과제 2

32200327 김경민

1.

Customer 테이블에 학생 본인의 정보를 입력하는 프로그램을 작성하고 그 결과를 보이세요.

- CustomerNumber 가 자동 생성될 수 있도록 워크밴치에서 customers 테이블의 cuttomerNumber 속성 AI 체크

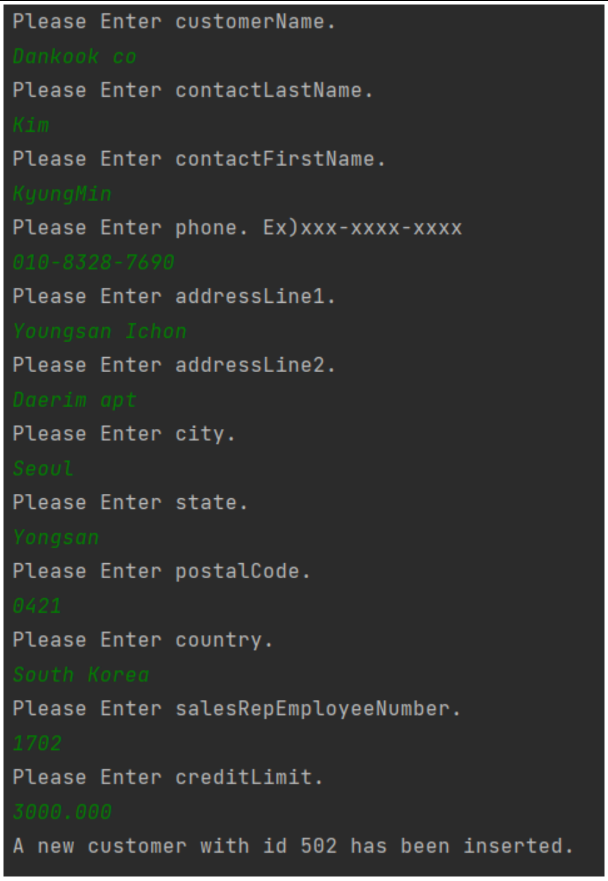
코드

```
package sklee.jdbc.sec0;


import java.sql.*;

import java.util.Scanner;


public class InsertCustomer {

    /**

     * Insert a new candidate

     * @param customerName

     * @param contactLastName

     * @param contactFirstName

     * @param phone

     * @param addressLine1

     * @param addressLine2

     * @param city

     * @param state

     * @param postalCode

     * @param country

     * @param salesRepEmployeeNumber

     * @param creditLimit

     * @return
     */
    public static int insertCustomer(String customerName, String contactLastName, String
contactFirstName, String phone, String addressLine1, String addressLine2,
                                             String city, String state, String postalCode, String
country, int salesRepEmployeeNumber, Double creditLimit){

        // for insert a new customer
        ResultSet rs = null;
        int customerNumber = 0;

        String sql = "INSERT INTO
customers(customerName,contactLastName,contactFirstName,phone,addressLine1,addressLi
ne2,city,state,postalCode, country, salesRepEmployeeNumber,creditLimit) "
                + "VALUES(?,?,?,?,?,?,?,?,?,?,?,?) "; //수행할 쿼리

        try (Connection conn = JDBC_Util.getConnection();
             PreparedStatement pstmt = conn.prepareStatement(sql,
Statement.RETURN_GENERATED_KEYS);) { //statement 객체 생성해 다양한 메소드 사용 가능
(키 자동 생성)

            // set parameters for statement
            pstmt.setString(1, customerName);
            pstmt.setString(2, contactLastName);
            pstmt.setString(3, contactFirstName);
            pstmt.setString(4, phone);
            pstmt.setString(5, addressLine1);
            pstmt.setString(6, addressLine2);
            pstmt.setString(7, city);
            pstmt.setString(8, state);
            pstmt.setString(9, postalCode);
            pstmt.setString(10, country);
            pstmt.setInt(11, salesRepEmployeeNumber);
            pstmt.setDouble(12, creditLimit);

            int rowAffected = pstmt.executeUpdate();
            if(rowAffected == 1) //영향을 받은 행의 갯수가 1개인 경우
            {
                // get candidate id
                rs = pstmt.getGeneratedKeys(); //sql문을 실행한 결과를 반환(key가 생성 되
고 이를 반환 하는 메소드 실행)
                if(rs.next()) {
                    customerNumber = rs.getInt(1); //insert된 행의 첫번째 인덱스의 값 가져
오기
                }

            }
        } catch (SQLException ex) {
            System.out.println(ex.getMessage());
        }finally {
            try {
                if(rs != null)   rs.close();
            } catch (SQLException e) {
                System.out.println(e.getMessage());
            }
        }

        return customerNumber;

    }
    /**
     * @param args the command line arguments
     */
    public static void main(String[] args){
        //enter customer info
        Scanner scanner = new Scanner(System.in);

        String customerName = null;
        String contactLastName = null;
        String contactFirstName = null;
        String phone = null;
        String addressLine1 = null;
        String addressLine2 = null;
        String city = null;
        String state = null;
        String postalCode = null;
        String country = null;
        int salesRepEmployeeNumber = 0;
        Double creditLimit = 0.0;

        System.out.println("Please Enter customerName.");
        customerName = scanner.nextLine();

        System.out.println("Please Enter contactLastName.");
        contactLastName = scanner.nextLine();

        System.out.println("Please Enter contactFirstName.");
        contactFirstName = scanner.nextLine();

        System.out.println("Please Enter phone. Ex)xxx-xxxx-xxxx");
        phone = scanner.nextLine();

        System.out.println("Please Enter addressLine1.");
        addressLine1 = scanner.nextLine();

        System.out.println("Please Enter addressLine2.");
        addressLine2 = scanner.nextLine();
```

