데이터베이스실무 과제

UPDATE, DELETE 기능 구현

2501110203

김호석

2025.10.31

목차

1. 과제 개요

1-1. 과제 개요

1. mvc_db_test 패키지의 클래스를 사용합니다.

2. 고객테이블의 저장된 고객 정보를 수정하는 기능 추가
   - 아래의 진행 순서를 따를 것
      1. 업데이트할 고객 아이디 입력
      2. 해당 고객의 정보 출력
      3. 수정할 데이터 선택 (ID 는 변경 불가)
      4. 수정할 데이터 입력
      5. 데이터 UPDATE
      6. 수정된 고객 정보 출력

3. 고객테이블의 저장된 고객 정보를 삭제하는 기능 추가
   - 고객 id 를 입력 받을 것
   - 삭제할 것인지 확인 절차를 진행할 것

1-2. 과제 수행 목표

1. 조회, 추가, 수정, 삭제 기능을 모두 이용할 수 있도록 하고자 함.
2. Switch Case 문을 사용하여 각 기능을 분기별로 관리하고자 함.
3. 고객 테이블 뿐만 아니라 제품, 주문 테이블도 접근할 수 있도록 하고자 함.

2. 코드

GitHub Link

https://github.com/WaltDev29/Course_SQL/tree/main/SQL/src/mvc_jdbc_test

2-1. Controller Package

2-1-1. MainController Class2 (MainController Class 를 제작 후 리팩토링 함.)

```java
package mvc_jdbc_test.controller;

import ...

public class MainController2 {  Waltdev29
    public static void main(String[] args) {  Waltdev29
        // 변수 선언
        Scanner sc = new Scanner( source: System.in);
        Connection con;
        MainView mv = new MainView();

        // 결과 담을 List 선언
        ArrayList<Customer> customerList;
        ArrayList<Product> productList;
        ArrayList<Order> orderList;

        // JDBC 연결
        con = JDBCConnector.getConnection();

        int mainState = 0;
        int subState;
        String mode;
        String pk = "";
        boolean proceed;
        // 메인 프로그램
        while (true) {
            // Mode 선택
            if (mainState == 0) { // 메인
                mv.showHomeView();
                mainState = mv.inputAnswer( sc: sc,  min: 0,  max: 4);
                if (mainState == 0) break;
            }

            // 잘못된 입력
            if (mainState < 0 || mainState > 4) {
                System.out.println("잘못된 입력입니다. 1~3의 정수를 입력해주세요.");
            }

```

```
48          // Mode 할당
49          mode = switch (mainState) {
50              case 1 -> "조회";
51              case 2 -> "추가";
52              case 3 -> "수정";
53              case 4 -> "삭제";
54              default -> "";
55          };
56
57          // DB 선택
58          mv.showMainView(mode);
59          subState = mv.inputAnswer( sc: sc,  min: 0,  max: 3);
60
61          // 처음으로 돌아가기
62          if (subState == 0) {
63              mainState = 0;
64              continue;
65          }
66
67
68          // 데이터 추가
69          if (mainState == 2) {
70              if (subState == 1) {
71                  insertCustomerInfo(con,  sc: sc);
72              } else if (subState == 2) {
73                  insertProductInfo(con,  sc: sc);
74              } else if (subState == 3) {
75                  insertOrderInfo(con,  sc: sc);
76              }
77              mv.inputEnter( sc: sc);
78              continue;
79          }
80
```

```
81          // DB 출력
82          if (subState == 1) {
83              customerList = getCustomerList(con,  target: null);
84              printItemList( itemList: customerList,  view: new CustomerView());
85          } else if (subState == 2) {
86              productList = getProductList(con,  target: null);
87              printItemList( itemList: productList,  view: new ProductView());
88          } else if (subState == 3) {
89              orderList = getOrderList(con,  target: null);
90              printItemList( itemList: orderList,  view: new OrderView());
91          }
92
93
94          // 데이터 조회일 경우 바로 처음으로
95          if (mainState == 1) {
96              mv.inputEnter( sc: sc);
97              continue;
98          }
99
100
101         // PK 입력
102         if (subState == 1) {
103             pk = inputCustomerPk(con,  sc: sc);
104             customerList = getCustomerList(con,  target: pk);
105             if (mainState == 3) printItemWithIndex( item: customerList.get(0),  view: new CustomerView());
106             else printItem( item: customerList.get(0),  view: new CustomerView());
107         } else if (subState == 2) {
108             pk = inputProductPk(con,  sc: sc);
109             productList = getProductList(con,  target: pk);
110             if (mainState == 3) printItemWithIndex( item: productList.get(0),  view: new ProductView());
111             else printItem( item: productList.get(0),  view: new ProductView());
112         } else if (subState == 3) {
113             pk = inputOrderPk(con,  sc: sc);
114             orderList = getOrderList(con,  target: pk);
115             if (mainState == 3) printItemWithIndex( item: orderList.get(0),  view: new OrderView());
116             else printItem( item: orderList.get(0),  view: new OrderView());
117         }
118
```

```
119             // 삭제
120             if (mainState == 4) {
121                 System.out.println("\n해당 데이터를 삭제하시겠습니까?");
122                 if (mv.askYorN( sc: sc,  yesText: "삭제",  yesChar: "Y",  noText: "취소",  noChar: "N")) {
123                     deleteInfo(con, subState,  target: pk);
124                     mv.inputEnter( sc: sc);
125                 }
126             }
127 
128             // 데이터 수정
129             if (mainState == 3) {
130                 System.out.println("\n수정할 항목의 번호를 입력해주세요. 뒤로가기 : 0");
131                 proceed = updateInfo(con,  sc: sc, subState,  target: pk);
132                 if (!proceed) continue;
133 
134                 System.out.println("\n수정이 완료되었습니다.");
135                 System.out.print("\n--- 수정 정보 ---");
136                 if (subState == 1) {
137                     customerList = getCustomerList(con,  target: pk);
138                     printItemWithIndex( item: customerList.get(0),  view: new CustomerView());
139                 } else if (subState == 2) {
140                     productList = getProductList(con,  target: pk);
141                     printItem( item: productList.get(0),  view: new ProductView());
142                 } else if (subState == 3) {
143                     orderList = getOrderList(con,  target: pk);
144                     printItem( item: orderList.get(0),  view: new OrderView());
145                 }
146                 mv.inputEnter( sc: sc);
147             }
148         }
149 
150         System.out.println("\n프로그램을 종료합니다.");
151         sc.close();
152     }
153 
154 
```

```java
    // Input
    private static String inputCustomerPk(Connection con, @NotNull Scanner sc) {  1개 사용 위치  Waltdev29
        ArrayList<Customer> customerList = getCustomerList(con, target: null);
        String target;
        System.out.println("고객의 고객 아이디를 입력하세요.\n");
        while (true) {
            System.out.print("고객 아이디 : ");
            target = sc.nextLine();
            if (validatePk( EntityList: customerList, pk: target)) break;
            else System.out.println("존재하지 않는 고객 아이디입니다. 다시 입력해주세요.\n");
        }
        return target;
    }

    private static String inputProductPk(Connection con, @NotNull Scanner sc) {  1개 사용 위치  Waltdev29
        ArrayList<Product> productList = getProductList(con, target: null);
        String target;
        System.out.println("제품의 제품번호를 입력하세요.\n");
        while (true) {
            System.out.print("제품번호 : ");
            target = sc.nextLine();
            if (validatePk( EntityList: productList, pk: target)) break;
            else System.out.println("존재하지 않는 제품입니다. 다시 입력해주세요.\n");
        }
        return target;
    }

    private static String inputOrderPk(Connection con, @NotNull Scanner sc) {  1개 사용 위치  Waltdev29
        ArrayList<Order> orderList = getOrderList(con, target: null);
        String target;
        System.out.println("주문의 주문번호를 입력하세요.\n");
        while (true) {
            System.out.print("주문번호 : ");
            target = sc.nextLine();
            if (validatePk( EntityList: orderList, pk: target)) break;
            else System.out.println("존재하지 않는 주문번호입니다. 다시 입력해주세요.\n");
        }
        return target;
    }
```

```java
    // Validate
    private static boolean validatePk( @NotNull ArrayList<? extends Entity> EntityList, String pk) {  3개 사용 위치  Waltdev29
        for (Entity e : EntityList) if (e.getId().equalsIgnoreCase( anotherString: pk)) return true;
        return false;
    }


    // Print
    private static <T> void printItem(T item,  @NotNull ObjectView<T> view) {  5개 사용 위치  Waltdev29
        view.printHead();
        view.printCols();
        view.printItem(item);
        System.out.println();
    }

    private static <T> void printItemList( @NotNull ArrayList<T> itemList,  @NotNull ObjectView<T> view) {  6개 사용 위치  Wa
        view.printHead();
        view.printCols();
        for (T item : itemList) {
            view.printItem(item);
            System.out.println();
        }
        view.printFoot();
    }

    private static <T> void printItemWithIndex(T item,  @NotNull ObjectView<T> view) {  4개 사용 위치  Waltdev29
        view.printItemWithIndex(item);
    }


    // SELECT
    @NotNull
    private static ArrayList<Customer> getCustomerList(Connection con, String target) {  4개 사용 위치  Waltdev29
        ArrayList<Customer> customerList = new ArrayList<>();
        Customer customer;
        String sql;

        if (target != null) sql = "SELECT * FROM 고객 WHERE 고객아이디 = ?";
        else sql = "SELECT * FROM 고객";
        try {
            PreparedStatement ps = con.prepareStatement(sql);
            if (target != null) ps.setString( parameterIndex: 1,  x: target);
            ResultSet rs = ps.executeQuery();

            while (rs.next()) {
                customer = new Customer(
                        id: rs.getString( columnLabel: "고객아이디"),
                        name: rs.getString( columnLabel: "고객이름"),
                        age: rs.getInt( columnLabel: "나이"),
                        grade: rs.getString( columnLabel: "등급"),
                        job: rs.getString( columnLabel: "직업"),
                        point: rs.getInt( columnLabel: "적립금"));
                customerList.add(customer);
            }

            ps.close();
            rs.close();

        } catch (SQLException e) {
            System.out.println("Statement or SQL Error");
            throw new RuntimeException(e);
        }
        return customerList;
    }

