) {
        for (int i = 0; i < rowCount; i++) {
            if
(Boolean.TRUE.equals(employeeTable.getValueAt(i, checkBoxColumn))) {
                // Ssn 값을 문자열로 가져옴
                String ssn = (String)
employeeTable.getValueAt(i, ColumnSsn);
                String dno = null; // 기본값 설정
                switch (selectedUpdateInput) {
                    case "Research":
                        dno = "5";
                        break;
                    case "Administration":
                        dno = "4";
                        break;
                    case "Headquarters":
                        dno = "1";
                        break;
                    default:
                        System.out.println("유효하지 않은

```

```

Dname 값입니다.");

JOptionPane.showMessageDialog(null, "유효하지 않은 Dname 값입니다.", "Unspecified
Behavior", JOptionPane.ERROR_MESSAGE);
    }
    if (dno != null) {
        // 데이터베이스 업데이트
        String updateSQL = "UPDATE EMPLOYEE SET
Dno = ? WHERE Ssn = ?";
        PreparedStatement p =
conn.prepareStatement(updateSQL);

        if (dno != null) {
            p.setString(1, dno);
        } else {
            p.setNull(1, Types.VARCHAR); // null
처리

        }
        p.setString(2, ssn);
        p.executeUpdate();

employeeTable.setValueAt(selectedUpdateInput, i, ColumnDepartment);
        JOptionPane.showMessageDialog(null, "성
공적으로 값이 업데이트 되었습니다", "message", JOptionPane.INFORMATION_MESSAGE);
        //searchButton.doClick();
        System.out.println("db수정 완료");
    }
    }
    searchButton.doClick();
} else {
    System.out.println("업데이트 컬럼과 값을 선택해 주세요.");
    JOptionPane.showMessageDialog(null, "업데이트 컬럼과 값
을 선택해 주세요.", "Unspecified Behavior", JOptionPane.ERROR_MESSAGE);
}
} catch (SQLException e1) {
    e1.printStackTrace();
    String errorMessage = "SQL Error : 입력값을 다시 확인해주세
요.\n" + e1.getMessage();
    JOptionPane.showMessageDialog(null, errorMessage, "SQL
Error", JOptionPane.ERROR_MESSAGE);
}
} else {
    JOptionPane.showMessageDialog(null, "Ssn을 선택해야만 Update가
가능합니다.", "Unspecified Behavior", JOptionPane.ERROR_MESSAGE);
    System.out.println("Ssn을 선택해야만 Update가 가능합니다.");
}
}
}
});

```

update도 delete와 마찬가지로 ssn값을 넣어서 조회하지 않으면 업데이트할 수 없도록 하였다. for문을 돌며 해당 직원의 체크박스가 true인지 확인하고 체크되어있다면 입력받은 updateValue로 값을 변경하도록 하였다.

또한, update 실행 시 sql exeception이 일어나지는 않으나, 이 프로그램 상에서 따로 오류로 정의해둔 것들이 존재한다. (ssn을 선택해야만 업데이트 가능, 입력된 ssn을 가진 직원을 찾을 수 없는 경우) 이 경우는 에러 창을 띄울 때 "SQL Error"가 아닌 "Unspecified Behavior"라는 제목으로 오류창을 출력하였다.

3. 결과화면 실행창

1) 보고서 출력

The screenshot shows a web application titled "Information Retrieval System". On the left is a dark sidebar with icons for home, search, add, and subtract. The main area has a search bar with "Range" set to "All" and a "Search" button. Below the search bar is a table of employee data with columns: Name, Ssn, Bdate, Address, Sex, Salary, Supervisor, Dname, and check. The table contains 8 rows of data. Below the table is a large empty box for additional information or reports.