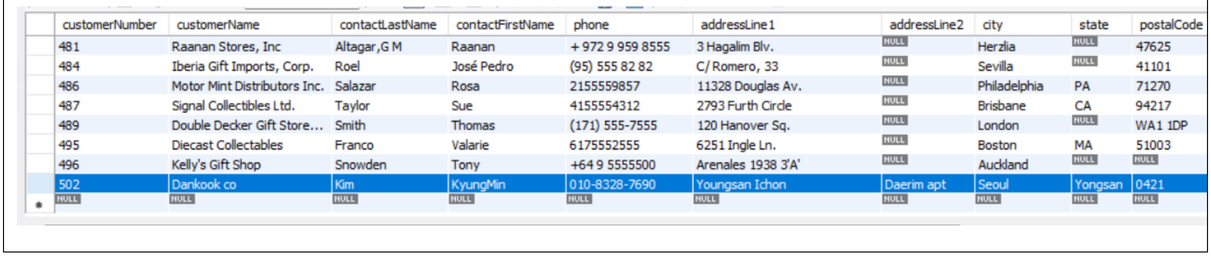
}

```
        System.out.println("Please Enter city.");
        city = scanner.nextLine();

        System.out.println("Please Enter state.");
        state = scanner.nextLine();

        System.out.println("Please Enter postalCode.");
        postalCode = scanner.nextLine();

        System.out.println("Please Enter country.");
        country = scanner.nextLine();

        System.out.println("Please Enter salesRepEmployeeNumber.");
        salesRepEmployeeNumber = scanner.nextInt();

        System.out.println("Please Enter creditLimit.");
        creditLimit = scanner.nextDouble();



        // insert a new customer
        int customerNumber = insertCustomer(customerName, contactLastName,
contactFirstName, phone,
                addressLine1, addressLine2, city, state, postalCode, country,
salesRepEmployeeNumber, creditLimit);


        System.out.println(String.format("A new customer with id %d has been inserted.",
customerNumber));
    }
}
```