```

```java
private static ArrayList<Product> getProductList(Connection con, String target) {  4개 사용 위치  Waltdev29
    ArrayList<Product> productList = new ArrayList<>();
    Product product;
    String sql;

    if (target != null) sql = "SELECT * FROM 제품 WHERE 제품번호 = ?";
    else sql = "SELECT * FROM 제품";
    try {
        PreparedStatement ps = con.prepareStatement(sql);
        if (target != null) ps.setString( parameterIndex: 1,  x: target);
        ResultSet rs = ps.executeQuery();

        while (rs.next()) {
            product = new Product(
                    id: rs.getString( columnLabel: "제품번호"),
                    productName: rs.getString( columnLabel: "제품명"),
                    productAmount: rs.getInt( columnLabel: "재고량"),
                    productPrice: rs.getInt( columnLabel: "단가"),
                    manufacturer: rs.getString( columnLabel: "제조업체")
            );
            productList.add(product);
        }

        ps.close();
        rs.close();

    } catch (SQLException e) {
        System.out.println("Statement or SQL Error");
        throw new RuntimeException(e);
    }

    return productList;
}

private static ArrayList<Order> getOrderList(Connection con, String target) {  4개 사용 위치  Waltdev29
    ArrayList<Order> ordersList = new ArrayList<>();
    Order order;
    String sql;

    if (target != null) sql = "SELECT * FROM 주문 WHERE 주문번호 = ?";
    else sql = "SELECT * FROM 주문";
    try {
        PreparedStatement ps = con.prepareStatement(sql);
        if (target != null) ps.setString( parameterIndex: 1,  x: target);
        ResultSet rs = ps.executeQuery();

        while (rs.next()) {
            order = new Order(
                    id: rs.getString( columnLabel: "주문번호"),
                    customerId: rs.getString( columnLabel: "주문고객"),
                    orderedProduct: rs.getString( columnLabel: "주문제품"),
                    amount: rs.getInt( columnLabel: "수량"),
                    deliveryAddress: rs.getString( columnLabel: "배송지"),
                    orderDate: rs.getDate( columnLabel: "주문일자")
            );
            ordersList.add(order);
        }
        ps.close();
        rs.close();
    } catch (SQLException e) {
        System.out.println("Statement or SQL Error");
        throw new RuntimeException(e);
    }
    return ordersList;
}
```

```java
    // INSERT
    private static void insertCustomerInfo(Connection con, Scanner sc) {  1개 사용 위치  Waltdev29
        MainView mv = new MainView();
        InputCustomerInfoView iciv = new InputCustomerInfoView();
        ArrayList<Customer> inputCustomerList = new ArrayList<>();

        while (true) {
            Customer customer = iciv.inputCustomerInfo( sc: sc);

            // 데이터 저장 여부 확인
            if (!mv.askYorN( sc: sc,  yesText: "데이터 저장",  yesChar: "S",  noText: "다시 입력",  noChar: "R")) continue;

            // 데이터 INSERT
            try {
                String sql = "INSERT INTO 고객 VALUES(?,?,?,?,?,?)";
                PreparedStatement ps = con.prepareStatement(sql);
                ps.setString( parameterIndex: 1,  x: customer.getId());
                ps.setString( parameterIndex: 2,  x: customer.getName());
                ps.setInt( parameterIndex: 3,  x: customer.getAge());
                ps.setString( parameterIndex: 4,  x: customer.getGrade());
                ps.setString( parameterIndex: 5,  x: customer.getJob());
                ps.setInt( parameterIndex: 6,  x: customer.getPoint());
                ps.executeUpdate();
                ps.close();
            } catch (SQLException e) {
                System.out.println("Statement or SQL Error");
                throw new RuntimeException(e);
            }
            System.out.println("데이터가 저장되었습니다.\n");
            inputCustomerList.add(customer);

            // 계속 입력받을지 선택
            if (!mv.askYorN( sc: sc,  yesText: "계속 입력",  yesChar: "C",  noText: "입력 종료",  noChar: "E")) break;
        }

        // 입력 내용 출력
        System.out.println("\n--- 입력 종료 ---");
        System.out.println("====== 입력 내용 ======\n");
        printItemList( itemList: inputCustomerList,  view: new CustomerView());
    }
```

```java

    private static void insertProductInfo(Connection con, Scanner sc) {  1개 사용 위치  Waltdev29
        MainView mv = new MainView();
        InputProductInfoView ipiv = new InputProductInfoView();
        ArrayList<Product> inputProductList = new ArrayList<>();

        while (true) {
            Product product = ipiv.inputProductInfo( sc: sc);

            // 데이터 저장 여부 확인
            if (!mv.askYorN( sc: sc,  yesText: "데이터 저장",  yesChar: "S",  noText: "다시 입력",  noChar: "R")) continue;

            // 데이터 INSERT
            try {
                String sql = "INSERT INTO 제품 VALUES(?,?,?,?,?)";
                PreparedStatement ps = con.prepareStatement(sql);
                ps.setString( parameterIndex: 1,  x: product.getId());
                ps.setString( parameterIndex: 2,  x: product.getProductName());
                ps.setInt( parameterIndex: 3,  x: product.getProductAmount());
                ps.setInt( parameterIndex: 4,  x: product.getProductPrice());
                ps.setString( parameterIndex: 5,  x: product.getManufacturer());
                ps.executeUpdate();
                ps.close();
