School of Information Technology and Engineering (SITE)

Vellore Institute of Technology
(Deemed to be University under section 3 of UGC Act, 1956)

ONLINE SHOPPING PORTAL: Stop N Shop

ITE1003 DATABASE MANAGEMENT SYSTEMS
J COMPONENT - REVIEW 3

Submitted by:

David Emmanuel 18BIT0046
Harsh Raj 18BIT0072
Awani Kendurkar 18BIT0082
Vishal Vikram Das 19BIT0052

Submitted to:

Prof. Bimal Kumar Ray

INTRODUCTION

The following database management system represents the working of an online shopping portal as a company oriented system. It helps the company to track the records of any customer and any product it provides. The company stores various details of the customers, keeps a record of all the payments made, their respective orders and all the offers they used up. The product a particular customer purchases can also be tracked to its inventory i.e. the stock, the supplier details and all the analysis a finance firm does over it.

Here every customer detail will be recorded i.e. name, email ID, password, address, phone number etc. Each customer shall have a customer ID to uniquely identify his/her transactions.

A list of products purchased by a customer is available and identified using product ID. The product history from the inventory could also be traced via stock-type. For analysis, finance firm uses composite primary keys to identify and rectify the records. For transparency policy the company also maintains supplier records with a unique supplier ID.

For checking the transaction details company can rectify the payments made by a customer with the help of a unique payment ID. Through payment details, the company can track the offers applied for the particular payment and the discount gained.

The placed order will be mentioned in order table with unique order ID and will also show order date and expected date of product delivery.

The design for this company oriented database management system is made by keeping in mind:

- Utilising less memory
- Avoiding redundancy
- Persistence in database
- Removing anomalies and performing various operations
- Providing concurrent user interface

The title we are assigning for this database is Stop N Shop - Online Shopping Portal.

DATA AND FUNCTIONAL REQUIREMENTS

Data to be stored:

• **Customer information**: The website is based around them. They are also the biggest part of our website. This is where a client's information is stored. This will store basic info of the customer, such as a unique ID, name, username, address, phone number, contact email, password.

• **Product information**: This stores the information about the product. This has a unique ID, name, price, brand.

• **Order information**: Any order that is placed on this website needs to be stored as an invoice for future references. The order may include a unique order ID, order date, and expected date of delivery of the order.

• **Payment information:** The details about payments done for the orders will be stored here. We store the unique payment ID, amount paid and mode of payment (type).

• **Offers provided:** This will cover the offer and discounts of various types that are available on a particular product. Here we will store the coupon number and the discount received on application of this coupon.

• **Supplier information:** The details about the suppliers, i.e. the person/company that wishes to sell their products through this website, will be stored here. The details stored will be name, supplier ID and their address.

• **Inventory information:** The suppliers' inventory information will be stored here. The stock type, stock name and stock number will constitute this data.

• **Finance firm**: This is the finance department's data. The details stored here will be firm type, firm ID and the tax associated with the purchase.

*The above data has to be defined as follows (**domain constraints**):*

- Customer ID: String starting with 'C' followed by 5 digits
- Customer username: String which is unique and consists of 8 characters.
- Customer name: String
- Customer address: String of length around 50
- Customer contact number: String of length 10
- Customer email ID: String of the form "abc@xyz.com"
- Customer password: String consisting of a combination of letters and digits, with at least one uppercase letter, one lowercase letter, and one digit.
- Product ID: String starting with 'PR' followed by 4 numbers
- Product name: String
- Product price: Number

- Product brand: String recognising a brand
- Supplier ID: String starting with ‘S’ followed by 5 numbers
- Order ID: String starting with ‘O’ followed by 5 numbers
- Order date: Date
- Expected delivery date: Date
- Payment ID: String starting with ‘P’ followed by 5 numbers
- Amount paid: Number
- Payment type: String
- Coupon number: String starting with ‘DIS’ followed by 5 numbers
- Discount percent: Number greater than zero and less than 100
- Supplier name: String
- Stock type: String
- Stock number: Number
- Firm type: String
- Tax percent: Number greater than zero and less than 100

Following relationships will be accounted for:

1. The relationship PLACED_BY exists between CUSTOMER and ORDER. Cardinality ratio is 1:N for CUSTOMER to ORDER.
2. The relationship PAID exists between ORDER and PAYMENT. Cardinality ratio is 1:1 for ORDER to PAYMENT.
3. The relationship APPLIED_FOR exists between OFFER and PAYMENT. Cardinality ratio is 1:N for OFFER to PAYMENT.
4. The relationship REDEEMS exists between OFFER and STUDENT. Cardinality ratio is N:M for OFFER to STUDENT.
5. The relationship ACCESSES exists between CUSTOMER and PRODUCT. Cardinality ratio is M:N for CUSTOMER to PRODUCT.
6. The relationship INCLUDES exists between STOCK and PRODUCT. Cardinality ratio is 1:N for STOCK to PRODUCT.
7. The relationship SUPPLIES exists between STOCK and SUPPLIER. Cardinality ratio is N:1 for STOCK to SUPPLIER.
8. The relationship FACILITATES exists between STOCK and FINANCE_FIRM. Cardinality ratio is N:1 for STOCK to FINANCE_FIRM.

