Part 1.1

```
CREATE VIEW OrderDetail AS
SELECT o.OrderId, o.OrderDate,
       c.CustomerId, c.CustomerName,
       p.ProductId, p.ProductDescription, p.ProductStandardPrice,
       ol.OrderedQuantity, SUM(p.ProductStandardPrice * ol.OrderedQuantity) AS OrderPrice
FROM Order_T o
JOIN Customer_T c ON o.CustomerId = c.CustomerId
JOIN OrderLine_T ol ON o.OrderId = ol.OrderId
JOIN Product_T p ON ol.ProductId = p.ProductId
GROUP BY o.OrderId, o.OrderDate, c.CustomerId, c.CustomerName, p.ProductDescription,
p.ProductId, p.ProductStandardPrice, ol.OrderedQuantity;

SELECT * FROM ORDERDETAIL;
```

Script Output | Query Result

SQL | All Rows Fetched: 18 in 0 seconds

	ORDERID	ORDERDATE	CUSTOMERID	CUSTOMERNAME	PRODUCTID	PRODUCTDESCRIPTION	PRODUCTSTANDARDPRICE	ORDEREDQUANTITY	ORDERPRICE
1	1008	30-OCT-10	12	Battle Creek Furniture	3	Computer Desk	375	3	1125
2	1006	24-OCT-10	2	Value Furniture	5	Writers Desk	325	2	650
3	1004	22-OCT-10	5	Impressions	6	8-Drawer Desk	750	2	1500
4	1008	30-OCT-10	12	Battle Creek Furniture	8	Computer Desk	250	3	750
5	1004	22-OCT-10	5	Impressions	8	Computer Desk	250	2	500
6	1001	21-OCT-10	1	Contemporary Casuals	1	End Table	175	2	350
7	1001	21-OCT-10	1	Contemporary Casuals	2	Coffee Table	200	2	400
8	1003	22-OCT-10	15	Mountain Scenes	3	Computer Desk	375	3	1125
9	1005	24-OCT-10	3	Home Furnishings	4	Entertainment Center	650	3	1950
10	1009	05-NOV-10	4	Eastern Furniture	4	Entertainment Center	650	2	1300
11	1009	05-NOV-10	4	Eastern Furniture	7	Dining Table	800	3	2400
12	1010	05-NOV-10	1	Contemporary Casuals	8	Computer Desk	250	10	2500
13	1006	24-OCT-10	2	Value Furniture	4	Entertainment Center	650	1	650
14	1006	24-OCT-10	2	Value Furniture	7	Dining Table	800	2	1600
15	1007	27-OCT-10	11	American Euro Lifestyles	1	End Table	175	3	525
16	1001	21-OCT-10	1	Contemporary Casuals	4	Entertainment Center	650	1	650
17	1007	27-OCT-10	11	American Euro Lifestyles	2	Coffee Table	200	2	400
18	1002	21-OCT-10	8	California Classics	3	Computer Desk	375	5	1875

Part 1.2

```
SELECT OrderId, COUNT(OrderId) AS "Number of Order Items", SUM(OrderPrice) AS "Total Order Price"
```

Script Output | Query Result

SQL | All Rows Fetched: 18 in 0.01 seconds

	ORDERID	ORDERDATE	CUSTOMERID	CUSTOMERNAME	PRODUCTID	PRODUCTDESCRIPTION	PRODUCTSTANDARDPRICE	ORDEREDQUANTITY	ORDERPRICE
1	1010	05-NOV-10	1	Contemporary Casuals	8	Computer Desk	250	10	2500
2	1009	05-NOV-10	4	Eastern Furniture	7	Dining Table	800	3	2400
3	1009	05-NOV-10	4	Eastern Furniture	4	Entertainment Center	650	2	1300
4	1008	30-OCT-10	12	Battle Creek Furniture	3	Computer Desk	375	3	1125
5	1008	30-OCT-10	12	Battle Creek Furniture	8	Computer Desk	250	3	750
6	1007	27-OCT-10	11	American Euro Lifestyles	2	Coffee Table	200	2	400
7	1007	27-OCT-10	11	American Euro Lifestyles	1	End Table	175	3	525
8	1006	24-OCT-10	2	Value Furniture	7	Dining Table	800	2	1600
9	1006	24-OCT-10	2	Value Furniture	4	Entertainment Center	650	1	650
10	1006	24-OCT-10	2	Value Furniture	5	Writers Desk	325	2	650
11	1005	24-OCT-10	3	Home Furnishings	4	Entertainment Center	650	3	1950
12	1004	22-OCT-10	5	Impressions	6	8-Drawer Desk	750	2	1500
13	1004	22-OCT-10	5	Impressions	8	Computer Desk	250	2	500
14	1003	22-OCT-10	15	Mountain Scenes	3	Computer Desk	375	3	1125
15	1001	21-OCT-10	1	Contemporary Casuals	2	Coffee Table	200	2	400
16	1001	21-OCT-10	1	Contemporary Casuals	1	End Table	175	2	350
17	1001	21-OCT-10	1	Contemporary Casuals	4	Entertainment Center	650	1	650
18	1002	21-OCT-10	8	California Classics	3	Computer Desk	375	5	1875