            } catch (SQLException e) {
                System.out.println("Statement or SQL Error");
                throw new RuntimeException(e);
            }
            System.out.println("데이터가 저장되었습니다.\n");
            inputProductList.add(product);

            // 계속 입력받을지 선택
            if (!mv.askYorN( sc: sc,  yesText: "계속 입력",  yesChar: "C",  noText: "입력 종료",  noChar: "E")) break;
        }

        // 입력 내용 출력
        System.out.println("\n--- 입력 종료 ---");
        System.out.println("====== 입력 내용 ======\n");
        printItemList( itemList: inputProductList,  view: new ProductView());
    }
```

```java
    private static void insertOrderInfo(Connection con, Scanner sc) {  1개 사용 위치  ± Waltdev29
        MainView mv = new MainView();
        InputOrderInfoView ioiv = new InputOrderInfoView();
        ArrayList<Order> inputOrderList = new ArrayList<>();

        while (true) {
            Order order = ioiv.inputOrderInfo( sc: sc);

            // 데이터 저장 여부 확인
            if (!mv.askYorN( sc: sc,  yesText: "데이터 저장",  yesChar: "S",  noText: "다시 입력",  noChar: "R")) continue;

//            SimpleDateFormat sdf = new SimpleDateFormat("yyyy-MM-dd");
//            Date date = Date.valueOf(sdf.format(order.getOrderDate()));
            Date date = new Date( date: order.getOrderDate().getTime());

            // 데이터 INSERT
            try {
                String sql = "INSERT INTO 주문 VALUES(?,?,?,?,?,?)";
                PreparedStatement ps = con.prepareStatement(sql);
                ps.setString( parameterIndex: 1,  x: order.getId());
                ps.setString( parameterIndex: 2,  x: order.getCustomerId());
                ps.setString( parameterIndex: 3,  x: order.getOrderedProduct());
                ps.setInt( parameterIndex: 4,  x: order.getAmount());
                ps.setString( parameterIndex: 5,  x: order.getDeliveryAddress());
                ps.setDate( parameterIndex: 6,  x: date);
                ps.executeUpdate();
                ps.close();
            } catch (SQLException e) {
                System.out.println("Statement or SQL Error");
                throw new RuntimeException(e);
            }
            System.out.println("데이터가 저장되었습니다.\n");
            inputOrderList.add(order);

            // 계속 입력받을지 선택
            if (!mv.askYorN( sc: sc,  yesText: "계속 입력",  yesChar: "C",  noText: "입력 종료",  noChar: "E")) break;
        }

        // 입력 내용 출력
        System.out.println("\n--- 입력 종료 ---");
        System.out.println("====== 입력 내용 ======\n");
        printItemList( itemList: inputOrderList,  view: new OrderView());
    }
```

```java
// UPDATE
private static boolean updateInfo(Connection con, Scanner sc, int subState, String target) {  1개 사용 위치  Waltdev29
    MainView mv = new MainView();
    String value = "";
    int valueInt = 0;
    String[] cols = new String[]{};
    String table = "";
    String pk = "";
    int intCol1 = 0;
    int intCol2 = 0;

    if (subState == 1) {
        cols = new String[]{"고객이름", "나이", "등급", "직업", "적립금"};
        intCol1 = 2;
        intCol2 = 5;
        table = "고객";
        pk = "고객아이디";
    } else if (subState == 2) {
        cols = new String[]{"제품명", "재고량", "단가", "제조업체"};
        intCol1 = 2;
        intCol2 = 3;
        table = "제품";
        pk = "제품번호";
    } else if (subState == 3) {
        cols = new String[]{"제품명", "배송지", "수량", "주문일자"};
        intCol2 = 3;
        table = "주문";
        pk = "주문번호";
    }

    int index = mv.inputAnswer( sc: sc,  min: 0,  max: cols.length);

    if (index == 0) return false;

    System.out.printf("\n수정할 %s 입력\n", cols[index - 1]);

    if (index == intCol1 || index == intCol2) valueInt = mv.inputAnswer( sc: sc,  min: 0,  max: 100000);
    else {
        System.out.print("\n입력 : ");
        value = sc.nextLine();
    }

    String sql = "UPDATE " + table + " SET " + cols[index - 1] + " = ? WHERE " + pk + " = ?";
    try {
        PreparedStatement ps = con.prepareStatement(sql);
        if (index == 2 || index == 5) ps.setInt( parameterIndex: 1,  x: valueInt);
        else ps.setString( parameterIndex: 1,  x: value);
        ps.setString( parameterIndex: 2,  x: target);
        ps.executeUpdate();
        ps.close();
    } catch (SQLException e) {
        System.out.println("Statement or SQL Error");
        throw new RuntimeException(e);
    }
    return true;
}
```

```java
    // DELETE
    private static void deleteInfo(Connection con, int subState, String target) {  1개 사용 위치  Waltdev29
        String table = "";
        String pk = "";

        if (subState == 1) {
            table = "고객";
            pk = "고객아이디";
        } else if (subState == 2) {
            table = "제품";
            pk = "제품번호";
        } else if (subState == 3) {
            table = "주문";
            pk = "주문번호";
        }

        String sql = "DELETE FROM " + table + " WHERE " + pk + " = ?";

        try {
            PreparedStatement ps = con.prepareStatement(sql);
            ps.setString( parameterIndex: 1,  x: target);
            ps.executeUpdate();
            ps.close();
        } catch (SQLException e) {
            System.out.println("Statement or SQL Error");
            throw new RuntimeException(e);
        }
        System.out.println("\n데이터가 삭제되었습니다.");
    }
}
```