실행 결과 (캡쳐)



```
Please Enter customerName.
Dankook co
Please Enter contactLastName.
Kim
Please Enter contactFirstName.
KyungMin
Please Enter phone. Ex)xxx-xxxx-xxxx
010-8328-7690
Please Enter addressLine1.
Youngsan Ichon
Please Enter addressLine2.
Daerim apt
Please Enter city.
Seoul
Please Enter state.
Yongsan
Please Enter postalCode.
0421
Please Enter country.
South Korea
Please Enter salesRepEmployeeNumber.
1702
Please Enter creditLimit.
3000.000
A new customer with id 502 has been inserted.
```



	customerNumber	customerName	contactLastName	contactFirstName	phone	addressLine1	addressLine2	city	state	postalCode
	481	Raanan Stores, Inc	Altagar,G M	Raanan	+ 972 9 959 8555	3 Hagalim Blv.	NULL	Herzlia	NULL	47625
	484	Iberia Gift Imports, Corp.	Roel	José Pedro	(95) 555 82 82	C/ Romero, 33	NULL	Sevilla	NULL	41101
	486	Motor Mint Distributors Inc.	Salazar	Rosa	2155559857	11328 Douglas Av.	NULL	Philadelphia	PA	71270
	487	Signal Collectibles Ltd.	Taylor	Sue	4155554312	2793 Furth Circle	NULL	Brisbane	CA	94217
	489	Double Decker Gift Store...	Smith	Thomas	(171) 555-7555	120 Hanover Sq.	NULL	London	NULL	WA1 1DP
	495	Diecast Collectables	Franco	Valarie	6175552555	6251 Ingle Ln.	NULL	Boston	MA	51003
	496	Kelly's Gift Shop	Snowden	Tony	+64 9 5555500	Arenales 1938 3'A'	NULL	Auckland	NULL	NULL
	502	Dankook co	Kim	KyungMin	010-8328-7690	Youngsan Ichon	Daerim apt	Seoul	Yongsan	0421
*	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

2.

본인이 원하는 제품 명을 입력하면 제품 관련 정보를 찾아 orders 와 orderdetails 에 입력하는 프로그램을 작성하고 그 결과를 보이세요.

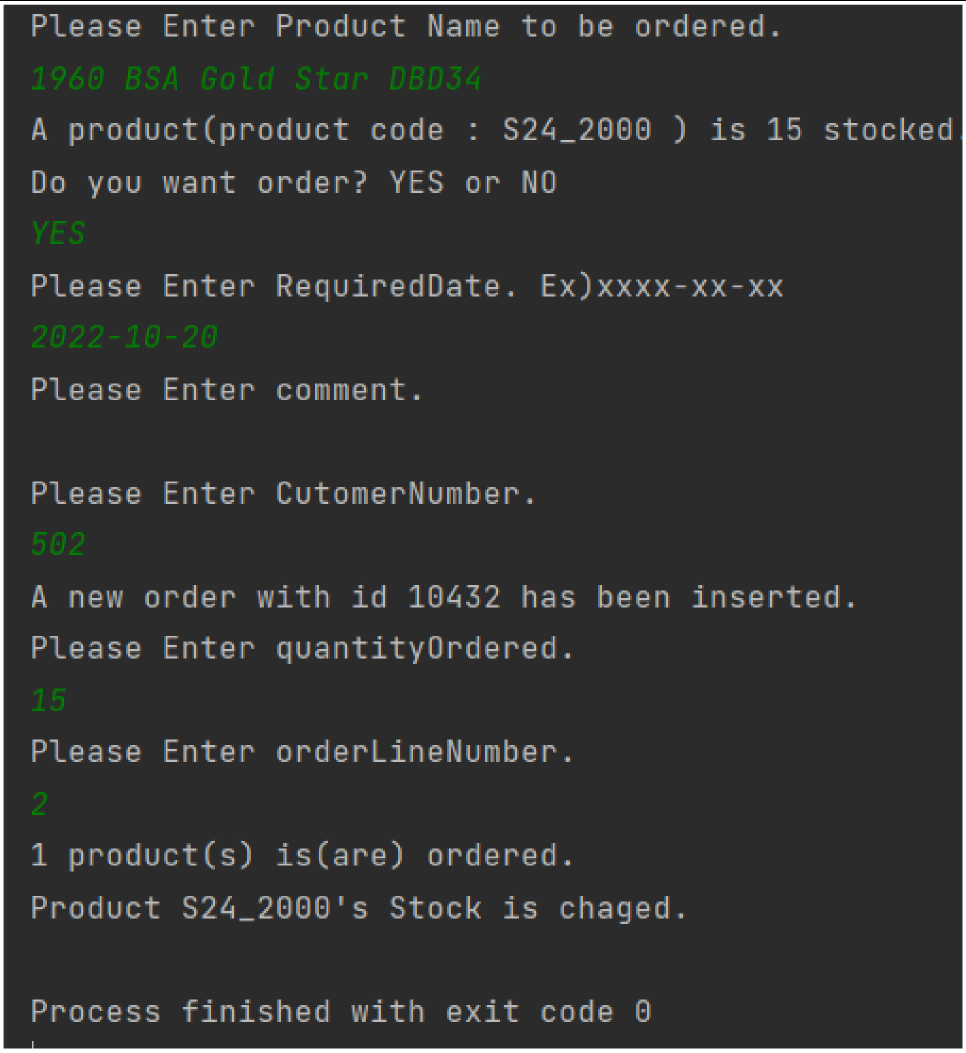
- 제품이름(productName) 입력시, 제품코드(productCode)와 권장소비자가격(MSRP) 추출 * orderdetails 테이블의 priceEach 가 상품마다 다르기 때문에 정해진 가격이 딱히 없다고 판단하고, products 테이블의 buyPrice 도 판매 가격이 아닌 도매 가격이라고 판단. 따라서 priceEach 는 권장소비자가격을 뜻하는 MSPR 로 정함
- 상품코드로 orders 테이블에 주문 생성, orderNumber 은 1 번 문제와 동일하게 자동생성 되도록 설정 바꿔줌
- 주문 번호로 orderdetails 테이블에 주문상세 생성
- 주문 생성 완료 후 products 테이블의 재고(quanttyInStock) update