Part 1.3

```
SELECT OrderId, COUNT(OrderId) AS "Number of Order Items", SUM(OrderPrice) AS "Total Order Price"
FROM OrderDetail GROUP BY OrderId;
```

Script Output | Query Result

SQL | All Rows Fetched: 10 in 0 seconds

	ORDERID	Number of Order Items	Total Order Price
1	1003	1	1125
2	1009	2	3700
3	1006	3	2900
4	1001	3	1400
5	1002	1	1875
6	1007	2	925
7	1004	2	2000
8	1010	1	2500
9	1008	2	1875
10	1005	1	1950

Part 1.4

```
DROP VIEW OrderDetail;
```

Script Output | Query Result

Task completed in 0.031 secc

```
view ORDERDETAIL dropped.
```

Part 2.1

```
CREATE SEQUENCE EmployeeIdSeq
START WITH 100
MAXVALUE 999
CYCLE
INCREMENT BY 1;
```

Script Output

Task completed in 0.005 seconds

```
sequence EMPLOYEEIDSEQ created.
```

Part 2.2

```
57  Alter table  EMPLOYEE_T modify employeeId varchar2(15);
58
59  INSERT INTO Employee_T  (EmployeeID, EmployeeName, EmployeeAddress, EmployeeCity,
60  EmployeeState, EmployeeZipCode, EmployeeDateHired, EmployeeBirthDate, EmployeeSupervisor)
61  VALUES  (to_char(SYSDATE,'YYYY') || '-' || to_char(SYSDATE,'MM') || '-' ||
62  EmployeeIdSeq.nextval, 'Brandon Young', '', 'Fort Wayne', 'IN', '46805', SYSDATE, '', '');
63
64  INSERT INTO Employee_T  (EmployeeID, EmployeeName, EmployeeAddress, EmployeeCity,
65  EmployeeState, EmployeeZipCode, EmployeeDateHired, EmployeeBirthDate, EmployeeSupervisor)
66  VALUES  (to_char(SYSDATE,'YYYY') || '-' || to_char(SYSDATE,'MM') || '-' ||
67  EmployeeIdSeq.nextval, 'Bailey Whitehill', '', 'Fort Wayne', 'IN', '46805', SYSDATE, '', '');
```

Script Output | Query Result | Query Result 1 | Query Result 2

Task completed in 0.124 seconds

```
table EMPLOYEE_T altered.
1 rows inserted.
1 rows inserted.
```

Part 2.3

```
CREATE VIEW EmployeeIN  AS
SELECT EmployeeId, EmployeeName, EmployeeCity, EmployeeState
FROM Employee_T
WHERE EmployeeState = 'IN'
WITH CHECK OPTION;
```

Script Output | Query Result

Task completed in 0.025 seconds

```
view EMPLOYEEIN created.
```

Part 2.4

```
80  /**Part 2.4
81  Insert the following employee record into EMPLOYEE_T table via EmployeeIN view.**/
82  INSERT INTO EmployeeIN  (EmployeeID, EmployeeName, EmployeeCity, EmployeeState)
83  VALUES  (to_char(SYSDATE,'YYYY') || '-' || to_char(SYSDATE,'MM') || '-'
84  || EmployeeIdSeq.nextval, 'Sebastain Andaluz', 'Fort Wayne', 'IN');
```

Script Output | Query Result | Query Result 1 | Query Result 2

Task completed in 0.002 seconds

```
1 rows inserted.
```

Part 2.5

```
UPDATE EmployeeIN SET EmployeeState = 'CA' WHERE EmployeeName = 'Brandon Young';
```

Script Output | Query Result

Task completed in 0.037 seconds

```
Error starting at line : 46 in command -
UPDATE EmployeeIN SET EmployeeState = 'CA' WHERE EmployeeName = 'Brandon Young'
Error report -
SQL Error: ORA-01402: view WITH CHECK OPTION where-clause violation
01402. 00000 -  "view WITH CHECK OPTION where-clause violation"
*Cause:
*Action:
```

Because of the WITH CHECK OPTION the Oracle Database prohibits any changes to EmployeeIN that would produce rows that are not included in the subquery.