2-2. Entity Package

2-2-1. Entity Class

```java
package mvc_jdbc_test.entity;

public abstract class Entity {  9개 사용 위치  3개 상속자  Waltdev29
    public abstract String getId();  11개 사용 위치  3개 구현  Waltdev29
}
```

2-2-2. Customer Class

```java
package mvc_jdbc_test.entity;

public class Customer extends Entity {  26개 사용 위치  Waltdev29
    String id;  3개 사용 위치
    String name;  3개 사용 위치
    int age;  3개 사용 위치
    String grade;  3개 사용 위치
    String job;  3개 사용 위치
    int point;  3개 사용 위치

    public Customer(String id, String name, int age, String grade, String job, int point) {  3개 사용 위치  Waltdev29
        this.id = id;
        this.name = name;
        this.age = age;
        this.grade = grade;
        this.job = job;
        this.point = point;
    }

    public String getId() { return id; }
    public void setId(String id) { this.id = id; }

    public String getName() { return name; }
    public void setName(String name) { this.name = name; }

    public int getAge() { return age; }
    public void setAge(int age) { this.age = age; }

    public String getGrade() { return grade; }
    public void setGrade(String grade) { this.grade = grade; }

    public String getJob() { return job; }
    public void setJob(String job) { this.job = job; }

    public int getPoint() { return point; }
    public void setPoint(int point) { this.point = point; }
}
```

2-2-3. Product Class

```java
package mvc_jdbc_test.entity;

public class Product extends Entity{  26개 사용 위치  Waltdev29
    String id;  2개 사용 위치
    String productName;  2개 사용 위치
    int productAmount;  2개 사용 위치
    int productPrice;  2개 사용 위치
    String manufacturer;  2개 사용 위치

    public Product(String id, String productName, int productAmount, int productPrice, String manufacturer) {  3개 사용 위치
        this.id = id;
        this.productName = productName;
        this.productAmount = productAmount;
        this.productPrice = productPrice;
        this.manufacturer = manufacturer;
    }

    public String getId() { return id; }

    public String getProductName() { return productName; }

    public int getProductAmount() { return productAmount; }

    public int getProductPrice() { return productPrice; }

    public String getManufacturer() { return manufacturer; }
}
```

2-2-4. Order Class

```java
package mvc_jdbc_test.entity;

import java.util.Date;

public class Order extends Entity {  Waltdev29 *
    private String id;
    private String customerId;
    private String orderedProduct;
    private int amount;
    private String deliveryAddress;
    private Date orderDate;

    public Order(String id, String customerId, String orderedProduct, int amount, String deliveryAddress, Date orderDate)
        this.id = id;
        this.customerId = customerId;
        this.orderedProduct = orderedProduct;
        this.amount = amount;
        this.deliveryAddress = deliveryAddress;
        this.orderDate = orderDate;
    }

    public String getId() { return id; }

    public String getCustomerId() { return customerId; }

    public String getOrderedProduct() { return orderedProduct; }

    public int getAmount() { return amount; }

    public String getDeliveryAddress() { return deliveryAddress; }

    public Date getOrderDate() { return orderDate; }
}
```

2-3. View Package

2-3-1. ObjectView Class

```java
package mvc_jdbc_test.view;

public abstract class ObjectView<T> {  8개 사용 위치  3개 상속자  Waltdev29

    public abstract void printItem(T item);  4개 사용 위치  3개 구현  Waltdev29

    public abstract void printHead();  4개 사용 위치  3개 구현  Waltdev29

    public abstract void printCols();  4개 사용 위치  3개 구현  Waltdev29

    public void printFoot() {  2개 사용 위치  Waltdev29
        System.out.println("\n=========================");
        System.out.println("====== Print Done ======");
        System.out.println("=========================\n");
    }

    public abstract void printItemWithIndex(T item);  4개 사용 위치  3개 구현  Waltdev29
}
```

2-3-2. CustomerView Class

```java
package mvc_jdbc_test.view;

import mvc_jdbc_test.entity.Customer;

public class CustomerView extends ObjectView<Customer> {  13개 사용 위치  Waltdev29
    String title = "고객 정보";  1개 사용 위치

    public void printHead() {  4개 사용 위치  Waltdev29
        System.out.println("\n========================");
        System.out.println("========" + title + "=======");
        System.out.println("========================\n");
    }

    public void printCols() {  4개 사용 위치  Waltdev29
        System.out.printf("%-10s %-10s %-4s %-10s %-10s %-8s\n", "고객아이디", "고객이름", "나이", "등급", "직업", "적립금");
        System.out.println("-------------------------------------------------------------------------------------");
    }

    @Override  4개 사용 위치  Waltdev29
    public void printItem( @NotNull Customer customer) {
        System.out.printf("%-14s ", customer.getId());
        System.out.printf("%-10s ", customer.getName());
        System.out.printf("%-5d ", customer.getAge());
        System.out.printf("%-11s ", customer.getGrade());
        System.out.printf("%-10s ", customer.getJob());
        System.out.printf("%-8d ", customer.getPoint());
    }

    public void printItemWithIndex( @NotNull Customer customer) {  4개 사용 위치  Waltdev29
        System.out.println("\n====== 고객 정보 ======");
        System.out.printf("1. 고객이름 : %s\n", customer.getName());
        System.out.printf("2. 나이 : %d\n", customer.getAge());
        System.out.printf("3. 등급 : %s\n", customer.getGrade());
        System.out.printf("4. 직업 : %s\n", customer.getJob());
        System.out.printf("5. 적립금 : %d\n", customer.getPoint());
    }
}
```

2-3-3. ProductView Class

```java
package mvc_jdbc_test.view;

import mvc_jdbc_test.entity.Product;

public class ProductView extends ObjectView<Product> {  13개 사용 위치  Waltdev29
    String title = "제품 정보";  1개 사용 위치

    public void printHead() {  4개 사용 위치  Waltdev29
        System.out.println("\n========================");
        System.out.println("========" + title + "=======");
        System.out.println("========================\n");
    }

    public void printCols() {  4개 사용 위치  Waltdev29
        System.out.printf("%-8s %-12s %-8s %-10s %-16s\n", "제품번호", "제품명", "재고량", "단가", "제조업체");
        System.out.println("-------------------------------------------------------------------------------------");
    }

    @Override  4개 사용 위치  Waltdev29
    public void printItem( @NotNull Product product) {
        System.out.printf("%-10s ", product.getId());
        System.out.printf("%-12s ", product.getProductName());
        System.out.printf("%-10d ", product.getProductAmount());
        System.out.printf("%-11d ", product.getProductPrice());
        System.out.printf("%-16s ", product.getManufacturer());
    }

    public void printItemWithIndex( @NotNull Product product) {  4개 사용 위치  Waltdev29
        System.out.println("\n====== 주문 정보 ======");
        System.out.printf("1. 제품명 : %s", product.getProductName());
        System.out.printf("2. 재고량 : %d", product.getProductAmount());
        System.out.printf("3. 단가 : %d", product.getProductPrice());
        System.out.printf("4. 제조업체 : %s\n", product.getManufacturer());
    }
}
```

