[스마트 웹 & 앱 과정]

SQL 활용 포트폴리오

김문선

2019.07.02

목차

1. 프로그램 개요

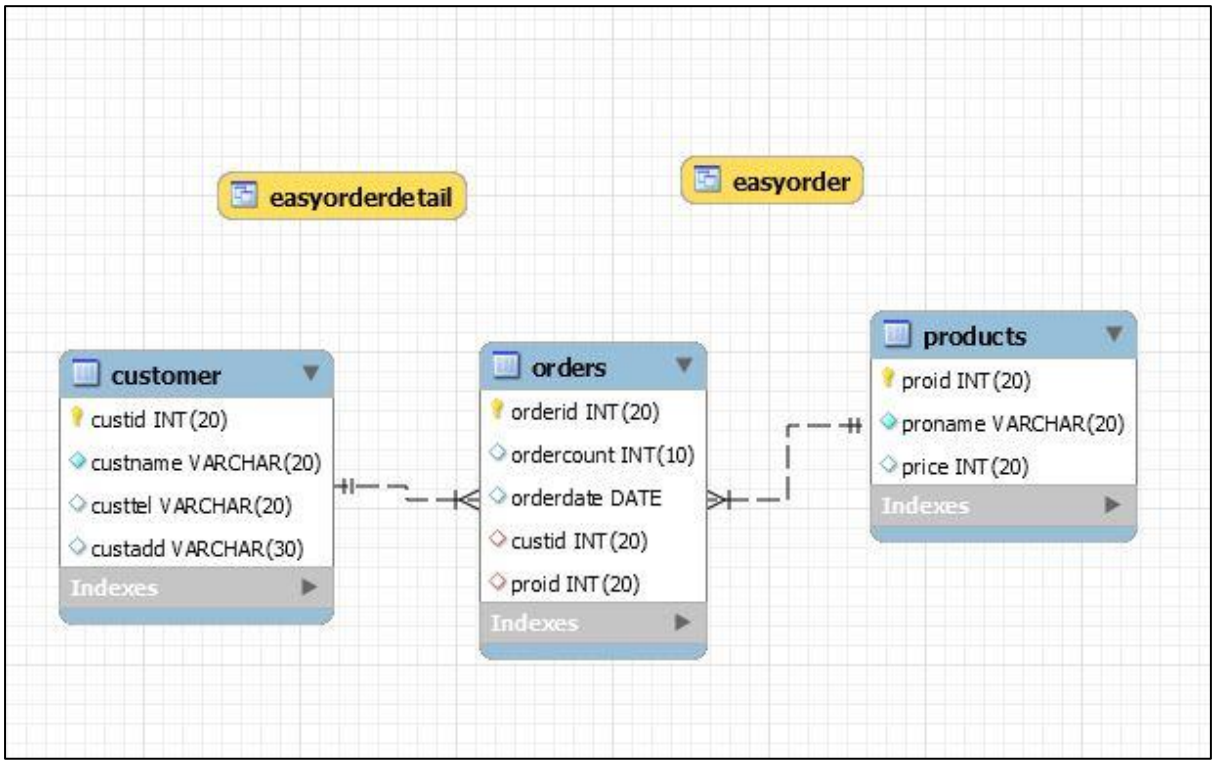
본 프로그램은 데이터베이스 'shopping' 관리를 위한 MySQL, JAVA 연동 프로그램입니다. MySQL에서 조회(출력)(select), 입력(insert), 삭제(delete), 수정(update)로 데이터베이스를 관리하려면 쿼리를 일일이 작성해야하는 불편함이 생기게 됩니다. 이를 불편함을 해소하고, 쉽고 간단하게 데이터베이스를 관리할 수 있도록 JAVA 프로그램을 제작하게 되었습니다.

해당 프로그램은 MySQL에서 필요한 쿼리를 일일이 작성 안 해도 되는 편리성 그리고 여러기능들을 메뉴로 구현, 정보들을 사용자가 간편하게 번호를 선택해 편집할 수 있는 쉬운 UI를 가지고 있습니다.

프로그램의 기능은 기존 테이블 관리 기능에서 다양한 기능들을 추가했습니다. 추가 기능은 권한기능, 계정 확인과 삭제 기능, 뷰 출력 기능 총 3가지가 있습니다.

추가기능 중 첫 번째 권한 기능에서는 세부 메뉴를 통해 권한 부여, 확인, 회수 기능을 실행할 수 있습니다. 두 번째 계정 확인과 삭제 기능에서는 모든 계정을 출력하고 삭제하고 싶은 계정을 선택해 삭제할 수 있습니다. 세 번째 뷰 출력 기능은 MySQL에서 shopping 데이터베이스 내 테이블 3개를 조인하여 만든 2개의 뷰를 확인할 수 있게 뷰 내용을 출력할 수 있습니다.

2. ERD



3. 테이블 내용

해당 프로그램에서 연동한 테이블(데이터베이스, 뷰)을 소개합니다.

(1) 데이터베이스

1) 데이터베이스명 : shopping

2) 생성 쿼리

```
create database shopping
    default character set utf8
    default collate utf8_general_ci;
```

(2) 테이블

1) 고객정보테이블

① 테이블명 : customer

② 생성 쿼리

```
create table customer (
    custid int(20) not null primary key,
    custname varchar(20) not null,
    custtel varchar(20),
    custadd varchar(30)
);
```

③ 테이블 구조

➢ desc customer ;

Field	Type	Null	Key	Default	Extra
custid	int(20)	NO	PRI	NULL	
custname	varchar(20)	NO		NULL	
custtel	varchar(20)	YES		NULL	
custadd	varchar(30)	YES		NULL	

➔ custid : 고객id custname : 고객명 custtel : 고객전화번호 custadd : 고객주소

④ 테이블 내용

➢ select * from customer ;

custid	custname	custtel	custadd
1	오하나	010-2245-3344	경기도 용인시
2	김지혜	010-6643-9864	경기도 성남시
3	최수정	010-9258-3821	서울 강남구
4	이혜진	010-4422-1133	전남 해남군
5	윤정선	010-5544-2238	독도는 우리땅
6	김문애	010-1234-4332	제주도 제주시

2) 상품정보테이블

① 테이블명 : products

② 생성 쿼리

```
create table products (
    proid int(20) not null primary key,
    proname varchar(20) not null,
    price int(20)
);
```

③ 테이블 구조

➢ desc products ;

Field	Type	Null	Key	Default	Extra
proid	int(20)	NO	PRI	NULL	
proname	varchar(20)	NO		NULL	
price	int(20)	YES		NULL	

➔ proid : 상품ID proname : 상품명 price : 가격

④ 테이블 내용

➢ select * from products ;

proid	proname	price
1	다리미	60000
2	텀블러	12000
3	파우치	4000
4	핸드폰케이스	13000
5	충전기	8000
101	열쇠고리	5000

3) 주문정보테이블

① 테이블명 : orders

② 생성 쿼리

```
create table orders (
    orderid int(20) not null primary key,
    ordercount int(10),
    orderdate varchar(20),
    custid int(20),
    proid int(20),
    foreign key(custid) references customer(custid),
    foreign key(proid) references products(proid)
);
```

③ 테이블 구조

➢ desc orders ;

Field	Type	Null	Key	Default	Extra
orderid	int(20)	NO	PRI	NULL	
ordercount	int(10)	YES		NULL	
orderdate	varchar(20)	YES		NULL	
custid	int(20)	YES	MUL	NULL	
proid	int(20)	YES	MUL	NULL	

➔ orderid : 주문 번호 ordercount : 주문 수량 orderdate : 주문 날짜
custid : 주문 고객 ID proid : 주문 상품 ID

④ 테이블 내용

➢ select * from orders ;

orderid	ordercount	orderdate	custid	proid
1	2	2019-06-02	1	2
2	1	2019-06-05	3	1
3	3	2019-06-08	2	3
4	1	2019-06-11	4	5
5	2	2019-06-20	5	4
15	2	2019-06-06	3	1

(3) 뷰

1) 주문간단정보 뷰

① 뷰 명 : easyorder

② 생성 쿼리

```
create view easyorder as
select  o.custid as '고객아이디', c.custname as '고객이름',
        o.proid as '상품아이디', p.proname as '상품명'
from customer c, products p, orders o
where c.custid = o.custid
and p.proid = o.proid;
```