Part 3.1

```
50 SELECT c.CustomerId, c.CustomerName, c.CustomerPostalCode, COUNT(o.OrderId) AS "Order Frequency"
51 FROM Customer_T c, Order_T o
52 WHERE o.CustomerId = c.CustomerId
53 GROUP BY c.CustomerId, c.CustomerName, c.CustomerPostalCode
54 ORDER BY COUNT(o.OrderId) DESC;
55
```

Script Output | Query Result | Query Result 1 | Query Result 2

SQL | All Rows Fetched: 9 in 0.008 seconds

	CUSTOMERID	CUSTOMERNAME	CUSTOMERPOSTALCODE	Order Frequency
1	1	Contemporary Casuals	32601-2871	3
2	15	Mountain Scenes	84403-4432	2
3	12	Battle Creek Furniture	49015-3401	1
4	4	Eastern Furniture	07008-3188	1
5	2	Value Furniture	75094-7743	1
6	5	Impressions	94206-4056	1
7	11	American Euro Lifestyles	07508-5621	1
8	8	California Classics	96915-7754	1
9	3	Home Furnishings	12209-1125	1

Part 3.2

```
56  DROP TABLE TopKCustomer_T;
57  CREATE TABLE TopKCustomer_T
58  AS
59  SELECT CUSTOMERID, CUSTOMERNAME, CUSTOMERPOSTALCODE
60  FROM CUSTOMER_T
61  WHERE 1=2;
62  ALTER TABLE TopKCustomer_T
63  ADD (Crank NUMBER, OrderFrequency Number, RankGenerateDate Date);
64
```

Script Output x | Query Result x

Task completed in 0.267 seconds

```
table TOPKCUSTOMER_T dropped.
table TOPKCUSTOMER_T created.
table TOPKCUSTOMER_T altered.
```

```
65  Describe TopKCustomer_T;
66
```

Script Output x | Query Result x

Task completed in 0.003 seconds

```
Describe TopKCustomer_T
Name                 Null     Type
-------------------- -------- ------------
CUSTOMERID           NOT NULL NUMBER(11)
CUSTOMERNAME         NOT NULL VARCHAR2(25)
CUSTOMERPOSTALCODE            VARCHAR2(10)
CRANK                         NUMBER
ORDERFREQUENCY                NUMBER
RANKGENERATEDATE              DATE
```

Part 3.3

```
129  SET SERVEROUTPUT ON
130  CREATE OR REPLACE PROCEDURE TopKCustomer(k NUMBER) AS
131    rowCount NUMBER;
132    ci CHAR(9);
133    cn VARCHAR2(40);
134    cp VARCHAR2(40);
135    cr NUMBER:=1;
136    ofq VARCHAR2(40);
137    prevofq VARCHAR2(40):=NULL;
138
139    CURSOR c1 IS SELECT c.CustomerId, c.CustomerName, c.CustomerPostalCode, COUNT(o.OrderId) AS "Order Frequency"
140    FROM Customer_T c, Order_T o
141    WHERE o.CustomerId = c.CustomerId
142    GROUP BY c.CustomerId, c.CustomerName, c.CustomerPostalCode
143    ORDER BY COUNT(o.OrderId) DESC;
144  BEGIN
145    SELECT Count(*) INTO rowCount FROM Customer_T;
146    OPEN C1;
147    LOOP
148      FETCH c1 INTO ci, cn, cp, ofq;
149      IF (prevofq = NULL) OR (ofq != prevofq) THEN
150        cr:=cr+1;
151      END IF;
152      EXIT WHEN c1%NOTFOUND OR cr > k;
153      INSERT INTO TopKCustomer_T (CustomerId,CustomerName,CustomerPostalCode,CRank,OrderFrequency,RankGenerateDate)
154      VALUES (ci, cn, cp, cr, ofq, SYSDATE);
155      prevofq:=ofq;
156    END LOOP;
157    CLOSE C1;
158  END TopKCustomer;
159
```