2-3-4. OrderView Class

```java
package mvc_jdbc_test.view;

import mvc_jdbc_test.entity.Order;

public class OrderView extends ObjectView<Order> {  13개 사용 위치  Waltdev29
    String title = "주문 정보";  1개 사용 위치

    public void printHead() {  4개 사용 위치  Waltdev29
        System.out.println("\n========================");
        System.out.println("========" + title + "========");
        System.out.println("========================\n");
    }

    public void printCols() {  4개 사용 위치  Waltdev29
        System.out.printf("%-6s %-10s %-10s %-15s %-4s %-10s\n", "주문번호", "고객아이디", "제품명", "배송지", "수량", "주문일자");
        System.out.println("-------------------------------------------------------------------------------");
    }

    @Override  4개 사용 위치  Waltdev29
    public void printItem( @NotNull Order order) {
        System.out.printf("%-7s", order.getId());
        System.out.printf("%-13s ", order.getCustomerId());
        System.out.printf("%-10s ", order.getOrderedProduct());
        System.out.printf("%-13s ", order.getDeliveryAddress());
        System.out.printf("%-5d ", order.getAmount());
        System.out.printf("%-10s ", order.getOrderDate());
    }

    public void printItemWithIndex( @NotNull Order order) {  4개 사용 위치  Waltdev29
        System.out.println("\n====== 제품 정보 ======");
        System.out.printf("1. 제품명 : %s", order.getOrderedProduct());
        System.out.printf("2. 배송지 : %s", order.getDeliveryAddress());
        System.out.printf("3. 수량 : %d", order.getAmount());
        System.out.printf("4. 주문일자 : %s\n", order.getOrderDate());
    }
}
```

2-3-5. MainView Class

```java
package mvc_jdbc_test.view;

import java.util.Scanner;

public class MainView {  26개 사용 위치  Waltdev29
    public void showHomeView() {  2개 사용 위치  Waltdev29
        System.out.println("\n###### Shop DB ######\n");
        System.out.println("--- 모드 선택 ---");
        System.out.println("0. 프로그램 종료");
        System.out.println("1. 데이터 조회");
        System.out.println("2. 데이터 추가");
        System.out.println("3. 데이터 수정");
        System.out.println("4. 데이터 삭제");
    }

    public void showMainView(String mode) {  5개 사용 위치  Waltdev29
        System.out.printf("\n###### %s 모드 ######\n\n", mode);
        System.out.printf("--- %s DB 선택 ---\n", mode);
        System.out.println("0. 뒤로가기");
        System.out.println("1. 고객 DB");
        System.out.println("2. 제품 DB");
        System.out.println("3. 주문 DB");
    }

    public static void inputEnter( @NotNull Scanner sc) {  16개 사용 위치  Waltdev29
        System.out.println("\nEnter를 눌러 돌아가기");
        sc.nextLine();
    }

    public int inputAnswer( @NotNull Scanner sc, int min, int max) {  15개 사용 위치  Waltdev29
        int answer;

        while (true) {
            try {
                System.out.print("\n입력 : ");
                String line = sc.nextLine();
                answer = Integer.parseInt( s: line);
                if (answer < min || answer > max) {
                    System.out.printf("%d~%d의 숫자를 입력해주세요.", min, max);
                    continue;
                }
                break;
            } catch (NumberFormatException e) {
                System.out.println("잘못된 입력입니다. 숫자를 입력해주세요.");
            }
        }
        return answer;
    }

    public boolean askYorN( @NotNull Scanner sc, String yesText, String yesChar, String noText, String noChar) {  14개 사용
        String answer;
        while (true) {
            System.out.printf("\n%s : %s\n%s : %s\n", yesText, yesChar, noText, noChar);
            System.out.print("\n입력 : ");
            answer = sc.nextLine();

            if (answer.equalsIgnoreCase( anotherString: yesChar)) return true;
            else if (answer.equalsIgnoreCase( anotherString: noChar)) return false;
            else System.out.println("잘못된 입력입니다. 다시 입력해주세요.\n");
        }
    }
}
```

2-3-6. InputCustomerInfoView Class

```java
package mvc_jdbc_test.view;

import mvc_jdbc_test.entity.Customer;

import java.util.Scanner;

public class InputCustomerInfoView {  4개 사용 위치  Waltdev29

    public Customer inputCustomerInfo( @NotNull Scanner sc) {  2개 사용 위치  Waltdev29
        String customerId;
        String customerName;
        int customerAge;
        String customerGrade;
        String customerJob;
        int customerPoint;

        System.out.println("\n고객 정보를 입력해주세요.");
        System.out.println("(고객 아이디, 고객 이름, 고객 나이, 고객 등급, 고객 직업, 고객 적립금)");
        System.out.print("\n고객 아이디 : ");
        customerId = sc.nextLine();
        System.out.print("\n고객 이름 : ");
        customerName = sc.nextLine();

        while (true) {
            try {
                System.out.print("\n고객 나이 : ");
                String line = sc.nextLine();
                customerAge = Integer.parseInt( s: line.trim());
                break;
            } catch (NumberFormatException e) {
                System.out.println("잘못된 입력입니다. 숫자를 입력해주세요.\n");
            }
        }

        System.out.print("\n고객 등급 : ");
        customerGrade = sc.nextLine();
        System.out.print("\n고객 직업 : ");
        customerJob = sc.nextLine();

        while (true) {
            try {
                System.out.print("\n고객 적립금 : ");
                String line = sc.nextLine();
                customerPoint = Integer.parseInt( s: line.trim());
                break;
            } catch (NumberFormatException e) {
                System.out.println("잘못된 입력입니다. 숫자를 입력해주세요.\n");
            }
        }


        System.out.println("### 입력 정보 ###");
        System.out.println("고객 아이디 : " + customerId);
        System.out.println("고객 이름 : " + customerName);
        System.out.println("고객 나이 : " + customerAge);
        System.out.println("고객 등급 : " + customerGrade);
        System.out.println("고객 직업 : " + customerJob);
        System.out.println("고객 적립금 : " + customerPoint);

        return new Customer( id: customerId, customerName, customerAge, customerGrade, customerJob, customerPoint);
    }
}
```