코드

```
package sklee.jdbc.sec0;


import java.sql.*;
import java.text.DateFormat;
import java.text.SimpleDateFormat;
import java.util.Calendar;
import java.util.Scanner;
import java.util.Date;
import static java.lang.System.exit;

public class OrderProgram {
    String productCode = null;
    double msrp = 0;
    int quantityInStock = 0;

    public void searchProduct(String product){

        //search product info
        String sql = "SELECT productCode, quantityInStock, MSRP FROM products WHERE productName = ? "; //수행할 쿼리

        try (Connection conn = JDBC_Util.getConnection(); //DB 연결 객체 생성
             PreparedStatement pstmt  = conn.prepareStatement(sql);) {   //SQL문을 실행하고 ResultSet 형식의 결과 리턴

            pstmt.setString(1, product);
            ResultSet rs = pstmt.executeQuery();

            while (rs.next()) {
                productCode = rs.getString("productCode");
                quantityInStock = rs.getInt("quantityInStock");
                msrp = rs.getDouble("MSRP");

                if(productCode == null){
                    System.out.println("Product Name is invalid.");
                }

                if(quantityInStock == 0){
                    System.out.println("Sorry, We don't have stock.");
                    exit(0);
                }
            }
        } catch (SQLException ex) {
            System.out.println(ex.getMessage());
        }

        System.out.println(String.format("A product(product code : %s ) is %d
stocked.",productCode, quantityInStock));
    }

    public int createOrder(String requiredDate,String comments,int customerNumber){

        ResultSet rs = null;
        int orderNumber = 0;

        String sql = "INSERT INTO orders(orderDate, requiredDate, shippedDate, status,
comments, customerNumber)"
                + "VALUES(?,?,?,?,?,?) "; //수행할 쿼리

        try (Connection conn = JDBC_Util.getConnection(); //DB 연결 객체 생성
             PreparedStatement pstmt = conn.prepareStatement(sql,
Statement.RETURN_GENERATED_KEYS);) {  //SQL문을 실행하고 ResultSet 형식의 결
과 리턴
```