Script Output | Query Result

Task completed in 0 seconds

```
PROCEDURE TOPKCUSTOMER compiled
```

```
160  DELETE FROM TopKCustomer_T;
161  EXECUTE TopKCustomer(2);
162  SELECT CRANK, ORDERFREQUENCY, CUSTOMERID,
163  CUSTOMERNAME, CUSTOMERPOSTALCODE, RANKGENERATEDATE
164  FROM TopKCustomer_T;
```

Script Output | Query Result

SQL | All Rows Fetched: 2 in 0.234 seconds

	CRANK	ORDERFREQUENCY	CUSTOMERID	CUSTOMERNAME	CUSTOMERPOSTALCODE	RANKGENERATEDATE
1	1	3	1	Contemporary Casuals	32601-2871	25-APR-17
2	2	2	15	Mountain Scenes	84403-4432	25-APR-17

```
166 DELETE FROM TopKCustomer_T;
167 EXECUTE TopKCustomer(3);
168 SELECT CRANK, ORDERFREQUENCY, CUSTOMERID,
169 CUSTOMERNAME, CUSTOMERPOSTALCODE, RANKGENERATEDATE
170 FROM TopKCustomer_T;
171
```

Script Output | Query Result | Query Result 1 | Query Result 2

SQL | All Rows Fetched: 9 in 0 seconds

	CRANK	ORDERFREQUENCY	CUSTOMERID	CUSTOMERNAME	CUSTOMERPOSTALCODE	RANKGENERATEDATE
1	1	3	1	Contemporary Casuals	32601-2871	25-APR-17
2	2	2	15	Mountain Scenes	84403-4432	25-APR-17
3	3	1	12	Battle Creek Furniture	49015-3401	25-APR-17
4	3	1	4	Eastern Furniture	07008-3188	25-APR-17
5	3	1	2	Value Furniture	75094-7743	25-APR-17
6	3	1	5	Impressions	94206-4056	25-APR-17
7	3	1	11	American Euro Lifestyles	07508-5621	25-APR-17
8	3	1	8	California Classics	96915-7754	25-APR-17
9	3	1	3	Home Furnishings	12209-1125	25-APR-17

Part 4.1

```
112 CREATE TABLE AuditProduct_T
113 (ProductID number(11),
114 ProductLineId number(11),
115 ProductDescription varchar2(50),
116 ProductFinish varchar2(20),
117 productStandardPrice number(6,2),
118 ChangeDate date,
119 ChangeStatus varchar2(10) Constraint Status_C
120 CHECK (ChangeStatus IN ('Insert','Update','Delete')));
121
122
```

Script Output | Query Result

Task completed in 0.031 seconds

```
table AUDITPRODUCT_T created.
```

```
122 CREATE TABLE AuditProductPrice_T
123  (ProductID number(11),
124  productOldPrice number(6,2),
125  productNewPrice number(6,2),
126  ChangeDate date);
```

Script Output | Query Result | Task completed in 0.016 seconds

```
table AUDITPRODUCTPRICE_T created.
```

Part 4.2

```
128 CREATE OR REPLACE TRIGGER AuditProductTrig
129   AFTER INSERT OR UPDATE OR DELETE ON Product_T
130   REFERENCING NEW AS NewTuple OLD AS OldTuple
131   FOR EACH ROW
132
133   BEGIN
134   CASE
135     WHEN INSERTING THEN
136       INSERT INTO AuditProduct_T (ProductId, ProductLineId, ProductDescription,
137       ProductFinish, ProductStandardPrice, ChangeDate, ChangeStatus)
138       VALUES (:NewTuple.ProductId, :NewTuple.ProductLineId, :NewTuple.ProductDescription, :NewTuple.ProductFinish,
139       :NewTuple.ProductStandardPrice, SYSDATE, 'Insert');
140     WHEN DELETING THEN
141       INSERT INTO AuditProduct_T (ProductId, ProductLineId, ProductDescription,
142       ProductFinish, ProductStandardPrice, ChangeDate, ChangeStatus)
143       VALUES (:OldTuple.ProductId, :OldTuple.ProductLineId, :OldTuple.ProductDescription, :OldTuple.ProductFinish,
144       :OldTuple.ProductStandardPrice, SYSDATE, 'Delete');
145     WHEN UPDATING THEN
146       INSERT INTO AuditProduct_T (ProductId, ProductLineId, ProductDescription,
147       ProductFinish, ProductStandardPrice, ChangeDate, ChangeStatus)
148       VALUES (:NewTuple.ProductId, :NewTuple.ProductLineId, :NewTuple.ProductDescription, :NewTuple.ProductFinish,
149       :NewTuple.ProductStandardPrice, SYSDATE, 'Update');
150       INSERT INTO AuditProductPrice_T (ProductId, ProductOldPrice, ProductNewPrice, ChangeDate)
151       VALUES (:NewTuple.ProductId, :OldTuple.ProductStandardPrice, :NewTuple.ProductStandardPrice, SYSDATE);
152   END CASE;
153   END;
```