Name	Ssn	Bdate	Address	Sex	Salary	Supervisor	Dname	check
John B Smith	123456789	1965-01-09	731 Fondren, Ho...	M	30000.00	Franklin T Wong	Research	<input type="checkbox"/>
Franklin T Wong	333445555	1955-12-08	638 Voss, Houst...	M	40000.00	James E Borg	Research	<input type="checkbox"/>
Joyce A English	453453453	1972-07-31	5631 Rice, Hous...	F	25000.00	Franklin T Wong	Research	<input type="checkbox"/>
Ramesh K Nara...	666884444	1962-09-15	975 Fire Oak, H...	M	38000.00	Franklin T Wong	Research	<input type="checkbox"/>
James E Borg	888665555	1937-11-10	450 Stone, Hous...	M	55000.00	NULL	Headquarters	<input type="checkbox"/>
Jennifer S Walla...	987654321	1941-06-20	291 Berry, Bellai...	F	43000.00	James E Borg	Administration	<input type="checkbox"/>
Ahmad V Jabbar	987987987	1969-03-29	980 Dallas, Hou...	M	25000.00	Jennifer S Walla...	Administration	<input type="checkbox"/>
Alicia J Zelaya	999887777	1968-01-19	3321 Castle, Sp...	F	25000.00	Jennifer S Walla...	Administration	<input type="checkbox"/>

모든 범위에서, Item을 모두 선택했을 때 , DB에 있는 모든 직원이 검색된다.

2) 조건 검색

Information Retrieval System

Range: Department Research

Item: ☒ Name ☒ Ssn ☐ Bdate ☐ Address ☒ Sex ☒ Salary ☒ Supervisor ☒ Department

Search

Name	Ssn	Sex	Salary	Supervisor	Dname	check
John B Smith	123456789	M	30000.00	Franklin T Wong	Research	<input type="checkbox"/>
Franklin T Wong	333445555	M	40000.00	James E Borg	Research	<input type="checkbox"/>
Joyce A English	453453453	F	25000.00	Franklin T Wong	Research	<input type="checkbox"/>
Ramesh K Narayan	666884444	M	38000.00	Franklin T Wong	Research	<input type="checkbox"/>

Bdate와 Address를 제외하고 Department "Research" 부서에 근무하는 직원을 검색했다.

Information Retrieval System

Range: Salary 30000

Item: ☒ Name ☒ Ssn ☒ Bdate ☒ Address ☒ Sex ☒ Salary ☒ Supervisor ☒ Department

Search

Name	Ssn	Bdate	Address	Sex	Salary	Supervisor	Dname	check
John B Smith	123456789	1965-01-09	731 Fondren, Ho...	M	30000.00	Franklin T Wong	Research	<input type="checkbox"/>
Franklin T Wong	333445555	1955-12-08	638 Voss, Houst...	M	40000.00	James E Borg	Research	<input type="checkbox"/>
Ramesh K Nara...	666884444	1962-09-15	975 Fire Oak, H...	M	38000.00	Franklin T Wong	Research	<input type="checkbox"/>
James E Borg	888665555	1937-11-10	450 Stone, Hous...	M	55000.00	NULL	Headquarters	<input type="checkbox"/>
Jennifer S Walla...	987654321	1941-06-20	291 Berry, Bellai...	F	43000.00	James E Borg	Administration	<input type="checkbox"/>

Salary 가 30,000 이상인 직원 모두를 검색했다.

Information Retrieval System

Range Sex M

Item ☒ Name ☒ Ssn ☒ Bdate ☐ Address ☒ Sex ☐ Salary ☒ Supervisor ☐ Department **Search**


Name	Ssn	Bdate	Sex	Supervisor	check
John B Smith	123456789	1965-01-09	M	Franklin T Wong	<input type="checkbox"/>
Franklin T Wong	333445555	1955-12-08	M	James E Borg	<input type="checkbox"/>
Ramesh K Narayan	666884444	1962-09-15	M	Franklin T Wong	<input type="checkbox"/>
James E Borg	888665555	1937-11-10	M	NULL	<input type="checkbox"/>
Ahmad V Jabbar	987987987	1969-03-29	M	Jennifer S Wallace	<input type="checkbox"/>

Address와 Salary, Department를 제외하고, 성별이 남자인 직원 모두를 검색했다.