③ 뷰 구조

➢ desc easyorder ;

Field	Type	Null	Key	Default	Extra
고객아이디	int(20)	YES		NULL	
고객이름	varchar(20)	NO		NULL	
상품아이디	int(20)	YES		NULL	
상품명	varchar(20)	NO		NULL	

④ 뷰 내용

➢ select * from easyorder ;

고객아이디	고객이름	상품아이디	상품명
1	오하나	2	텀블러
3	최수정	1	다리미
2	김지혜	3	파우치
4	이혜진	5	충전기
5	윤정선	4	핸드폰케이스
3	최수정	1	다리미

⑤ 뷰 생성 목적

주문간단정보 “easyorder” 뷰를 생성한 목적은 테이블만으로 주문 정보를 한 눈에 확인할 수 있는 방법이 없어서 해당 뷰를 생성했습니다. ‘orders’ 테이블로 주문정보를 확인할 수 있지만, 고객명과 상품명이 조회되지않아 어떤 고객이 어떤 상품을 주문했는지 확인하기가 번거롭습니다. 그래서 ‘orders’, ‘products’, ‘customer’ 3개 테이블을 조인하여 누가 어떤 상품을 주문했는지 한 눈에 들어오게 주문간단정보 뷰를 만들었습니다.

2) 주문상세정보 뷰

① 뷰 명 : easyorderdetail

② 생성 쿼리

```
create view easyorderdetail as
select concat(c.custname,'님께서 ',p.proname,' 상품을 ',o.ordercount,'개 주문으로 ',
    o.orderdate,'일자의 총 주문금액은 ',o.ordercount*p.price,'원 입니다.') as '주문정보설명'
from customer c, products p, orders o
where c.custid = o.custid
and p.proid = o.proid;
```

③ 뷰 구조

➢ desc easyorderdetail ;

Field	Type	Null	Key	Default	Extra
주문정보설명	varchar(135)	YES		NULL	

④ 뷰 내용

➢ select * from easyorderdetail ;

주문정보설명
오하나님께서 텀블러 상품을 2개 주문으로 2019-06-02일자의 총 주문금액은 24000원 입니다.
최수정님께서 다리미 상품을 1개 주문으로 2019-06-05일자의 총 주문금액은 60000원 입니다.
김지혜님께서 파우치 상품을 3개 주문으로 2019-06-08일자의 총 주문금액은 12000원 입니다.
이혜진님께서 충전기 상품을 1개 주문으로 2019-06-11일자의 총 주문금액은 8000원 입니다.
윤정선님께서 핸드폰케이스 상품을 2개 주문으로 2019-06-20일자의 총 주문금액은 26000원 입니다.
최수정님께서 다리미 상품을 2개 주문으로 2019-06-06일자의 총 주문금액은 120000원 입니다.

⑤ 뷰 생성 목적

주문상세정보 'easyorderdetail' 뷰를 생성한 목적은 주문간단정보 'easyorder' 뷰 생성 목적이랑 비슷합니다. 조인을 연습삼아 고객이 주문한 정보를 서술형으로 출력되게 하는 뷰를 생성해봤습니다. 해당 뷰에서 관리자가 어떤 고객이 무엇을 언제 몇 개 주문했는지 총 주문금액은 얼마인지를 서술형으로 가독성이 높게 생성해봤습니다. (고객도 언제 고객이 어떤 상품을 주문했는지 총 지불해야 할 금액을 알 수 있습니다.)

4. 소스 코드

(1) Process.java

```
import java.util.*;

public class Process {
	Scanner scan = new Scanner(System.in);
	Scanner scan2 = new Scanner(System.in);
	customerInfo customer;
	ordersInfo orders;
	productsInfo products;

	public Process() {
		System.out.println("테이블에 입력 값을 넣어주세요!!!");

		customer = new customerInfo();
		orders = new ordersInfo();
		products = new productsInfo();
	}

	customerInfo inputInfo() {
		System.out.print("> 고객 ID 입력하세요 : ");
		customer.setCustid(scan.nextInt());
		System.out.print("> 고객 이름 입력하세요 : ");
		customer.setCustname(scan2.nextLine());
		System.out.print("> 고객 전화번호를 입력하세요 : ");
		customer.setCusttel(scan.next());
		System.out.print("> 고객 주소를 입력하세요 : ");
		customer.setCustadd(scan2.nextLine());

		return customer;
	}

	productsInfo inputInfo2() {
		System.out.print("> 상품 ID 입력하세요 : ");
		products.setProid(scan.nextInt());
		System.out.print("> 상품명 입력하세요 : ");
		products.setProname(scan2.nextLine());
		System.out.print("> 상품 가격 입력하세요 : ");
		products.setPrice(scan.nextInt());
```

```
                return products;
        }

        ordersInfo inputInfo3() {
                System.out.print("> 주문 번호 입력하세요 : ");
                orders.setOrderid(scan.nextInt());
                System.out.print("> 상품 주문 개수 입력하세요 : ");
                orders.setOrdercount(scan.nextInt());
                System.out.print("> 주문일자 입력하세요 : ");
                orders.setOrderdate(scan2.nextLine());
                System.out.print("> 주문 고객 ID 입력하세요 : ");
                orders.setCustid(scan.nextInt());
                System.out.print("> 주문 상품 ID 입력하세요 : ");
                orders.setProid(scan.nextInt());

                return orders;
        }
}
```

(2) productsInfo.java

```
public class productsInfo {
        private String proname;
        private int proid, price;

        public productsInfo() {

        }

        public String getProname() {
                return proname;
        }

        public void setProname(String proname) {
                this.proname = proname;
        }

        public int getProid() {
                return proid;
        }

        public void setProid(int proid) {
                this.proid = proid;
        }

        public int getPrice() {
                return price;
        }

        public void setPrice(int price) {
                this.price = price;
        }
}
```

(3) ordersInfo.java

```
public class ordersInfo {
	private String orderdate;
	private int orderid, ordercount, custid, proid;

	public ordersInfo() {

	}

	public String getOrderdate() {
		return orderdate;
	}

	public void setOrderdate(String orderdate) {
		this.orderdate = orderdate;
	}

	public int getOrderid() {
		return orderid;
	}

	public void setOrderid (int orderid) {
		this.orderid = orderid;
	}

	public int getOrdercount() {
		return ordercount;
	}

	public void setOrdercount (int ordercount) {
		this.ordercount = ordercount;
	}

	public int getCustid() {
		return custid;
	}

	public void setCustid (int custid) {
		this.custid = custid;
	}

	public int getProid() {
		return proid;
	}

	public void setProid(int proid) {
		this.proid = proid;
	}
}
```

(4) customerInfo.java

```
public class customerInfo {
	private String custname, custtel, custadd;
	private int custid;

	public customerInfo() {

	}

	public String getCustname() {
		return custname;
	}

	public void setCustname(String custname) {
		this.custname = custname;
	}

	public String getCusttel() {
		return custtel;
	}

	public void setCusttel(String custtel) {
		this.custtel = custtel;
	}

	public String getCustadd() {
		return custadd;
	}

	public void setCustadd(String custadd) {
		this.custadd = custadd;
	}

	public int getCustid() {
		return custid;
	}

	public void setCustid(int custid) {
		this.custid = custid;
	}
}
```