Script Output | Query Result | Task completed in 0 seconds

```
TRIGGER AUDITPRODUCTTRIG compiled
```

Part 4.3

```
158 SELECT * FROM Product_T;
```

Query Result | SQL | All Rows Fetched: 8 in 0 seconds

	PRODUCTID	PRODUCTLINEID	PRODUCTDESCRIPTION	PRODUCTFINISH	PRODUCTSTANDARDPRICE
1	1	1	End Table	Cherry	175
2	2	2	Coffee Table	Natural Ash	200
3	3	2	Computer Desk	Natural Ash	375
4	4	3	Entertainment Center	Natural Maple	650
5	5	1	Writers Desk	Cherry	325
6	6	2	8-Drawer Desk	White Ash	750
7	7	2	Dining Table	Natural Ash	800
8	8	3	Computer Desk	Walnut	250

```
159 INSERT INTO Product_T  (ProductID, ProductDescription, ProductFinish, ProductStandardPrice, ProductLineID)
160 VALUES  (77, 'Computer Desk XL', 'Walnut', 275, 4);
```

Query Result | Script Output | Task completed in 0 seconds

```
1 rows inserted.
```

```
162 SELECT * FROM Product_T;
```

Script Output | Query Result | Query Result 1 | SQL | All Rows Fetched: 9 in 0 seconds

	PRODUCTID	PRODUCTLINEID	PRODUCTDESCRIPTION	PRODUCTFINISH	PRODUCTSTANDARDPRICE
1	1	1	End Table	Cherry	175
2	2	2	Coffee Table	Natural Ash	200
3	3	2	Computer Desk	Natural Ash	375
4	4	3	Entertainment Center	Natural Maple	650
5	5	1	Writers Desk	Cherry	325
6	6	2	8-Drawer Desk	White Ash	750
7	7	2	Dining Table	Natural Ash	800
8	8	3	Computer Desk	Walnut	250
9	77	4	Computer Desk XL	Walnut	275

```
163 SELECT * FROM AuditProduct_T;
164
165
```

Script Output | Query Result | Query Result 1 | Query Result 2 | SQL | All Rows Fetched: 1 in 0 seconds

	PRODUCTID	PRODUCTLINEID	PRODUCTDESCRIPTION	PRODUCTFINISH	PRODUCTSTANDARDPRICE	CHANGEDATE	CHANGESTATUS
1	77	4	Computer Desk XL	Walnut	275	23-APR-17	Insert

```
164  UPDATE Product_T SET ProductStandardPrice = 300 WHERE ProductId = 77;
165
166
167
```

Script Output | Query Result | Query Result 1 | Query Result 2

Task completed in 0.015 seconds

```
1 rows updated.
```

```
165  SELECT * FROM Product_T;
```

Script Output | Query Result | Query Result 1 | Query Result 2

SQL | All Rows Fetched: 9 in 0 seconds

	PRODUCTID	PRODUCTLINEID	PRODUCTDESCRIPTION	PRODUCTFINISH	PRODUCTSTANDARDPRICE
1	1	1	End Table	Cherry	175
2	2	2	Coffee Table	Natural Ash	200
3	3	2	Computer Desk	Natural Ash	375
4	4	3	Entertainment Center	Natural Maple	650
5	5	1	Writers Desk	Cherry	325
6	6	2	8-Drawer Desk	White Ash	750
7	7	2	Dining Table	Natural Ash	800
8	8	3	Computer Desk	Walnut	250
9	77	4	Computer Desk XL	Walnut	300

```
166  SELECT * FROM AuditProduct_T;
```

Script Output | Query Result | Query Result 1 | Query Result 2

SQL | All Rows Fetched: 2 in 0 seconds

	PRODUCTID	PRODUCTLINEID	PRODUCTDESCRIPTION	PRODUCTFINISH	PRODUCTSTANDARDPRICE	CHANGEDATE	CHANGESTATUS
1	77	4	Computer Desk XL	Walnut	275	23-APR-17	Insert
2	77	4	Computer Desk XL	Walnut	300	23-APR-17	Update

```
168  SELECT * FROM AuditProductPrice_T;
169
170
```