Information Retrieval System

Range Name

Item ☒ Name ☒ Ssn ☒ Bdate ☒ Address ☒ Sex ☒ Salary ☒ Supervisor ☒ Department **Search**

Name	Ssn	Bdate	Address	Sex	Salary	Supervisor	Dname	check
 <p>NO RESULTS... please check your input and try again.</p>								

이때, 만일 찾고자 하는 결과가 존재하지 않으면, No Results 라는 이미지가 뜨도록 하였다.

Information Retrieval System

Home

Search

Range

Name

John

Search

Item

☒ Name
 ☒ Ssn
 ☒ Bdate
 ☒ Address
 ☒ Sex
 ☒ Salary
 ☒ Supervisor
 ☒ Department

Name	Ssn	Bdate	Address	Sex	Salary	Supervisor	Dname	check
John B Smith	123456789	1965-01-09	731 Fondren,...	M	30000.00	Franklin T Wo...	Research	<input type="checkbox"/>

Information Retrieval System

Home

Search

Range

Name

B

Search

Item

☒ Name
 ☒ Ssn
 ☒ Bdate
 ☒ Address
 ☒ Sex
 ☒ Salary
 ☒ Supervisor
 ☒ Department

Name	Ssn	Bdate	Address	Sex	Salary	Supervisor	Dname	check
John B Smith	123456789	1965-01-09	731 Fondren,...	M	30000.00	Franklin T Wo...	Research	<input type="checkbox"/>

Information Retrieval System

Home

Search

Range

Name

John B Smith

Search

Item

☒ Name
 ☒ Ssn
 ☒ Bdate
 ☒ Address
 ☒ Sex
 ☒ Salary
 ☒ Supervisor
 ☒ Department

Name	Ssn	Bdate	Address	Sex	Salary	Supervisor	Dname	check
John B Smith	123456789	1965-01-09	731 Fondren,...	M	30000.00	Franklin T Wo...	Research	<input type="checkbox"/>

그 다음으로, 이름을 검색하는 경우에는 First name, Middle Name Initial, Last name의 조합과 각각을 검색하는 경우를 모두 고려하였다. 따라서 각각, 또는 그 중 하나를 검색해도 해당 이름을 가진 직원이 검색된다.

3) 조건 삭제

Information Retrieval System

Range: All

Item: ☒ Name ☒ Ssn ☒ Bdate ☒ Address ☒ Sex ☒ Salary ☒ Supervisor ☒ Department



Search

Name	Ssn	Bdate	Address	Sex	Salary	Supervisor	Dname	check
Mia S Thomas	123456777	2000-01-01	California	F	36000.00	John B Smith	Research	<input checked="" type="checkbox"/>
John B Smith	123456789	1965-01-09	731 Fondren, Ho	M	30000.00	Franklin T Wong	Research	<input type="checkbox"/>
Franklin T Wong	333445555	1955-12-08	638 Voss, Hous	M	50000.00	William T Davis	Research	<input type="checkbox"/>
Joyce A English	453453453	1972-07-31	5631 Rice, Hous	F	25000.00	Franklin T Wong	Research	<input type="checkbox"/>
Ramesh K Nara	666884444	1962-09-15	975 Fire Oak, H	M	50000.00	Franklin T Wong	Research	<input type="checkbox"/>
William T Davis	888665555	1937-11-10	450 Stone, Hous	M	55000.00	NULL	Headquarters	<input type="checkbox"/>
Jennifer S Walla	987654321	1941-06-20	291 Berry, Bellai	F	43000.00	William T Davis	Administration	<input type="checkbox"/>
Ahmad V Jabbar	987987987	1969-03-29	980 Dallas, Hou	M	25000.00	Jennifer S Walla	Administration	<input type="checkbox"/>
Alicia J Zelaya	999887777	1968-01-19	3321 Castle, Sp	F	25000.00	Jennifer S Walla	Administration	<input type="checkbox"/>

선택한 직원 : Mia S Thomas

인원 수 : 1

수정 : Name

Information Retrieval System

Range: All