(5) Main.java

```
import java.util.*;

public class Main {
    public static void main(String[] args) {
       boolean check = true;
       int menu;
       Scanner scan = new Scanner(System.in);
       Scanner scan2 = new Scanner(System.in);
       Process pro;
       DBConnection connection = new DBConnection();
       customerInfo customer;  productsInfo products;  ordersInfo orders;
```

```java
System.out.println(" ▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒ ");
System.out.println(" ▒                                ▒ ");
System.out.println("▒            데이터베이스 관리 프로그램           ▒");
System.out.println(" ▒                                ▒");
System.out.println(" ▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒  ");
System.out.println("");

while (check) {
    System.out.println("  ▦▦▦▦▦▦▦▦▦▦▦▦▦▦▦▦▦▦▦▦▦▦▦▦▦▦▦▦▦▦▦▦▦▦▦▦▦▦▦▦▦▦▦▦▦▦▦▦▦▦▦▦");
    System.out.println("");
    System.out.println(" \t\t\t         ※  Select Menu ※");
    System.out.println("");
    System.out.println("1. Data 입력\t 2. Data 출력\t\t 3. Data 삭제\t 4. Data 수정\t\n"
                  + "    5. 권한부여 \t 6. 계정 확인 및 삭제 \t 7. 뷰 출력 \t 8. 종료");
    System.out.println("");
    System.out.println("  ▦▦▦▦▦▦▦▦▦▦▦▦▦▦▦▦▦▦▦▦▦▦▦▦▦▦▦▦▦▦▦▦▦▦▦▦▦▦▦▦▦▦▦▦▦▦▦▦▦▦▦");
    System.out.print(" >> 작업을 선택하세요 :  ");
    menu = scan.nextInt();
    int tablechoice, viewchoice;

       switch (menu) {
       case 1: // 데이터 입력 완료!!
          System.out.println(" => 1번 'Data 입력'을 선택했습니다.");
          System.out.println("");
          System.out.println("▶▶ ▶ Data 입력 화면(Insert) ◀◀◀ ");
          System.out.println(" :: 총 3개의 테이블이 있습니다.");
          System.out.println(" :: 1. 고객테이블 \t 2. 상품테이블 \t 3.주문정보테이블");
          System.out.print(" >> 입력할 테이블을 선택하세요 : ");
          tablechoice = scan.nextInt();
          System.out.println("=========================");

          switch (tablechoice) {
          case 1: // customer table
            System.out.println("☞ 고객정보테이블(customer)을 선택했습니다.");
            System.out.println("=========================");
            pro = new Process();
            customer = pro.inputInfo();
            connection.input(customer);
            System.out.println("");
            break;

          case 2: // products table
            System.out.println("☞ 상품정보테이블(products)을 선택했습니다.");
            System.out.println("=========================");
            pro = new Process();
            products = pro.inputInfo2();
            connection.input(products);
            System.out.println("");
            break;

          case 3: // orders table
            System.out.println("☞ 주문정보테이블(orders)을 선택했습니다.");
            System.out.println("=========================");
            pro = new Process();
            orders = pro.inputInfo3();
```

```
                connection.input(orders);
                System.out.println("");
                break;
                }
              break;

            case 2: // 정보 출력 select 완료!!!
              System.out.println(" => 2번 'Data 출력'을 선택했습니다.");
              System.out.println("");
              System.out.println("▶▶ ▶ Data 출력 화면(Select) ◀◀◀ ");
              System.out.println(" :: 총 3개의 테이블이 있습니다.");
              System.out.println(" :: 1. 고객테이블 \t 2. 상품테이블 \t 3.주문정보테이블");
              System.out.print(" >> 출력할 테이블을 선택하세요 : ");
              tablechoice = scan.nextInt();
              System.out.println("==========================");

              switch (tablechoice) {
              case 1: // customer table
                System.out.println("☞ 고객정보테이블(customer)을 선택했습니다.");
                System.out.println("==========================");
                System.out.println(" ==> customer table을 출력합니다.");
                System.out.println("");
                System.out.println(" 고객ID \t 고객명\t    고객전화번호 \t    고객주소");
                System.out.println("====================================");
                connection.output();
                break;
              case 2: // products table
                System.out.println(" ☞ 상품정보테이블(products)을 선택했습니다.");
                System.out.println("==========================");
                System.out.println(" ==> products table을 출력합니다.");
                System.out.println("");
                System.out.println("상품ID \t       상품명\t    가격 \t");
                System.out.println("====================================");
                connection.output2();
                break;
             case 3: // orders table
               System.out.println(" ☞ 주문정보테이블(orders)을 선택했습니다.");
               System.out.println("==========================");
               System.out.println(" ==> orders table을 출력합니다.");
               System.out.println("");
               System.out.println("주문번호 \t주문개수\t 주문날짜    주문고객ID   주문상품ID");
               System.out.println("====================================");
               connection.output3();
               break;
             }
            break;
            case 3:
              System.out.println(" => 3번 'Data 삭제'를 선택했습니다.");
              System.out.println("");
              System.out.println("▶▶ ▶ Data 삭제 화면(Delete) ◀◀◀ ");
              System.out.println(" :: 총 3개의 테이블이 있습니다.");
              System.out.println("::1.고객테이블 \t 2. 상품테이블 \t 3.주문정보테이블");
              System.out.print(" >> 삭제할 컬럼이 있는 테이블을 선택하세요 : ");
```

```
                tablechoice = scan.nextInt();
                System.out.println("==========================");

                switch (tablechoice) {
                case 1: // customer table
                  System.out.println(" ☞ 고객정보테이블(customer)을 선택했습니다.");
                  System.out.println("==========================");
                  System.out.println(" @ customer table 컬럼삭제 화면입니다 @");
                  System.out.print(" > 고객정보의 삭제할 고객ID를 입력하세요: ");
                  connection.delete(scan.nextInt());
                  System.out.println("");
                  break;

                case 2: // products table
                  System.out.println(" ☞ 상품정보테이블(products)을 선택했습니다.");
                  System.out.println("==========================");
                  System.out.println("  @ products table 컬럼삭제 화면입니다 @");
                  System.out.print(" > 상품정보의 삭제할 상품ID를 입력하세요: ");
                  connection.delete2(scan.nextInt());
                  System.out.println("");
                  break;

                case 3: // orders table
                  System.out.println(" ☞ 주문정보테이블(orders)을 선택했습니다.");
                  System.out.println("==========================");
                  System.out.println(" @ orders table 컬럼삭제 화면입니다 @ ");
                  System.out.print(" > 주문정보의 삭제할 주문번호를 입력하세요: ");
                  connection.delete3(scan.nextInt());
                  System.out.println("");
                  break;
                }
                break;

                case 4: //  update..
                  System.out.println(" => 3번 'Data 수정'을 선택했습니다.");
                  System.out.println("");
                  System.out.println("▶▶ ▶ Data 수정 화면(Update) ◀◀◀ ");
                  System.out.println(" :: 총 3개의 테이블이 있습니다.");
                  System.out.println(" :: 1. 고객테이블 \t 2. 상품테이블 \t 3.주문정보테이블");
                  System.out.print(" >> 수정할 테이블을 선택하세요 : ");
                  tablechoice = scan.nextInt();
                  System.out.println("==========================");

                  switch (tablechoice) {
                  case 1: // customer update
                    System.out.println("☞ 고객정보테이블(customer)을 선택했습니다.");
                    System.out.println("==========================");
                    System.out.println(" @ customer table 수정(update) 화면입니다 @");
                    connection.update();
                    break;
                  case 2: // products update
                    System.out.println(" ☞ 상품정보테이블(products)을 선택했습니다.");
                    System.out.println("==========================");
                    System.out.println(" @ products table 수정(update) 화면입니다 @ ");
```

```
                connection.update2();
                break;
            case 3: // orders update
                System.out.println(" ☞ 주문정보테이블(orders)을 선택했습니다.");
                System.out.println("=========================");
                System.out.println(" @ orders table 수정(update) 화면입니다 @");
                connection.update3();
                break;
            }
            break;
            case 5: // grant,revoke
              System.out.println(" => 5번 '권한부여'를 선택했습니다.");
              System.out.println("");
              System.out.println("▶▶ ▶ 권한 부여/확인/회수 화면(Grant/Revoke) ◀◀◀ ");
              connection.grant();
              System.out.println("");
              break;

            case 6:
              System.out.println(" => 6번 '계정 확인 및 삭제'를 선택했습니다.");
              System.out.println("");
              System.out.println("▶▶ ▶ 계정 확인 및 삭제 화면 ◀◀◀ ");
              connection.accountcontent();
              System.out.println("");
              break;
            case 7:
              System.out.println(" => 7번 '뷰 출력'을 선택했습니다.");
              System.out.println("");
              System.out.println("▶▶ ▶ 뷰 출력 화면 ◀◀◀ ");
              System.out.println(":: 총 2개의 뷰가 있습니다.");
              System.out.println("::1.주문간단정보 뷰(easyorder)\t
                                  2.주문상세정보 뷰(easyorderdetail)");
              System.out.print(" >> 출력할 뷰를 선택하세요 : ");
              viewchoice = scan.nextInt();
              System.out.println("=========================");

              switch (viewchoice) {
              case 1: // easyorder
                System.out.println("☞ 주문간단정보 뷰(easyorder)를 선택했습니다.");
                System.out.println(" # easyorder view를 출력합니다.");
                System.out.println("");
                System.out.println(" 고객ID    고객이름    상품ID    상품명");
                System.out.println("=================================");
                connection.view_output();
                break;
              case 2: // easyorderdetail
                System.out.println("☞ 주문상세정보 뷰(easyorderdetail)를 선택했습니다.");
                System.out.println(" # easyorderdetail view를 출력합니다");
                System.out.println("");
                System.out.println(" \t\t\t\t주문정보설명");
                System.out.println("=================================");
                connection.view_output2();
                break;
              }
```

```
                    break;
                case 8:
                    System.out.println("프로그램을 종료합니다!!");
                    check = false;
                    break;
                default:
                    System.out.println("메뉴를 잘못 선택하셨습니다. 다시 선택해주세요. \n");
                    break;
                }
            }
            scan.close();
            scan2.close();
            connection.end();
    }
}
```