Script Output | Query Result | Query Result 1 | Query Result 2

SQL | All Rows Fetched: 1 in 0 seconds

	PRODUCTID	PRODUCTOLDPRICE	PRODUCTNEWPRICE	CHANGEDATE
1	77	275	300	23-APR-17

```
169 DELETE Product_T WHERE ProductId=77;
```

Script Output | Query Result | Query Result 1 | Query Resu

Task completed in 0 seconds

```
1 rows deleted.
```

```
170 SELECT * FROM Product_T;
```

Script Output | Query Result | Query Result 1 | Query Result 2

SQL | All Rows Fetched: 8 in 0 seconds

	PRODUCTID	PRODUCTLINEID	PRODUCTDESCRIPTION	PRODUCTFINISH	PRODUCTSTANDARDPRICE
1	1	1	End Table	Cherry	175
2	2	2	Coffee Table	Natural Ash	200
3	3	2	Computer Desk	Natural Ash	375
4	4	3	Entertainment Center	Natural Maple	650
5	5	1	Writers Desk	Cherry	325
6	6	2	8-Drawer Desk	White Ash	750
7	7	2	Dining Table	Natural Ash	800
8	8	3	Computer Desk	Walnut	250

```
171 SELECT * FROM AuditProduct_T;
172
```

Script Output | Query Result | Query Result 1 | Query Result 2

SQL | All Rows Fetched: 3 in 0 seconds

	PRODUCTID	PRODUCTLINEID	PRODUCTDESCRIPTION	PRODUCTFINISH	PRODUCTSTANDARDPRICE	CHANGEDATE	CHANGESTATUS
1	77	4	Computer Desk XL	Walnut	275	23-APR-17	Insert
2	77	4	Computer Desk XL	Walnut	300	23-APR-17	Update
3	77	4	Computer Desk XL	Walnut	300	23-APR-17	Delete

Part 5

```
Enter product finish: Natural Ash

Order Frequency     Product Id          Description         Finish
-------------------  -------------------  -------------------  -------------------
4                   3                   Computer Desk       Natural Ash

2                   7                   Dining Table        Natural Ash

2                   2                   Coffee Table        Natural Ash
```

```
Enter product finish: Cherry

Order Frequency     Product Id          Description         Finish
-------------------  -------------------  -------------------  -------------------
2                   1                   End Table           Cherry

1                   5                   Writers Desk        Cherry
```

```java
package huntmj01.cs364.hw4;
import oracle.jdbc.*;
import oracle.jdbc.pool.OracleDataSource;
import java.sql.*;
import java.util.Scanner;

public class Main {

	public static void main(String[] args) throws SQLException {

		// make the connection to the database
		OracleDataSource ods = new OracleDataSource();
		ods.setUser("SYSTEM");
		ods.setPassword("admin");
		ods.setURL("jdbc:oracle:thin:@//localhost:1521/xe");
		Connection conn = ods.getConnection();

		// retrieve the user input
		String input;
		Scanner keyboard = new Scanner(System.in);
		System.out.print("Enter product finish: ");
		input = keyboard.nextLine();
		System.out.println("");

		// write and execute the query
		Statement stmt = conn.createStatement();
		ResultSet rs =
		stmt.executeQuery("SELECT COUNT(o.OrderId) AS \"Order Frequency\",
		p.ProductId, p.ProductDescription, p.ProductFinish "
		+ "FROM Product_T p, Order_T o, OrderLine_T ol "
		+ "WHERE p.ProductId = ol.ProductId AND o.OrderId = ol.OrderId "
		+ "AND p.ProductFinish LIKE " + "'"+ input + "' "
		+ "GROUP BY p.ProductId, p.ProductDescription, p.ProductFinish "
		+ "ORDER BY COUNT(o.OrderId) DESC");

		// print header to the console
		System.out.printf("%-24s %-24s %-24s %-24s\n", "Order Frequency",
		"Product Id", " Description", "Finish");
		System.out.printf("%-24s %-24s %-24s %-24s\n", "--------------------",
		"--------------------", " --------------------", "--------------------
		");

		// loop through the result set
		while(rs.next()){
			// print out the row
			System.out.println(String.format("%-24s %-24s %-24s %-24s\n",
			rs.getString(1), rs.getString(2), rs.getString(3),
			rs.getString(4)));
		}

		// close the the keyboard and db connection
		keyboard.close();
		conn.close();
	}
}
```