2-3-7. InputProductInfoView Class

```java
package mvc_jdbc_test.view;

import mvc_jdbc_test.entity.Product;

import java.util.Scanner;

public class InputProductInfoView {  4개 사용 위치  Waltdev29
    public Product inputProductInfo( @NotNull Scanner sc) {  2개 사용 위치  Waltdev29
        String productId;
        String productName;
        int productAmount;
        int productPrice;
        String manufacturer;

        System.out.println("\n제품 정보를 입력해주세요.");
        System.out.println("(제품번호, 제품명, 재고량, 단가, 제조업체)");

        System.out.print("\n제품번호 : ");
        productId = sc.nextLine();
        System.out.print("\n제품명 : ");
        productName = sc.nextLine();

        while (true) {
            try {
                System.out.print("\n재고량 : ");
                String line = sc.nextLine();
                productAmount = Integer.parseInt( s: line.trim());
                break;
            } catch (NumberFormatException e) {
                System.out.println("잘못된 입력입니다. 숫자를 입력해주세요.\n");
            }
        }

        while (true) {
            try {
                System.out.print("\n단가 : ");
                String line = sc.nextLine();
                productPrice = Integer.parseInt( s: line.trim());
                break;
            } catch (NumberFormatException e) {
                System.out.println("잘못된 입력입니다. 숫자를 입력해주세요.\n");
            }
        }

        System.out.print("\n제조업체 : ");
        manufacturer = sc.nextLine();

        System.out.println("### 입력 정보 ###");
        System.out.println("제품번호 : " + productId);
        System.out.println("제품명 : " + productName);
        System.out.println("재고량 : " + productAmount);
        System.out.println("단가 : " + productPrice);
        System.out.println("제조사 : " + manufacturer);


        return new Product( id: productId, productName, productAmount, productPrice, manufacturer);
    }
}
```

2-3-8. InputOrderInfoView Class

```java
package mvc_jdbc_test.view;

import mvc_jdbc_test.entity.Order;

import java.text.ParseException;
import java.text.SimpleDateFormat;
import java.util.Date;
import java.util.Scanner;

public class InputOrderInfoView {  4개 사용 위치  Waltdev29
    public Order inputOrderInfo( @NotNull Scanner sc) {  2개 사용 위치  Waltdev29
        String orderId;
        String customerId;
        String orderedProduct;
        int amount;
        String deliveryAddress;
        Date orderDate;

        System.out.println("\n주문 정보를 입력해주세요.");
        System.out.println("(주문 번호, 주문 고객, 주문 제품, 수량, 배송지, 주문일자)");

        System.out.print("\n주문 번호 : ");
        orderId = sc.nextLine();
        System.out.print("\n주문 고객 : ");
        customerId = sc.nextLine();
        System.out.print("\n주문 제품 : ");
        orderedProduct = sc.nextLine();

        while (true) {
            try {
                System.out.print("\n수량 : ");
                String line = sc.nextLine();
                amount = Integer.parseInt( s: line.trim());
                break;
            } catch (NumberFormatException e) {
                System.out.println("잘못된 입력입니다. 숫자를 입력해주세요.\n");
            }
        }

        System.out.print("\n배송지 : ");
        deliveryAddress = sc.nextLine();

        while (true) {
            try {
                System.out.print("\n주문일자 : ");
                String line = sc.nextLine();
                SimpleDateFormat sdf = new SimpleDateFormat( pattern: "yyyy-MM-dd");
                orderDate = sdf.parse( source: line);
                break;
            } catch (ParseException e) {
                System.out.println("잘못된 입력입니다. yyyy-MM-dd 형식으로 입력해주세요.\n");
            }
        }

        System.out.println("### 입력 정보 ###");
        System.out.println("주문 번호 : " + orderId);
        System.out.println("주문 고객 : " + customerId);
        System.out.println("주문 제품 : " + orderedProduct);
        System.out.println("수량 : " + amount);
        System.out.println("배송지 : " + deliveryAddress);
        System.out.println("주문일자 : " + orderDate);

        return new Order( id: orderId, customerId, orderedProduct, amount, deliveryAddress, orderDate);
    }
}
```

3. 결과 및 분석

4-1. 결과

1. 메인 화면

```
###### Shop DB ######

--- 모드 선택 ---
0. 프로그램 종료
1. 데이터 조회
2. 데이터 추가
3. 데이터 수정
4. 데이터 삭제

입력 : qw
잘못된 입력입니다. 숫자를 입력해주세요.

입력 : 5
0~4의 숫자를 입력해주세요.
입력 : 1
```

- 숫자입력 예외처리 적용됨

2-1. 조회 화면

```
###### 조회 모드 ######

--- 조회 DB 선택 ---
0. 뒤로가기
1. 고객 DB
2. 제품 DB
3. 주문 DB

입력 : 1
```

2-2. 고객 테이블 조회

```
========================
========고객 정보========
========================

고객아이디      고객이름      나이   등급        직업        적립금
------------------------------------------------------------------------------
apple          정소화        20    gold        학생        1000
banana         김선우        25    vip         간호사       2500
carrot         고명석        28    gold        교사        4500
orange         김용욱        22    silver      학생        0
melon          성원용        35    gold        회사원       5000
peach          오형준        0     silver      의사        300
pear           채광주        31    silver      회사원       500
qwer           qwer       10    qwer        qwer       20
strawberry     최유경        30    vip         공무원       100
grapes         허선희        0     vip         회사원       1200
moonberry      문세진        25    gold        학생        10000

========================
====== Print Done ======
========================


Enter를 눌러 돌아가기
```

3-1. 추가 화면

```
###### 추가 모드 ######

--- 추가 DB 선택 ---
0. 뒤로가기
1. 고객 DB
2. 제품 DB
3. 주문 DB

입력 : 1
```

3-2. 추가 데이터 입력

```
고객 정보를 입력해주세요.
(고객 아이디, 고객 이름, 고객 나이, 고객 등급, 고객 직업, 고객 적립금)

고객 아이디 : qwer

고객 이름 : qwer

고객 나이 : 30

고객 등급 : qwre

고객 직업 : qwer

고객 적립금 : wre
잘못된 입력입니다. 숫자를 입력해주세요.


고객 적립금 : 30
```

- 숫자입력 예외처리 적용됨

3-3. 입력 데이터 확인 및 저장

```
### 입력 정보 ###
고객 아이디 : qwer
고객 이름 : qwer
고객 나이 : 30
고객 등급 : qwre
고객 직업 : qwer
고객 적립금 : 30

데이터 저장 : S
다시 입력 : R

입력 : s
```

3-4. 입력 종료 및 데이터 확인

```
데이터가 저장되었습니다.


계속 입력 : C
입력 종료 : E

입력 : e

--- 입력 종료 ---
====== 입력 내용 ======


=========================
========고객 정보========
=========================

고객아이디      고객이름       나이   등급         직업         적립금
-------------------------------------------------------------------------------
qwer           qwer       30    qwre        qwer       30

=========================
====== Print Done ======
=========================


Enter를 눌러 돌아가기
```

4-1. 수정 화면

```
###### 수정 모드 ######

--- 수정 DB 선택 ---
0. 뒤로가기
1. 고객 DB
2. 제품 DB
3. 주문 DB

입력 : 1
```