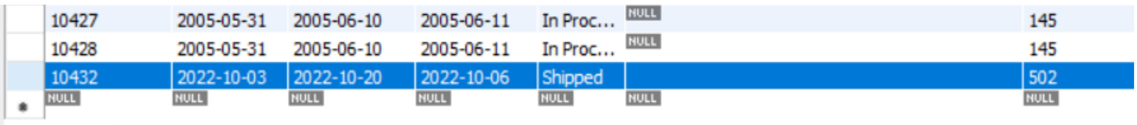
```
        DateFormat df = new SimpleDateFormat("yyyy-MM-dd");
        Calendar cal = Calendar.getInstance();
        cal.setTime(new Date());
        String orderDate = df.format(cal.getTime());
        //set shippedDate(orderDate + 3)
        cal.add(Calendar.DATE, 3);
        String shippedDate = df.format(cal.getTime());

        pstmt.setString(1, orderDate);
        pstmt.setString(2, requiredDate);
        pstmt.setString(3, shippedDate);
        pstmt.setString(4, "Shipped");
        pstmt.setString(5, comments);
        pstmt.setInt(6, customerNumber);

        int rowAffected = pstmt.executeUpdate();
        if(rowAffected == 1) //영향을 받은 행의 갯수가 1개인 경우
        {
            // get candidate id
            rs = pstmt.getGeneratedKeys();
            if(rs.next()) {
                orderNumber = rs.getInt(1);
            }
        }
    } catch (SQLException ex) {
        System.out.println(ex.getMessage());
    }finally {
        try {
            if(rs != null)   rs.close();
        } catch (SQLException e) {
            System.out.println(e.getMessage());
        }
    }
    return orderNumber;
}

public void createOrderDetail(int orderNumber, String productCode, int quantityOrdered, double priceEach, int orderLineNumber){

    ResultSet rs = null;

    String sql = "INSERT INTO orderdetails(orderNumber, productCode, quantityOrdered, priceEach, orderLineNumber)"
            + "VALUES(?,?,?,?,?) "; //수행할 쿼리

    try (Connection conn = JDBC_Util.getConnection(); //DB 연결 객체 생성
         PreparedStatement pstmt = conn.prepareStatement(sql);) {  //SQL문을 실행하고 ResultSet 형식의 결과 리턴

        pstmt.setInt(1, orderNumber);
        pstmt.setString(2, productCode);
        pstmt.setInt(3, quantityOrdered);
        pstmt.setDouble(4, priceEach);
        pstmt.setInt(5, orderLineNumber);

        int rowAffected = pstmt.executeUpdate();
        System.out.println(String.format("%d product(s) is(are) ordered.", rowAffected));

    } catch (SQLException ex) {
        System.out.println(ex.getMessage());
    }finally {
        try {
            if(rs != null)   rs.close();
        } catch (SQLException e) {
            System.out.println(e.getMessage());
        }
    }
}

public void updateStock(int quantityOrdered, String productCode){
    String sql = "UPDATE products "
            + "SET quantityInStock = ? "
            + "WHERE productCode = ? ";

    try (Connection conn = JDBC_Util.getConnection();
         PreparedStatement pstmt = conn.prepareStatement(sql)){

        pstmt.setInt(1, quantityInStock - quantityOrdered);
        pstmt.setString(2, productCode);

        int rowAffected = pstmt.executeUpdate();
        System.out.println(String.format("Product %s's Stock is
chaged.",productCode));

    } catch (SQLException ex) {
        System.out.println(ex.getMessage());
    }
}

public static void main(String[] args){

    OrderProgram newOrder = new OrderProgram();

    Scanner scanner = new Scanner(System.in);
    //Enter Product to be ordered
    System.out.println("Please Enter Product Name to be ordered.");
    String productName = scanner.nextLine();
    //여러개인 경우



    //search product code
    newOrder.searchProduct(productName);

    System.out.println("Do you want order? YES or NO");
    if(scanner.nextLine().equals("YES")){

        //Enter RequiredDate
        System.out.println("Please Enter RequiredDate. Ex)xxxx-xx-xx");
        String requiredDate = scanner.nextLine();

        //Enter comment
        System.out.println("Please Enter comment.");
        String comment = scanner.nextLine();

        //Enter CutomerNumber
        System.out.println("Please Enter CutomerNumber.");
        int cutomerNumber = scanner.nextInt();



        //create order
        int orderNumber = newOrder.createOrder(requiredDate, comment ,cutomerNumber);
        System.out.println(String.format("A new order with id %d has been inserted.", orderNumber));
        //Enter quantityOrdered
        System.out.println("Please Enter quantityOrdered.");
        int quantityOrdered = scanner.nextInt();

        //Enter orderLineNumber
        System.out.println("Please Enter orderLineNumber.");
        int orderLineNumber = scanner.nextInt();

        //create orderdetail
        newOrder.createOrderDetail(orderNumber, newOrder.productCode, quantityOrdered, newOrder.msrp,orderLineNumber);

        //update product stock
        newOrder.updateStock(quantityOrdered, newOrder.productCode);



    } else {
        exit(0);
    }
}
}
```