(6) DBConnection.java

```
import java.sql.*; import java.util.*;

public class DBConnection {
    private Connection con; private Statement st;
    private ResultSet rs;
    Scanner scan;     Scanner scan2;

    public DBConnection() {
        try {
              Class.forName("com.mysql.jdbc.Driver");
              String url = "jdbc:mysql://localhost:3306/shopping?"
                           + "autoReconnect=true&useUnicode=true&"
                           + "characterEncoding=utf8";
              con = DriverManager.getConnection(url, "root", "1234");
              st = con.createStatement();
        } catch (Exception e) {
              System.out.println("데이터베이스 연결오류: " + e.getMessage());
        }
    }

    public void input(customerInfo customer) {
         try {
              String SQL = "insert into customer " + "values ("+
                     customer.getCustid()+ ",'" + customer.getCustname() + "','" +
                     customer.getCusttel()+ "','" + customer.getCustadd() + "');";
              st.executeUpdate(SQL);
         } catch (Exception e) {
               System.out.println("데이터베이스 customer 입력 오류 : " + e.getMessage());
         }
     }

     public void input(ordersInfo orders) {
          try {
               String SQL = "insert into orders " + "values (" +
                    orders.getOrderid()+ "," + orders.getOrdercount() + ",'"
                    + orders.getOrderdate() + "','"+ orders.getCustid() + ","
                    + orders.getProid() + ");";
               st.executeUpdate(SQL);
```

```
        } catch (Exception e) {
              System.out.println("데이터베이스 orders 입력 오류 : " + e.getMessage());
        }
    }

    public void input(productsInfo products) {
         try {
              String SQL = "insert into products " + "values ("
                    + products.getProid()+ ",'" + products.getProname()+ "',"
                    + products.getPrice() + ");";
              st.executeUpdate(SQL);
         } catch (Exception e) {
               System.out.println("데이터베이스 products 입력 오류 : " + e.getMessage());
         }
    }

    public void output() {
        String custname, custtel, custadd;
        int custid;

        try {
              rs = st.executeQuery("select * from customer;");
              if (rs == null) {
                  System.out.println("저장된 데이터가 없습니다.");
              } else {
                  while (rs.next()) {
                       custid = rs.getInt("custid");
                       custname = rs.getString("custname");
                       custtel = rs.getString("custtel");
                       custadd = rs.getString("custadd");
                       System.out.printf("%5d \t %3s \t %8s \t %5s",
                                         custid, custname, custtel, custadd);
                       System.out.println();
                  }
                  System.out.println("");
               }
         } catch (Exception e) {
              System.out.println("데이터베이스 customer 출력 오류 : " + e.getMessage());
         }
    }

    public void output2() { // products
       String proname;
       int proid, price;

       try {
            rs = st.executeQuery("select * from products;");
            if (rs == null) {
                System.out.println("저장된 데이터가 없습니다.");
            } else {
                while (rs.next()) {
                     proid = rs.getInt("proid");
                     proname = rs.getString("proname");
                     price = rs.getInt("price");
                    System.out.printf(" %3d \t %10s \t %5d", proid, proname, price);
                    System.out.println();
```

```
            }
            System.out.println("");
        }
    } catch (Exception e) {
        System.out.println("데이터베이스 products 출력 오류 : " + e.getMessage());
    }
}

public void output3() { // orders
    String orderdate;
    int orderid, proid, custid, ordercount;

    try {
        rs = st.executeQuery("select * from orders;");
        if (rs == null) {
            System.out.println("저장된 데이터가 없습니다.");
        } else {
            while (rs.next()) {
                orderid = rs.getInt("orderid");
                ordercount = rs.getInt("ordercount");
                orderdate = rs.getString("orderdate");
                custid = rs.getInt("custid");
                proid = rs.getInt("proid");
                System.out.printf("%3d \t %3d \t %9s \t %d \t %2d",
                        orderid, ordercount, orderdate, custid, proid);
                System.out.println();
            }
            System.out.println("");
        }
    } catch (Exception e) {
        System.out.println("데이터베이스 orders 출력 오류 : " + e.getMessage());
    }
}

public void delete(int custid) {
    try {
        String SQL = "delete from customer where custid = " + custid + ";";
        st.executeUpdate(SQL);
    } catch (SQLException e) {
        System.out.println("데이터베이스 customer 삭제 오류 : " + e.getMessage());
    }
}

public void delete2(int proid) {
    try {
        String SQL = "delete from products where proid = " + proid + ";";
        st.executeUpdate(SQL);
    } catch (SQLException e) {
        System.out.println("데이터베이스 products 삭제 오류 : " + e.getMessage());
    }
}

public void delete3(int orderid) {
    try {
        String SQL = "delete from orders where orderid = " + orderid + ";";
        st.executeUpdate(SQL);
```

```
        } catch (SQLException e) {
            System.out.println("데이터베이스 orders 삭제 오류 : " + e.getMessage());
        }
    }

    public void update() {
        scan = new Scanner(System.in);
        scan2 = new Scanner(System.in);
        String sql;
        int menu, custid, modify2;
        String modify1;

        // 기준은 아이디로 지정한다..
        System.out.println("1.고객명  2.고객전화번호  3.고객주소");
        System.out.print("수정할 필드명을 선택하세요: >>");
        menu = scan.nextInt();
        try {
            switch (menu) {
            case 1: // 1번 고객명 수정
              System.out.print(" + 수정할 고객명을 입력하세요:>> ");
              modify1 = scan2.nextLine();
              System.out.print(" + 수정할 고객id를 입력하세요:>> ");
              custid = scan.nextInt();
              sql = "update customer set custname='" + modify1+ "' where custid="
                     + custid + ";";
              st.executeUpdate(sql);
              break;

            case 2: // 2번 고객전화번호 수정
              System.out.print(" + 수정할 고객전화번호를 입력하세요:>> ");
              modify1 = scan2.nextLine();
              System.out.print(" + 수정할 고객id를 입력하세요:>> ");
              custid = scan.nextInt();
              sql = "update customer set custtel='" + modify1
                     + "' where custid=" + custid + ";";
              st.executeUpdate(sql);
              break;

            case 3: // 3번 고객 주소 수정
              System.out.print(" + 수정할 고객 주소를 입력하세요:>> ");
              modify1 = scan2.nextLine();
              System.out.print(" + 수정할 고객id를 입력하세요:>> ");
              custid = scan.nextInt();
              sql = "update customer set custadd='" + modify1+ "' where custid="
                     + custid + ";";
              st.executeUpdate(sql);
              break;
            }
        } catch (Exception e) {
            System.out.println("데이터베이스 customer update 오류 : " + e.getMessage());
        }
    }
```

```
    public void update2() { // update문
        scan = new Scanner(System.in);
        scan2 = new Scanner(System.in);
        String sql;
        int menu, proid, modify2;
        String modify1;

        // 기준은 아이디로 지정한다..
        System.out.println(" 1.상품명  2.상품가격");
        System.out.print("수정할 필드명을 선택하세요: >>");
        menu = scan.nextInt();
        try {
            switch (menu) {
            case 1: // proname
              System.out.print(" + 수정할 상품명을 입력하세요:>> ");
              modify1 = scan2.nextLine();
              System.out.print(" + 수정할 상품id를 입력하세요:>> ");
              proid = scan.nextInt();
              sql = "update products set proname='" + modify1+ "' where proid="
                     + proid + ";";
              st.executeUpdate(sql);
              break;

            case 2: // price
              System.out.print(" + 수정할 상품가격를 입력하세요:>> ");
              modify2 = scan2.nextInt();
              System.out.print(" + 수정할 상품id를 입력하세요:>> ");
              proid = scan.nextInt();
              sql = "update products set price='" + modify2 + "' where proid="
                     + proid + ";";
              st.executeUpdate(sql);
              break;
            }
        } catch (Exception e) {
              System.out.println("데이터베이스 products update 오류 : " + e.getMessage());
        }
    }

     // 4번 데이터 수정... orders
    public void update3() { // update문
        scan = new Scanner(System.in);
        scan2 = new Scanner(System.in);
        String sql;
        int menu, orderid, modify2;
        String modify1;

        // 기준은 아이디로 지정한다..
        System.out.println(" 1.주문수량  2.주문날짜  3.주문고객ID 4.주문상품ID");
        System.out.print(" 수정할 필드명을 선택하세요: >> ");
        menu = scan.nextInt();
        try {
            switch (menu) {
            case 1: // ordercount
              System.out.print(" + 수정할 주문수량을 입력하세요:>> ");
```

```
            modify2 = scan2.nextInt();
            System.out.print(" + 수정할 주문번호를 입력하세요:>> ");
            orderid = scan.nextInt();
            sql = "update orders set ordercount=" + modify2 + " where orderid="
                  + orderid + ";";
            st.executeUpdate(sql);
            break;

          case 2: // orderdate
            System.out.print(" + 수정할 주문날짜를 입력하세요:>> ");
            modify1 = scan2.nextLine();
            System.out.print(" + 수정할 주문번호를 입력하세요:>> ");
            orderid = scan.nextInt();
            sql = "update orders set orderdate='" + modify1 + "' where orderid="
                  + orderid + ";";
            st.executeUpdate(sql);
            break;

          case 3: // custid
            System.out.print(" + 수정할 주문고객ID를 입력하세요:>> ");
            modify2 = scan2.nextInt();
            System.out.print(" + 수정할 주문번호를 입력하세요:>> ");
            orderid = scan.nextInt();
            sql = "update orders set custid=" + modify2 + " where orderid="
                  + orderid + ";";
            st.executeUpdate(sql);
            break;

          case 4: // proid
            System.out.print(" + 수정할 주문상품ID를 입력하세요:>> ");
            modify2 = scan2.nextInt();
            System.out.print(" + 수정할 주문번호를 입력하세요:>> ");
            orderid = scan.nextInt();
            sql = "update orders set proid=" + modify2 + " where orderid="
                  + orderid + ";";
            st.executeUpdate(sql);
            break;
          }
      } catch (Exception e) {
            System.out.println("데이터베이스 orders update 오류 : " + e.getMessage());
      }
  }

  public void grant() { // 권한 부여,회수,확인
      scan = new Scanner(System.in);
      scan2 = new Scanner(System.in);
      System.out.println("1. 권한부여 \t 2. 권한 확인 \t 3. 권한 회수");
      System.out.print(" 메뉴 번호를 선택해주세요>> ");
      int grantmenu = scan.nextInt();
      System.out.println("=============================================");
      String crud_rights; // 입력받게.... select,update,delete,insert,all
      String db_name; // 권한 줄 데이터베이스 이름 .. db이름...전체(*)
      String table_name; // 권한 줄 데이터베이스 이름.. 테이블 이름,...전체(*)
      String account; // 권한 계정 생성
```

```
        String account_pw; // 권한 계정 비밀번호 ...
        switch (grantmenu) {
        case 1: // 권한 부여
          System.out.println(" * 권한 부여 화면입니다 * ");
          try {
              System.out.println(" :: 권한을 여러개 부여받을시 ','로 이어주세요!!");
              System.out.println(" :: 전체 권한을 부여받고 싶으면 'all'을 입력해주시면 됩니다!!");
              System.out.println("=============================================");
              System.out.print(" >> 부여받을 권한을 입력하세요 :");
              crud_rights = scan.next();
              System.out.print(" >> 권한을 넣고싶은 DB명을 입력하세요 :");
              db_name = scan.next();
              System.out.print(" >> 권한을 넣고싶은 테이블명을 입력하세요 :");
              table_name = scan.next();
              System.out.print(" >> 계정명을 입력하세요 :");
              account = scan.next();
              System.out.print(" >> 비밀번호를 입력하세요 :");
              account_pw = scan.next();

            //grant all on bookstore.* to user1@localhost identified by 'user1234';
              String SQL = "grant " + crud_rights + " on " + db_name + "."
                           + table_name + " to " + account + "@localhost identified
                           by '" + account_pw + "';";
              st.executeUpdate(SQL);
          } catch (SQLException e) {
                System.out.println("데이터베이스 권한부여 오류 : " + e.getMessage());
          }
          break;

        case 2: // 권한 확인
          System.out.println(" * 권한 부여 및 회수 후 권한 확인하는 화면입니다. *");
          try {
              System.out.print(" >> 권한을 확인할 계정명을 입력하세요: ");
              String accout = scan.next();
              System.out.println("=====================================");
              System.out.println("☞ "+accout+"계정의 권한을 출력합니다.");
              String sql = "show grants for " + accout + "@localhost;";
              rs = st.executeQuery(sql);
              while (rs.next()) {
                    String db = rs.getString(1);
                    System.out.println(db);
              }
          } catch (SQLException e) {
                System.out.println("데이터베이스 권한확인 오류 : " + e.getMessage());
          }
          break;

        case 3: // 권한 회수
          System.out.println(" * 권한 회수하는 화면입니다. *");
          try {
              System.out.println(" :: 권한을 여러개 회수할시  ','로 이어주세요!!");
              System.out.println(" :: 전체 권한을 회수하고 싶으면 'all'을 입력해주시면 됩니다!!");
              System.out.println("=====================================");
```