Item: ☒ Name ☒ Ssn ☒ Bdate ☒ Address ☒ Sex ☒ Salary ☒ Supervisor ☒ Department

Search



Name	Ssn	Bdate	Address	Sex	Salary	Supervisor	Dname	check
John B Smith	123456789	1965-01-09	731 Fondren, Ho	M	30000.00	Franklin T Wong	Research	<input type="checkbox"/>
Franklin T Wong	333445555	1955-12-08	638 Voss, Hous	M	50000.00	William T Davis	Research	<input type="checkbox"/>
Joyce A English	453453453	1972-07-31	5631 Rice, Hous	F	25000.00	Franklin T Wong	Research	<input type="checkbox"/>
Ramesh K Nara	666884444	1962-09-15	975 Fire Oak, H	M	50000.00	Franklin T Wong	Research	<input type="checkbox"/>
William T Davis	888665555	1937-11-10	450 Stone, Hous	M	55000.00	NULL	Headquarters	<input type="checkbox"/>
Jennifer S Walla	987654321	1941-06-20	291 Berry, Bellai	F	43000.00	William T Davis	Administration	<input type="checkbox"/>
Ahmad V Jabbar	987987987	1969-03-29	980 Dallas, Hou	M	25000.00	Jennifer S Walla	Administration	<input type="checkbox"/>
Alicia J Zelaya	999887777	1968-01-19	3321 Castle, Sp	F	25000.00	Jennifer S Walla	Administration	<input type="checkbox"/>

Fname	Minit	Lname	Ssn	Bdate	Address	Sex	Salary	Super_ssn	Dno	created	modified
John	B	Smith	123456789	1965-01-09	731 Fondren, Houston, TX	M	30000.00	333445555	5	2023-11-06 18:19:01	2023-11-06 18:19:01
Franklin	T	Wong	333445555	1955-12-08	638 Voss, Houston, TX	M	50000.00	888665555	5	2023-11-06 18:19:01	2023-11-06 18:19:01
Joyce	A	English	453453453	1972-07-31	5631 Rice, Houston, TX	F	25000.00	333445555	5	2023-11-06 18:19:01	2023-11-06 18:19:01
Ramesh	K	Narayan	666884444	1962-09-15	975 Fire Oak, Humble, TX	M	50000.00	333445555	5	2023-11-06 18:19:01	2023-11-06 18:19:01
William	T	Davis	888665555	1937-11-10	450 Stone, Houston, TX	M	55000.00	NULL	1	2023-11-06 18:19:01	2023-11-06 18:19:01
Jennifer	S	Wallace	987654321	1941-06-20	291 Berry, Bellaire, TX	F	43000.00	888665555	4	2023-11-06 18:19:01	2023-11-06 18:19:01
Ahmad	V	Jabbar	987987987	1969-03-29	980 Dallas, Houston, TX	M	25000.00	987654321	4	2023-11-06 18:19:01	2023-11-06 18:19:01
Alicia	J	Zelaya	999887777	1968-01-19	3321 Castle, Spring, TX	F	25000.00	987654321	4	2023-11-06 18:19:01	2023-11-06 18:19:01

선택한 직원 :

인원 수 : 0

수정 : Name

Mia S Thomas 직원이 정상적으로 삭제되었다.

Ssn이 선택되지 않은 상태에서 삭제하려고 하면 오류 창이 발생한다.

4) 직원 추가

Information Retrieval System

Home

Search

Add

Remove

새로운 직원 정보 추가

First Name:

White

Address:

California

Middle Initial:

K

Sex:

female

Last Name:

Olivia

Salary:

36000

Ssn:

123456777

Super_ssn:

123456789

Birthdate:

2000-01-01

Dno:

5

Confirm

다음과 같이, 직원 추가 창에서 직원을 추가가 가능하다

Information Retrieval System

Home

Search

Add

Remove

새로운 직원 정보 추가

First Name:

Address:

Middle Initial:

Sex:

Last Name:

Salary:

Ssn:

Super_ssn:

Birthdate:

Dno:

Confirm

message

성공적으로 데이터가 추가되었습니다!

확인

추가가 정상적으로 완료되면 팝업창으로 알려준다.