실행 결과 (캡쳐)



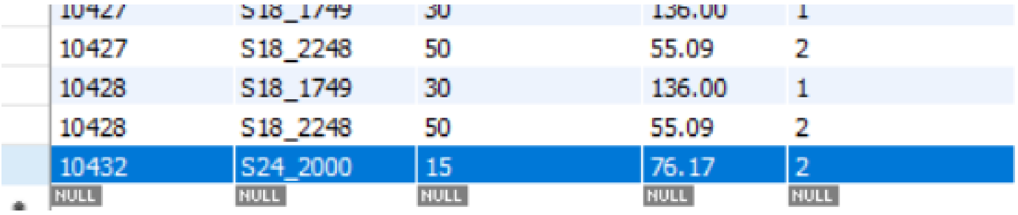
```
Please Enter Product Name to be ordered.
1960 BSA Gold Star DBD34
A product(product code : S24_2000 ) is 15 stocked.
Do you want order? YES or NO
YES
Please Enter RequiredDate. Ex)xxxx-xx-xx
2022-10-20
Please Enter comment.

Please Enter CutomerNumber.
502
A new order with id 10432 has been inserted.
Please Enter quantityOrdered.
15
Please Enter orderLineNumber.
2
1 product(s) is(are) ordered.
Product S24_2000's Stock is chaged.

Process finished with exit code 0
```

orders 테이블 결과 화면



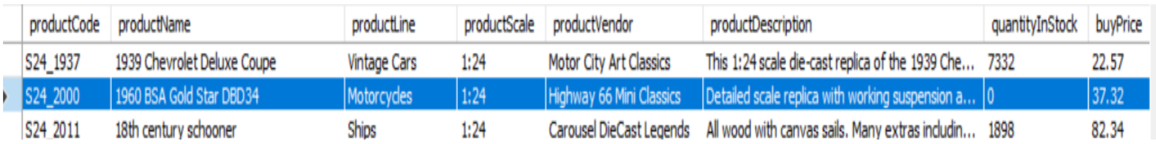
10427	2005-05-31	2005-06-10	2005-06-11	In Proc...	NULL	145
10428	2005-05-31	2005-06-10	2005-06-11	In Proc...	NULL	145
10432	2022-10-03	2022-10-20	2022-10-06	Shipped		502
NULL	NULL	NULL	NULL	NULL	NULL	NULL

orderdetails 테이블 결과 화면



10427	S18_1749	30	136.00	1
10427	S18_2248	50	55.09	2
10428	S18_1749	30	136.00	1
10428	S18_2248	50	55.09	2
10432	S24_2000	15	76.17	2
NULL	NULL	NULL	NULL	NULL

products 테이블 결과 화면(재고 update)



productCode	productName	productLine	productScale	productVendor	productDescription	quantityInStock	buyPrice
S24_1937	1939 Chevrolet Deluxe Coupe	Vintage Cars	1:24	Motor City Art Classics	This 1:24 scale die-cast replica of the 1939 Che...	7332	22.57
S24_2000	1960 BSA Gold Star DBD34	Motorcycles	1:24	Highway 66 Mini Classics	Detailed scale replica with working suspension a...	0	37.32
S24 2011	18th century schooner	Ships	1:24	Carousel DieCast Legends	All wood with canvas sails. Many extras includin...	1898	82.34