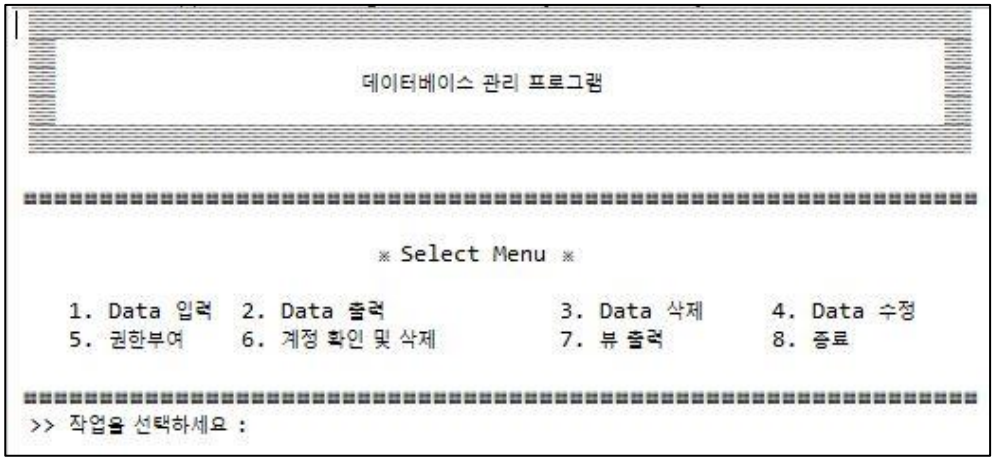
```
            System.out.print(" >> 회수할 권한을 입력하세요 :");
            crud_rights = scan.next();
            System.out.print(" >> 권한을 회수할 DB명을 입력하세요 :");
            db_name = scan.next();
            System.out.print(" >> 권한을 회수할 테이블명을 입력하세요 :");
            table_name = scan.next();
            System.out.print(" >> 권한을 회수할 계정명을 입력하세요 :");
            account = scan.next();

            // revoke all on bookstore.* from user1@localhost;
            String SQL = "revoke " + crud_rights + " on " + db_name + "."
                     + table_name + " from " + account+ "@localhost;";
            st.executeUpdate(SQL);
        } catch (SQLException e) {
            System.out.println("데이터베이스 권한회수 오류 : " + e.getMessage());
        }
        break;
      }
    }

    public void accountcontent() {
        int accountmenu; // 모든 계정확인 및 계정 삭제 메뉴를 고를 수 있는 변수명..
        scan = new Scanner(System.in);
        System.out.println(" * mysql 내에 있는 모든 계정 확인 및 삭제 메뉴입니다 * ");
        System.out.println(" 1. 모든 계정 확인 \t 2. 특정 계정 삭제");
        System.out.print(" >> 메뉴 번호를 선택하세요: ");
        accountmenu = scan.nextInt();
        System.out.println("==========================");

        switch (accountmenu) {
        case 1:
          System.out.println(" ## mysql내에 있는 모든 계정 출력 ##");
           /*
             * 등록된 계정 확인 use mysql; select user from user;
           */
          try {
                ResultSet rs = st.executeQuery("use mysql;");
                rs = st.executeQuery("select user from user;");
                while (rs.next()) {
                      String db = rs.getString(1);
                      System.out.println(db);
                }
          } catch (SQLException e) {
                 e.getStackTrace();
          }
          break;

        case 2:
          System.out.println(" ## mysql내에 있는 특정 계정 삭제 화면 ## ");
          /* drop user 계정명@호스트; */
          try {
                System.out.print(" >> 삭제할 계정명을 입력하세요 :");
                String account = scan.next();
                String SQL = "drop user " + account + "@localhost;";
                st.executeUpdate(SQL);
```

```
            } catch (SQLException e) {
                e.getStackTrace();
            }
            break;
        }
    }

    // 7번 메뉴 뷰를 출력하는 화면!! view_output
    public void view_output() {
        String 고객이름,상품명;
        int 고객아이디, 상품아이디;
        try {
            rs = st.executeQuery("select * from easyorder;");
            if (rs == null) {
              System.out.println("저장된 데이터가 없습니다.");
            } else {
                while (rs.next()) {
                    고객아이디 = rs.getInt("고객아이디");
                    고객이름 = rs.getString("고객이름");
                    상품아이디 = rs.getInt("상품아이디");
                    상품명 = rs.getString("상품명");
                    System.out.printf("%3d \t %5s \t %3d \t %6s",
                            고객아이디, 고객이름, 상품아이디, 상품명);
                    System.out.println();
                }
                System.out.println("");
            }
        } catch (Exception e) {
            System.out.println("데이터베이스 뷰  easyorder 출력 오류 : " + e.getMessage());
        }
    }

    public void view_output2() {
        String 주문정보설명;
        try {
            rs = st.executeQuery("select * from easyorderdetail;");
            if (rs == null) {
                System.out.println("저장된 데이터가 없습니다.");
            } else {
                while (rs.next()) {
                    주문정보설명 = rs.getString("주문정보설명");
                    System.out.printf("%s", 주문정보설명);
                    System.out.println();
                }
                System.out.println("");
            }
        } catch (Exception e) {
            System.out.println("데이터베이스 뷰  easyorder 출력 오류 : " + e.getMessage());
        }
    }

    public void end() {
        try {
            con.close();
```

```
        } catch (SQLException e) {
                System.out.println("데이터베이스 오류 : " + e.getMessage());
        }
    }
}
```