Information Retrieval System

새로운 직원 정보 추가

First Name: Address:

Middle Initial:

Last Name: City:

Ssn: Super_ssn:

Birthdate: Dno:

message

성공적으로 데이터가 추가되었습니다!

OK

아래는 새로 추가한 직원 데이터를 확인한 모습이다. super_ssn을 '123456789'로 설정하였기 때문에, 해당 ssn을 가진 직원인 'John B Smith'가 supervisor로 나온다.

Information Retrieval System

Range: Name Seo W Nam

Item: ☒ Name ☒ Ssn ☒ Bdate ☒ Address ☒ Sex ☒ Salary ☒ Supervisor ☒ Department

Name	Ssn	Bdate	Address	Sex	Salary	Supervisor	Dname	check
Seoa W Nam	123456780	2000-09-01	1011-1 Back...	F	1000.00	John B Smith	Headquarters	<input type="checkbox"/>

Range: All

Item: ☒ Name ☒ Ssn ☒ Bdate ☒ Address ☒ Sex ☒ Salary ☒ Supervisor ☒ Department **Search**

Name	Ssn	Bdate	Address	Sex	Salary	Supervisor	Dname	check
White K Olivia	123456777	2000-01-01	California	F	36000.00	John B Smith	Research	<input type="checkbox"/>
John B Smith	123456789	1965-01-09	731 Fondren, Ho...	M	30000.00	Franklin T Wong	Research	<input type="checkbox"/>
Franklin T Wong	333445555	1955-12-08	638 Voss, Houst...	M	40000.00	James E Borg	Research	<input type="checkbox"/>
Joyce A English	453453453	1972-07-31	5631 Rice, Hous...	F	25000.00	Franklin T Wong	Research	<input type="checkbox"/>
Ramesh K Nara	666884444	1962-09-15	975 Fire Oak, H...	M	38000.00	Franklin T Wong	Research	<input type="checkbox"/>
James E Borg	888665555	1937-11-10	450 Stone, Hous...	M	55000.00	NULL	Headquarters	<input type="checkbox"/>
Jennifer S Walla	987654321	1941-06-20	291 Berry, Bellai...	F	43000.00	James E Borg	Administration	<input type="checkbox"/>
Ahmad V Jabbar	987987987	1969-03-29	980 Dallas, Hou...	M	25000.00	Jennifer S Walla	Administration	<input type="checkbox"/>
Alicia J Zelaya	999887777	1968-01-19	3321 Castle, Sp...	F	25000.00	Jennifer S Walla	Administration	<input type="checkbox"/>

```
mysql> select* from employee;
```

Fname	Minit	Lname	Ssn	Bdate	Address	Sex	Salary	Super_ssn	Dno	created	modified
White	K	Olivia	123456777	2000-01-01	California	F	36000.00	123456789	5	2023-11-06 06:34:44	2023-11-06 06:34:44
John	B	Smith	123456789	1965-01-09	731 Fondren, Houston, TX	M	30000.00	333445555	5	2023-11-06 18:19:01	2023-11-06 18:19:01
Franklin	T	Wong	333445555	1955-12-08	638 Voss, Houston, TX	M	40000.00	888665555	5	2023-11-06 18:19:01	2023-11-06 18:19:01
Joyce	A	English	453453453	1972-07-31	5631 Rice, Houston, TX	F	25000.00	333445555	5	2023-11-06 18:19:01	2023-11-06 18:19:01
Ramesh	K	Narayan	666884444	1962-09-15	975 Fire Oak, Humble, TX	M	38000.00	333445555	5	2023-11-06 18:19:01	2023-11-06 18:19:01
James	E	Borg	888665555	1937-11-10	450 Stone, Houston, TX	M	55000.00	NULL	1	2023-11-06 18:19:01	2023-11-06 18:19:01
Jennifer	S	Wallace	987654321	1941-06-20	291 Berry, Bellaire, TX	F	43000.00	888665555	4	2023-11-06 18:19:01	2023-11-06 18:19:01
Ahmad	V	Jabbar	987987987	1969-03-29	980 Dallas, Houston, TX	M	25000.00	987654321	4	2023-11-06 18:19:01	2023-11-06 18:19:01
Alicia	J	Zelaya	999887777	1968-01-19	3321 Castle, Spring, TX	F	25000.00	987654321	4	2023-11-06 18:19:01	2023-11-06 18:19:01

9 rows in set (0.00 sec)

```
mysql>
```

이상이 없을 때 정상적으로 추가한 White K Olivia가 들어온다.