4-2. 고객 테이블 출력

```
=========================
========고객 정보=======
=========================

고객아이디      고객이름      나이   등급         직업         적립금
-------------------------------------------------------------------------------
apple          정소화        20    gold         학생         1000
banana         김선우        25    vip          간호사        2500
carrot         고명석        28    gold         교사         4500
orange         김용욱        22    silver       학생         0
melon          성원용        35    gold         회사원        5000
peach          오형준        0     silver       의사         300
pear           채광주        31    silver       회사원        500
strawberry     최유경        30    vip          공무원        100
qwer           qwer       30    qwre         qwer       30
grapes         허선희        0     vip          회사원        1200
moonberry      문세진        25    gold         학생         10000

=========================
====== Print Done ======
=========================

고객의 고객 아이디를 입력하세요.

고객 아이디 : qwer
```

4-3. 수정할 항목 결정

```
======= 고객 정보 =======
1. 고객이름 : qwer
2. 나이 : 30
3. 등급 : qwre
4. 직업 : qwer
5. 적립금 : 30

수정할 항목의 번호를 입력해주세요. 뒤로가기 : 0

입력 : 2
```

4-4. 수정할 데이터 입력

```
수정할 나이 입력

입력 : qwe
잘못된 입력입니다. 숫자를 입력해주세요.

입력 : 25
```

- 숫자 입력 예외처리 적용됨

4-5. 데이터 수정 완료

```
수정이 완료되었습니다.

--- 수정 정보 ---
====== 고객 정보 ======
1. 고객이름 : qwer
2. 나이 : 25
3. 등급 : qwre
4. 직업 : qwer
5. 적립금 : 30

Enter를 눌러 돌아가기
```

5-1. 삭제 화면

```
###### 삭제 모드 ######

--- 삭제 DB 선택 ---
0. 뒤로가기
1. 고객 DB
2. 제품 DB
3. 주문 DB

입력 : 1
```

5-2. 삭제할 데이터 선택

```
=========================
========고객 정보=======
=========================

고객아이디      고객이름      나이   등급         직업         적립금
-------------------------------------------------------------------------------
apple          정소화        20    gold         학생         1000
banana         김선우        25    vip          간호사        2500
carrot         고명석        28    gold         교사         4500
orange         김용욱        22    silver       학생         0
melon          성원용        35    gold         회사원        5000
peach          오형준        0     silver       의사         300
pear           채광주        31    silver       회사원        500
qwer           qwer        10    qwer         qwer        20
strawberry     최유경        30    vip          공무원        100
grapes         허선희        0     vip          회사원        1200
moonberry      문세진        25    gold         학생         10000

=========================
====== Print Done ======
=========================

고객의 고객 아이디를 입력하세요.

고객 아이디 : qwer
```

5-3. 삭제 확인

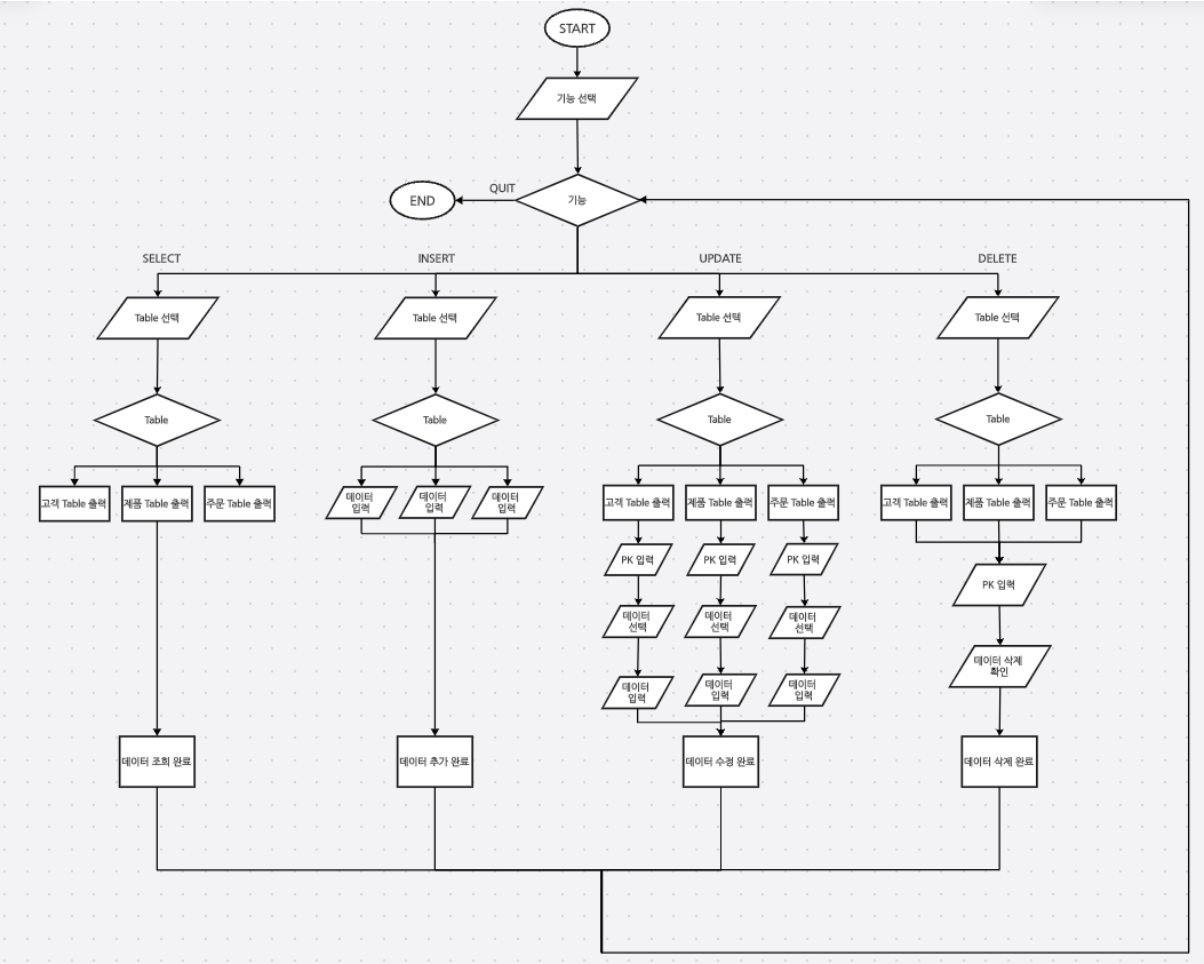
```
========================
========고객 정보=======
========================

고객아이디       고객이름       나이   등급         직업         적립금
--------------------------------------------------------------------------------
qwer           qwer       10    qwer        qwer       20

해당 데이터를 삭제하시겠습니까?

삭제 : Y
취소 : N

입력 : y
```

5-4. 데이터 삭제 완료

```
데이터가 삭제되었습니다.

Enter를 눌러 돌아가기
```

4-2. 분석

2. 처음 만들었던 mainController 는 아래와 같은 흐름으로 구현하였음.