5. 실행 내용

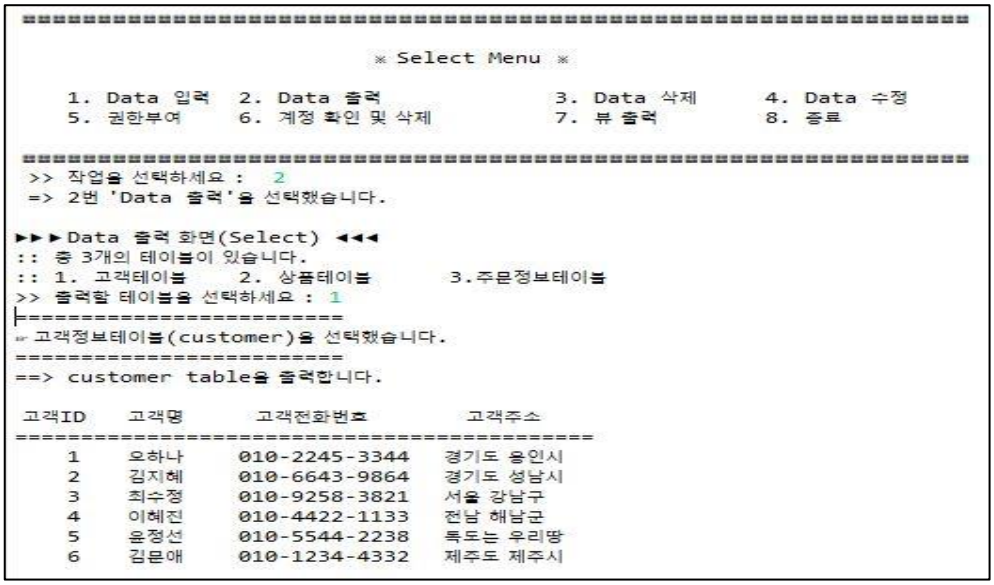
(0) 초기실행화면



➔ 프로그램을 실행하면 해당 초기 화면이 뜹니다.

(1) Data 입력 및 출력 메뉴

➔ '1번 데이터 입력'을 하기 전, '2번 데이터 출력'에서 insert 명령어를 실행시킬 테이블 내용을 확인해봅니다.



➔ 고객 (customer) 테이블에는 총 6개 행이 있는 것으로 확인됩니다. 그러면, '1번 데이터 입력' 메뉴를 통해서 추가 내용을 insert 해보겠습니다.

```
================================================================================

                              ※ Select Menu ※

    1. Data 입력   2. Data 출력              3. Data 삭제      4. Data 수정
    5. 권한부여    6. 계정 확인 및 삭제      7. 뷰 출력        8. 종료

================================================================================
 >> 작업을 선택하세요 :  1
 => 1번 'Data 입력'을 선택했습니다.

▶▶▶Data 입력 화면(Insert) ◀◀◀
:: 총 3개의 테이블이 있습니다.
:: 1. 고객테이블     2. 상품테이블       3.주문정보테이블
>> 입력할 테이블을 선택하세요 : 1
=========================
☞고객정보테이블(customer)을 선택했습니다.
=========================
테이블에 입력 값을 넣어주세요!!!
> 고객 ID 입력하세요 : 10
> 고객 이름 입력하세요 : 김한양
> 고객 전화번호를 입력하세요 : 010-1111-1212
> 고객 주소를 입력하세요 : 서울시 용산구
```

➔ '1번 데이터 입력' 메뉴를 통해 고객 테이블에 '김한양'이라는 고객을 insert했습니다. 그리고 입력한 값이 정상적으로 들어갔는지 다시 '2번 데이터 출력' 메뉴를 통해 확인합니다.

```
▶▶▶Data 출력 화면(Select) ◀◀◀
:: 총 3개의 테이블이 있습니다.
:: 1. 고객테이블     2. 상품테이블       3.주문정보테이블
>> 출력할 테이블을 선택하세요 : 1
=========================
☞고객정보테이블(customer)을 선택했습니다.
=========================
==> customer table을 출력합니다.

 고객ID    고객명       고객전화번호          고객주소
================================================
    1     오하나     010-2245-3344    경기도 용인시
    2     김지혜     010-6643-9864    경기도 성남시
    3     최수정     010-9258-3821    서울 강남구
    4     이혜진     010-4422-1133    전남 해남군
    5     윤정선     010-5544-2238    독도는 우리땅
    6     김문애     010-1234-4332    제주도 제주시
   10     김한양     010-1111-1212    서울시 용산구
```

➔ '2번 데이터 출력' 메뉴를 통해 입력한 값이 정상적으로 insert된 것을 확인했습니다.