5) 조건 업데이트

Range: All

Item: ☒ Name ☒ Ssn ☒ Bdate ☒ Address ☒ Sex ☒ Salary ☒ Supervisor ☒ Department **Search**

Name	Ssn	Bdate	Address	Sex	Salary	Supervisor	Dname	check
Mia S Thomas	123456777	2000-01-01	California	F	36000.00	John B Smith	Research	<input type="checkbox"/>
John B Smith	123456789	1965-01-09	731 Fondren, Ho...	M	30000.00	Franklin T Wong	Research	<input type="checkbox"/>
Franklin T Wong	333445555	1955-12-08	638 Voss, Houst...	M	40000.00	James E Borg	Research	<input type="checkbox"/>
Joyce A English	453453453	1972-07-31	5631 Rice, Hous...	F	25000.00	Franklin T Wong	Research	<input type="checkbox"/>
Ramesh K Nara	666884444	1962-09-15	975 Fire Oak, H...	M	38000.00	Franklin T Wong	Research	<input type="checkbox"/>
James E Borg	888665555	1937-11-10	450 Stone, Hous...	M	55000.00	NULL	Headquarters	<input checked="" type="checkbox"/>
Jennifer S Walla	987654321	1941-06-20	291 Berry, Bellai...	F	43000.00	James E Borg	Administration	<input type="checkbox"/>
Ahmad V Jabbar	987987987	1969-03-29	980 Dallas, Hou...	M	25000.00	Jennifer S Walla	Administration	<input type="checkbox"/>
Alicia J Zelaya	999887777	1968-01-19	3321 Castle, Sp...	F	25000.00	Jennifer S Walla	Administration	<input type="checkbox"/>

선택한 직원 : James E Borg
인원 수 : 1

수정 : Name
William T Davis  

James E Borg의 이름을 William T Davis로 변경하려고 한다.

Update 결과 다음과 같이 Jame E Borg가 Supervisor였던 사람들의 이름까지 update가 완료된다.

Range

All

Item

☒ Name
 ☐ Ssn
 ☒ Bdate
 ☒ Address
 ☒ Sex
 ☒ Salary
 ☒ Supervisor
 ☒ Department

Search

Name	Bdate	Address	Sex	Salary	Supervisor	Dname	check
Mia S Thomas	2000-01-01	California	F	36000.00	John B Smith	Research	<input type="checkbox"/>
John B Smith	1965-01-09	731 Fondren, Hous...	M	30000.00	Franklin T Wong	Research	<input checked="" type="checkbox"/>
Franklin T Wong	1955-12-08	638 Voss, Houston...	M	40000.00	William T Davis	Research	<input type="checkbox"/>
Joyce A English	1972-07-31	5631 Rice, Housto...	F	25000.00	Franklin T Wong	Research	<input type="checkbox"/>
Ramesh K Naraya...	1962-09-15	975 Fire Oak Hu...	M	38000.00	Franklin T Wong	Research	<input type="checkbox"/>
William T Davis	1937-11-10	450 S			NULL	Headquarters	<input type="checkbox"/>
Jennifer S Wallace	1941-06-20	291 B			William T Davis	Administration	<input type="checkbox"/>
Ahmad V Jabbar	1969-03-29	980 D			Jennifer S Wallace	Administration	<input type="checkbox"/>
Alicia J Zelaya	1968-01-19	3321			Jennifer S Wallace	Administration	<input type="checkbox"/>

Unspecified Behavior

Ssn을 선택해야만 Update가 가능합니다.

확인

선택한 직원 : John B Smith

수정 : Name

Jame K William

↑

↓

Ssn이 선택되어 있지 않으면, Update가 불가능하고, 팝업창으로 알려준다.