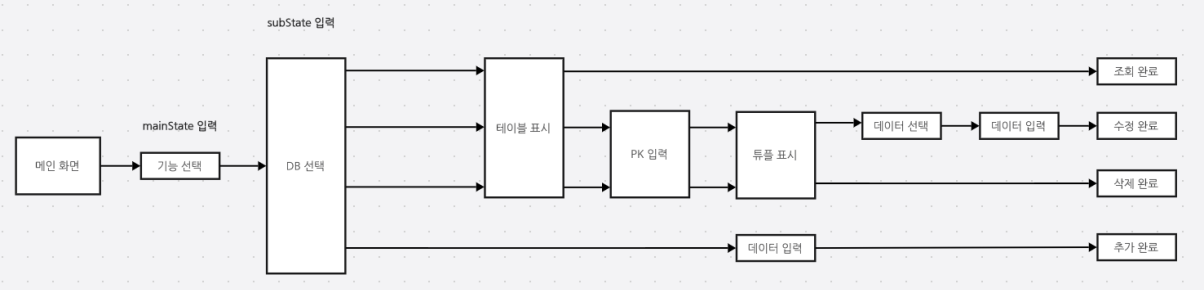
- Switch Case 로 각 기능에 따라 분기를 나누었음.

- 기능별로 분기가 나누어져 흐름을 제어하고 수정하는 것이 용이하였음.

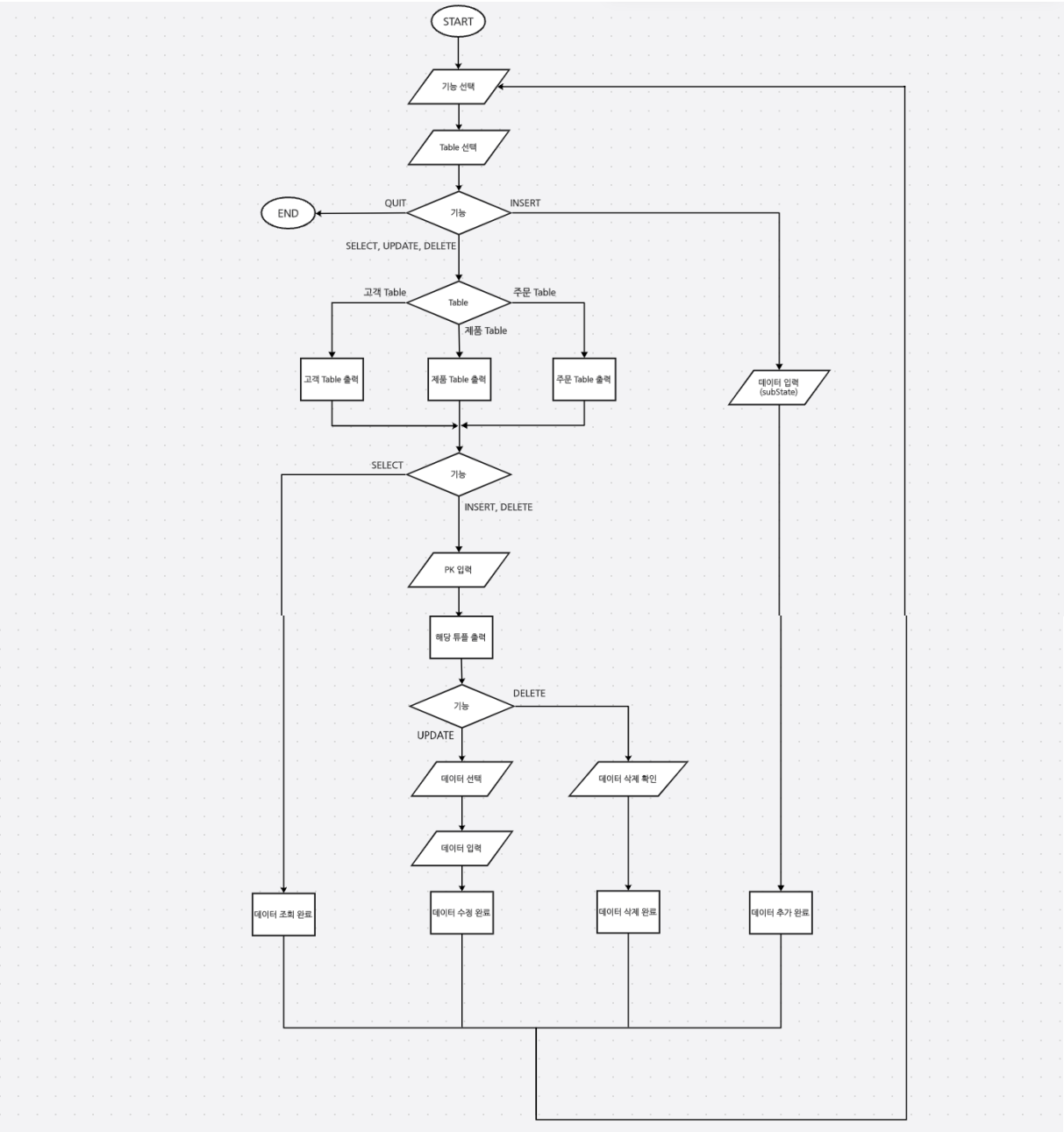
```java
        while (!quit) {
            switch (mainState) {
                // 메인
                case 0:
                    mv.showHomeView();
                    mainState = mv.inputAnswer( sc: sc,  min: 0,  max: 4);
                    if (mainState == 0) quit = true;
                    break;

                // 데이터 조회
                case 1:
                    switch (subState) {...}
                    break;

                // 데이터 추가 화면
                case 2:
                    switch (subState) {...}
                    break;

                // 데이터 수정 화면
                case 3:
                    switch (subState) {...}
                    break;

                // 데이터 삭제 화면
                case 4:
                    switch (subState) {...}
                    break;

                default:
                    System.out.println("잘못된 입력입니다. 1~3의 정수를 입력해주세요.");
            }
            if (mainState == -1) break;
        }

        System.out.println("\n프로그램을 종료합니다.");
        sc.close();
    }
```

[MainController main 함수]

- 하지만 분기별로 겹치는 프로세스가 있어 중복을 최대한 줄이고자 시스템 흐름을 아래와 같이 도식화 함.



3. 도식을 기반으로 중복을 줄여 코드를 재구성하였음.



- 이전의 코드가 처음부터 Switch Case로 분기가 나뉘어졌다면, 이번 코드는 흐름이 진행되며 if문을 통해 분기가 나누어지는 구조로 재구성 함.

- 코드의 흐름을 바꾸며 기능이 겹치는 함수들을 통합할 수 있었음.

- 중복을 많이 제거하여 코드의 효율성이 향상되었으며, 코드 라인 수가 100라인가량 짧아졌음.

3. 개발 중간부터 중복을 최소화하기 위해 추상화와 제네릭을 공부하였음.

Entity들을 다형성을 활용하여 관리하고 클래스, 함수들을 제네릭을 활용하여 하나로 통합하려 하였으나 중간부터 수정하려 하니 한계가 있었음.

설계부터 유연성을 고려하며 설계하는 것이 중요하다고 느꼈음.