(2) Data 삭제 메뉴

➔ 테이블 안에 있는 내용을 삭제하기 위해 '2번 데이터 출력' 메뉴에서 삭제할 데이터를 확인합니다.

```
▶▶▶Data 출력 화면(Select) ◀◀◀
:: 총 3개의 테이블이 있습니다.
:: 1. 고객테이블     2. 상품테이블       3.주문정보테이블
>> 출력할 테이블을 선택하세요 : 2
==========================
상품정보테이블(products)을 선택했습니다.
==========================
==> products table을 출력합니다.

상품ID        상품명          가격
===================================
  1          고데기         60000
  2          텀블러         12000
  3          파우치          4000
  4        핸드폰케이스      13000
  5          충전기          1000
  6          머그컵          7000
444          금지품목           0
```

➔ 상품(products) 테이블에 '상품ID'가 44번인 상품이 금지품목으로 지정되었기 때문에 해당 행을 삭제하도록 하겠습니다.

```
================================================================================

                          ※ Select Menu ※

   1. Data 입력   2. Data 출력              3. Data 삭제      4. Data 수정
   5. 권한부여    6. 계정 확인 및 삭제      7. 뷰 출력        8. 종료

================================================================================
 >> 작업을 선택하세요 :  3
 => 3번 'Data 삭제'를 선택했습니다.

▶▶▶Data 삭제 화면(Delete) ◀◀◀
:: 총 3개의 테이블이 있습니다.
:: 1. 고객테이블     2. 상품테이블       3.주문정보테이블
>> 삭제할 컬럼이 있는 테이블을 선택하세요 : 2
==========================
상품정보테이블(products)을 선택했습니다.
==========================
 @ products table 컬럼삭제 화면입니다 @
> 상품정보의 삭제할 상품ID를 입력하세요: 444
```

➔ 444번 상품을 삭제메뉴를 통해 삭제해봤습니다. 아래 화면 확인결과, 성공적으로 실행되었습니다

```
▶▶▶Data 출력 화면(Select) ◀◀◀
:: 총 3개의 테이블이 있습니다.
:: 1. 고객테이블     2. 상품테이블       3.주문정보테이블
>> 출력할 테이블을 선택하세요 : 2
==========================
상품정보테이블(products)을 선택했습니다.
==========================
==> products table을 출력합니다.

상품ID        상품명          가격
===================================
  1          고데기         60000
  2          텀블러         12000
  3          파우치          4000
  4        핸드폰케이스      13000
  5          충전기          1000
  6          머그컵          7000
```

(3) Data 수정

➔ '4번 데이터 수정' 메뉴를 실행하기 전에, 주문(orders) 테이블에서 수정할 내용을 확인합니다.

```
▶▶▶Data 출력 화면(Select) ◀◀◀
:: 총 3개의 테이블이 있습니다.
:: 1. 고객테이블     2. 상품테이블       3.주문정보테이블
>> 출력할 테이블을 선택하세요 : 3
=========================
☞주문정보테이블(orders)을 선택했습니다.
=========================
==> orders table을 출력합니다.

주문번호    주문개수    주문날짜    주문고객ID    주문상품ID
=================================================
  1        10      2018-03-03      2         5
  2         1      2019-06-05      3         1
  3         3      2019-06-08      2         3
  4         1      2019-06-11      4         5
  5         2      2019-06-20      5         4
```

➔ 주문번호 1의 주문개수가 '10'으로 확인이 됩니다. 이를 '1'로 데이터 수정을 해보겠습니다.

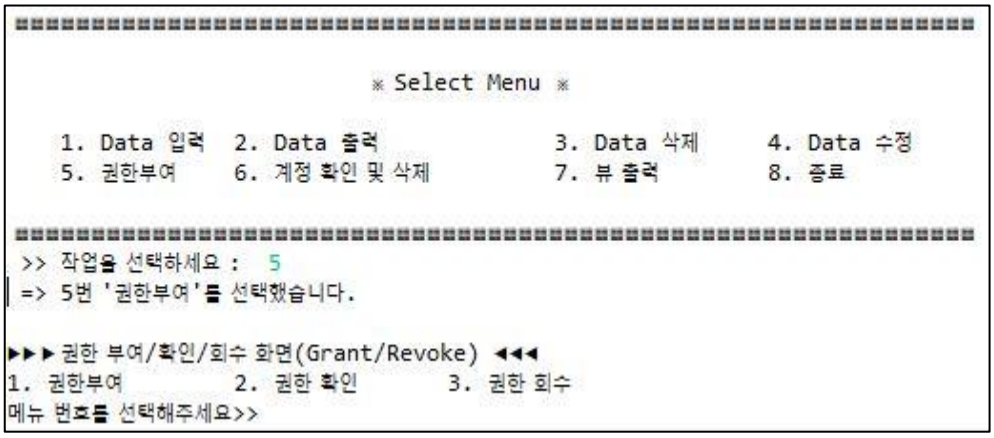
```
================================================================================

                          ※ Select Menu ※

    1. Data 입력   2. Data 출력          3. Data 삭제      4. Data 수정
    5. 권한부여    6. 계정 확인 및 삭제   7. 뷰 출력         8. 종료

================================================================================
 >> 작업을 선택하세요 :  4
=> 4번 'Data 수정'을 선택했습니다.

▶▶▶Data 수정 화면(Update) ◀◀◀
:: 총 3개의 테이블이 있습니다.
:: 1. 고객테이블     2. 상품테이블       3.주문정보테이블
>> 수정할 테이블을 선택하세요 : 3
=========================
☞주문정보테이블(orders)을 선택했습니다.
=========================
@ orders table 수정(update) 화면입니다 @
1.주문수량 2.주문날짜 3.주문고객ID 4.주문상품ID
수정할 필드명을 선택하세요: >>1
+ 수정할 주문수량을 입력하세요:>> 1
+ 수정할 주문번호를 입력하세요:>> 1
```

➔ '4번 데이터 수정' 메뉴는 수정하고 싶은 테이블의 속성명을 선택할 수 있으며 선택 후 수정할 내용을 입력하고 수정할 내용이 있는 기본키인 ID값을 입력하면 사용자가 원하는 값으로 변경됩니다.

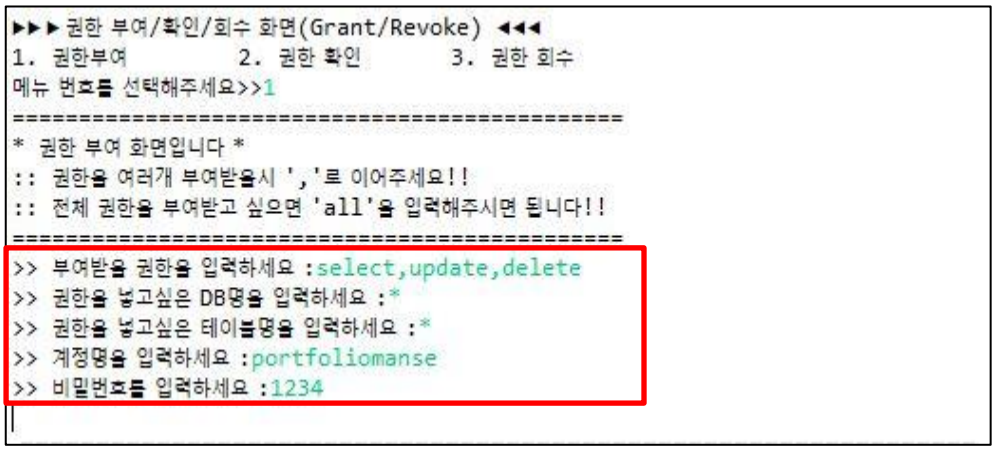
```
=========================
☞주문정보테이블(orders)을 선택했습니다.
=========================
==> orders table을 출력합니다.

주문번호    주문개수    주문날짜    주문고객ID    주문상품ID
=================================================
  1         1      2018-03-03      2         5
  2         1      2019-06-05      3         1
  3         3      2019-06-08      2         3
  4         1      2019-06-11      4         5
  5         2      2019-06-20      5         4
```

(4) 권한 부여

➔ 아래 화면은 '5번 권한 부여' 메뉴를 선택했을 시 초기화면입니다. 5번 메뉴를 통해 권한부여/권한확인/권한회수 3가지 기능을 실행할 수 있습니다.



➔ 5번의 1번 권한부여 메뉴를 통해 권한을 부여합니다.



➔ 5번의 2번 권한확인 메뉴를 통해 권한이 잘 들어갔는지 확인합니다.

➔ 그리고 5번의 3번 권한회수 메뉴를 통해 다시 'portfoliomanse' 계정에서 'DELETE'권한만 회수합니다.