Information Retrieval System

Range: All

Item: ☒ Name ☒ Ssn ☒ Bdate ☒ Address ☒ Sex ☒ Salary ☒ Supervisor ☒ Department

Search

Name	Ssn	Bdate	Address	Sex	Salary	Supervisor	Dname	check
Mia S Thomas	123456777	2000-01-01	California	F	36000.00	John B Smith	Research	<input type="checkbox"/>
John B Smith	123456789	1965-01-09	731 Fondren, Ho	M	30000.00	Franklin T Wong	Research	<input type="checkbox"/>
Franklin T Wong	333445555	1955-12-08	638 Voss, Hous.	M	50000.0	William T Davis	Research	<input checked="" type="checkbox"/>
Joyce A English	453453453	1972-07-31	5631 Rice, Hous.	F	25000.00	Franklin T Wong	Research	<input type="checkbox"/>
Ramesh K Nara.	666884444	1962-09-15				Franklin T Wong	Research	<input checked="" type="checkbox"/>
William T Davis	888665555	1937-11-10				NULL	Headquarters	<input type="checkbox"/>
Jennifer S Walla.	987654321	1941-06-20				William T Davis	Administration	<input type="checkbox"/>
Ahmad V Jabbar	987987987	1969-03-29				Jennifer S Walla.	Administration	<input type="checkbox"/>
Alicia J Zelaya	999887777	1968-01-19				Jennifer S Walla.	Administration	<input type="checkbox"/>

message

성공적으로 값이 업데이트 되었습니다

확인

Fname	Minit	Lname	Ssn	Bdate	Address	Sex	Salary	Super_ssn	Dno	created	modified
Mia	S	Thomas	123456777	2000-01-01	California	F	36000.00	123456789	5	2023-11-06 06:34:44	2023-11-06 06:34:44
John	B	Smith	123456789	1965-01-09	731 Fondren, Houston, TX	M	30000.00	333445555	5	2023-11-06 18:19:01	2023-11-06 18:19:01
Franklin	T	Wong	333445555	1955-12-08	638 Voss, Houston, TX	M	50000.00	888665555	5	2023-11-06 18:19:01	2023-11-06 18:19:01
Joyce	A	English	453453453	1972-07-31	5631 Rice, Houston, TX	F	25000.00	333445555	5	2023-11-06 18:19:01	2023-11-06 18:19:01
Ramesh	K	Narayan	666884444	1962-09-15	975 Fire Oak, Humble, TX	M	50000.00	333445555	5	2023-11-06 18:19:01	2023-11-06 18:19:01
William	T	Davis	888665555	1937-11-10	450 Stone, Houston, TX	M	55000.00	NULL	1	2023-11-06 18:19:01	2023-11-06 18:19:01
Jennifer	S	Wallace	987654321	1941-06-20	291 Berry, Bellaire, TX	F	43000.00	888665555	4	2023-11-06 18:19:01	2023-11-06 18:19:01
Ahmad	V	Jabbar	987987987	1969-03-29	980 Dallas, Houston, TX	M	25000.00	987654321	4	2023-11-06 18:19:01	2023-11-06 18:19:01
Alicia	J	Zelaya	999887777	1968-01-19	3321 Castle, Spring, TX	F	25000.00	987654321	4	2023-11-06 18:19:01	2023-11-06 18:19:01

선택한 직원 : Franklin T Wong Ramesh K Narayan

수정 : Salary

인원 수 : 2

50000

↑

↓

Franklin T Wong 과 Ramesh K Narayan의 Salary를 각각 50000으로 update 한 모습이다.

4. 팀원 역할 분담표

남서아	GUI 다수 디자인 / 구현 및 JDBC 기능 구현 (검색, 업데이트) 보고서 작성
허진혁	GUI 일부 구현 및 JDBC 기능 구현 (출력, 검색, 삭제, 업데이트) 보고서 작성
안희수	GUI 일부 구현 및 JDBC 기능 구현 (출력, 검색, 삭제, 업데이트) 보고서 작성