➔ ‘DELETE’ 권한이 정상적으로 회수되었는지 확인합니다.

```
▶▶▶권한 부여/확인/회수 화면(Grant/Revoke) ◀◀◀
1. 권한부여          2. 권한 확인        3. 권한 회수
메뉴 번호를 선택해주세요>>2
==========================================
* 권한 부여 및 회수 후 권한 확인하는 화면입니다. *
>> 권한을 확인할 계정명을 입력하세요: portfoliomanse
======================================
portfoliomanse계정의 권한을 출력합니다.
GRANT SELECT, UPDATE ON *.* TO 'portfoliomanse'@'localhost' IDENTIFIED BY
```

(5) 계정확인 및 삭제

➔ 아래 화면은 ‘6번 계정 확인 및 삭제’ 초기화면입니다. 해당 메뉴를 통해서 전체 계정을 확인할 수 있으며 특정 계정 하나를 골라서 삭제할 수 있습니다.

```
================================================================================

                          ※ Select Menu ※

    1. Data 입력   2. Data 출력          3. Data 삭제      4. Data 수정
    5. 권한부여    6. 계정 확인 및 삭제  7. 뷰 출력        8. 종료

================================================================================
 >> 작업을 선택하세요 :  6
=> 6번 '계정 확인 및 삭제'를 선택했습니다.

▶▶▶계정 확인 및 삭제 화면 ◀◀◀
 * mysql 내에 있는 모든 계정 확인 및 삭제 메뉴입니다 *
1. 모든 계정 확인     2. 특정 계정 삭제
>> 메뉴 번호를 선택하세요:
```

➔ 6번의 ‘1번 모든 계정 확인’을 통해 모든 계정을 확인해보겠습니다.

```
▶▶▶계정 확인 및 삭제 화면 ◀◀◀
 * mysql 내에 있는 모든 계정 확인 및 삭제 메뉴입니다 *
1. 모든 계정 확인     2. 특정 계정 삭제
>> 메뉴 번호를 선택하세요: 1
========================
## mysql내에 있는 모든 계정 출력 ##
root
root

a12345
ee11
moonsun
portfoliomanse
root

--------------------------------------------------------------------------------
```

➔ ‘ee11’ 계정을 6번의 ‘2번 특정 계정 삭제’ 메뉴를 통해서 삭제하겠습니다.

```
▶▶▶계정 확인 및 삭제 화면 ◀◀◀
 * mysql 내에 있는 모든 계정 확인 및 삭제 메뉴입니다 *
1. 모든 계정 확인     2. 특정 계정 삭제
>> 메뉴 번호를 선택하세요: 2
========================
## mysql내에 있는 특정 계정 삭제 화면 ##
>> 삭제할 계정명을 입력하세요 :ee11
```

➔ ‘eell’ 계정이 삭제되었는지 확인합니다.

```
▶▶▶계정 확인 및 삭제 화면◀◀◀
 * mysql 내에 있는 모든 계정 확인 및 삭제 메뉴입니다 *
1. 모든 계정 확인      2. 특정 계정 삭제
>> 메뉴 번호를 선택하세요: 1
╞=========================
## mysql내에 있는 모든 계정 출력 ##
root
root

a12345
moonsun
portfoliomanse
root
```

(6) 뷰 출력

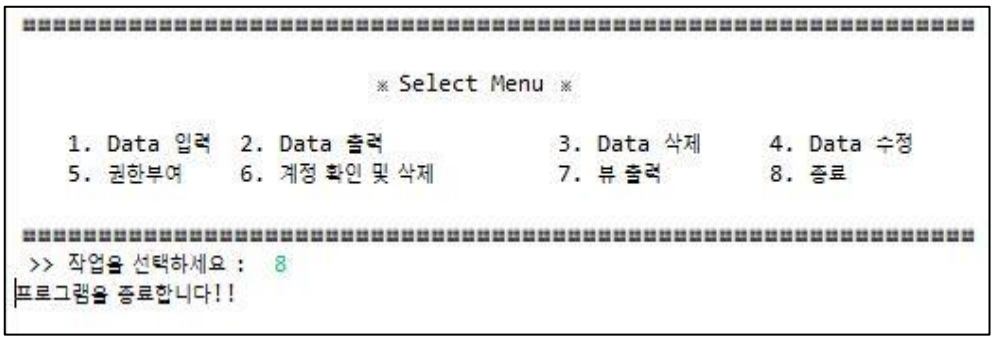
➔ ‘shopping’ 데이터베이스 내에 있는 주문관련 테이블들을 손쉽게 보고자 조인해 만들어놓은 뷰가 있습니다. ‘6번 뷰 출력’ 메뉴를 통해 조회할 수 있습니다.

```
================================================================================

                          ※ Select Menu ※

    1. Data 입력   2. Data 출력          3. Data 삭제      4. Data 수정
    5. 권한부여    6. 계정 확인 및 삭제   7. 뷰 출력         8. 종료

================================================================================
 >> 작업을 선택하세요 :  7
 => 7번 '뷰 출력'을 선택했습니다.

▶▶▶뷰 출력 화면◀◀◀
:: 총 2개의 뷰가 있습니다.
:: 1. 주문간단정보 뷰(easyorder)      2. 주문상세정보 뷰(easyorderdetail)
>> 출력할 뷰를 선택하세요 : 1
╞=========================
☞주문간단정보 뷰(easyorder)를 선택했습니다.
# easyorder view를 출력합니다.

 고객ID     고객이름   상품ID     상품명
==================================
   2         김지혜      5        충전기
   2         김지혜      3        파우치
   3         최수정      1        고데기
   4         이혜진      5        충전기
   5         윤정선      4      핸드폰케이스
```

➔ 위 화면은 1번 뷰를 출력한 부분입니다. 아래 화면은 2번 뷰를 출력한 화면입니다.

```
▶▶▶뷰 출력 화면◀◀◀
:: 총 2개의 뷰가 있습니다.
:: 1. 주문간단정보 뷰(easyorder)      2. 주문상세정보 뷰(easyorderdetail)
>> 출력할 뷰를 선택하세요 : 2
╞=========================
☞주문상세정보 뷰(easyorderdetail)를 선택했습니다.
# easyorderdetail view를 출력합니다

                                주문정보설명
================================================================================
김지혜님께서 충전기 상품을 1개 주문으로 2018-03-03일자의 총 주문금액은 1000원 입니다.
김지혜님께서 파우치 상품을 3개 주문으로 2019-06-08일자의 총 주문금액은 12000원 입니다.
최수정님께서 고데기 상품을 1개 주문으로 2019-06-05일자의 총 주문금액은 60000원 입니다.
이혜진님께서 충전기 상품을 1개 주문으로 2019-06-11일자의 총 주문금액은 1000원 입니다.
윤정선님께서 핸드폰케이스 상품을 2개 주문으로 2019-06-20일자의 총 주문금액은 26000원 입니다.
```

(7) 종료



➔ '8번 종료' 메뉴를 선택하면 해당 프로그램이 종료됩니다.

6. 포트폴리오 후기

MySQL 처럼 데이터베이스, 테이블 생성과 그것을 사용하는 기능을 추가하고 싶어서 추가를 해놨는데, 포트폴리오 페이지(워드파일과 pp파일)가 소스코드와 실행내용이 기하급수적으로 생성되는 바람에 감당할 수 없는 페이지를 넘겼고 이건 안되겠다 싶어 선생님께서 포트폴리오 요구하시는 것만 뽑아서 포트폴리오를 작성하게 되었습니다.

이클립스로 MySQL 연동하는 부분은 안해봐서 (visual studio에서 해봤었는데, 이클립스는 초보여서..) 선생님께서 올려주신 내용들을 많이많이 참고했으며 도움이 많이 되었습니다! 그리고 추가적인 기능 (계정확인 및 삭제) 한 개를 추가하고 프로그램 작성을 마무리했습니다.

이클립스로 연동하는 부분은 처음인지라 선생님께서 올려주신 내용을 공부해가면서 포트폴리오를 작성해나가니 공부가 많이많이 되었습니다. 도중에 sql 오류나 실행오류가 발생하면 인터넷을 찾아봐서 해결해나갔습니다. 포트폴리오하면서 모르는 부분을 캐치할 수 있는 것(?) , 실력이 향상되는 것이 보여 뿌듯합니다.

지금 포트폴리오 코드가 복잡하지만 다음 유지보수 때는 중복될 수 있는 코드들이 있으면 간단한 프로그램이 될 수 있도록 줄일 것이며 MySQL dbms 프로그램처럼 저만의 dbms 프로그램을 만들 수 있도록 할 것입니다. 포트폴리오 끝난 시점에도 SQL 언어를 사용하기 때문에 해당 프로그램은 계속 발전해나가도록 할 예정입니다. 프로그램 활용성이 뛰어나게 작성할 것입니다.