Functional requirements:

Data removal:

1. A customer can delete his/her account and the data associated with it.
2. An order can be deleted, if the customer wishes to cancel it.
3. A product can be deleted from the database if it is no longer available.
4. A supplier can delete their account if they no longer wish to supply their products to the company.

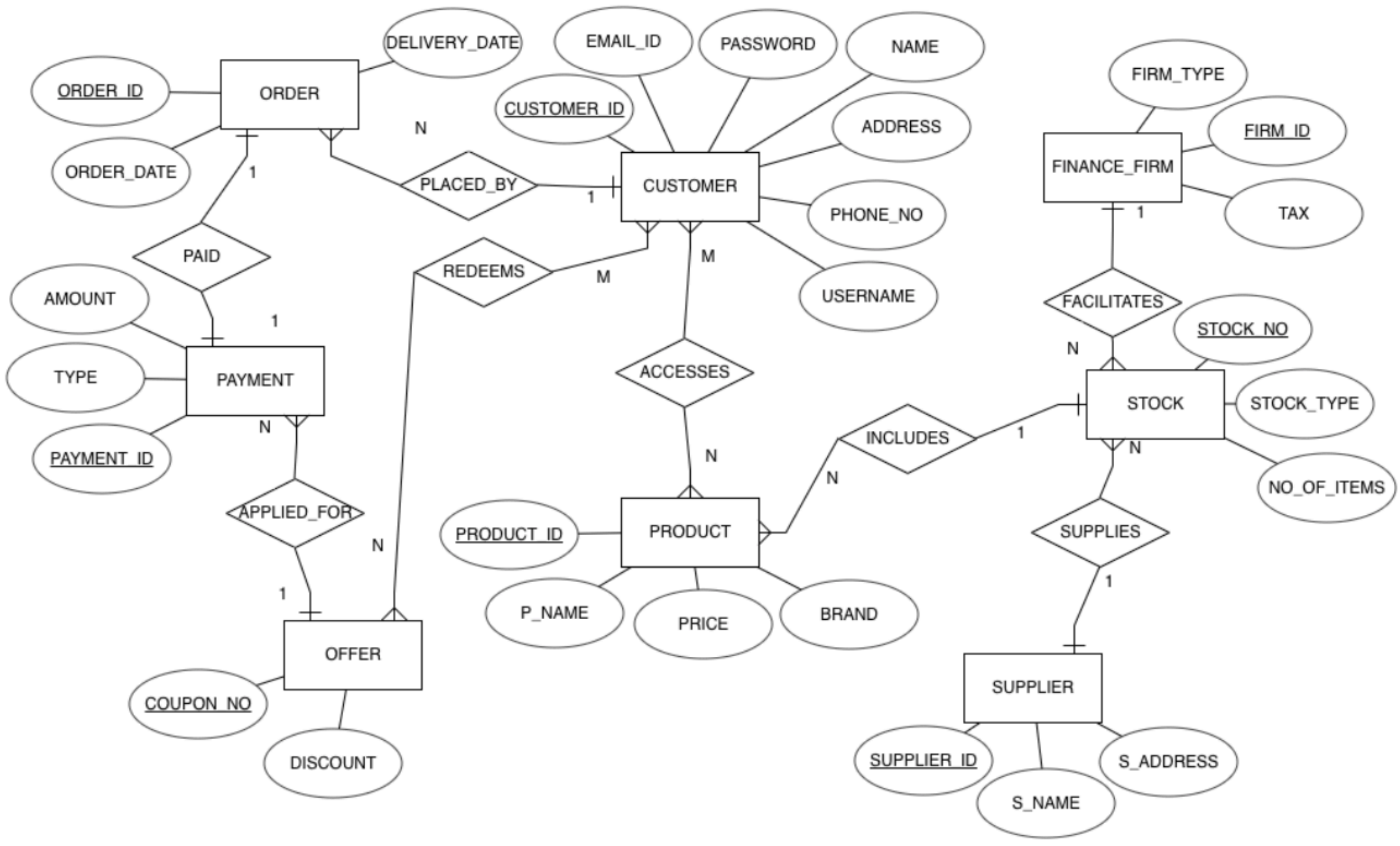
Data modification:

1. A customer can modify his/her details such as the address or phone number, if it needs updation.
2. A product's details can be changed, such as its price.
3. An offer can be modified, if the discount it offers changes.
4. A finance firm can change its details, such as the tax it levies.

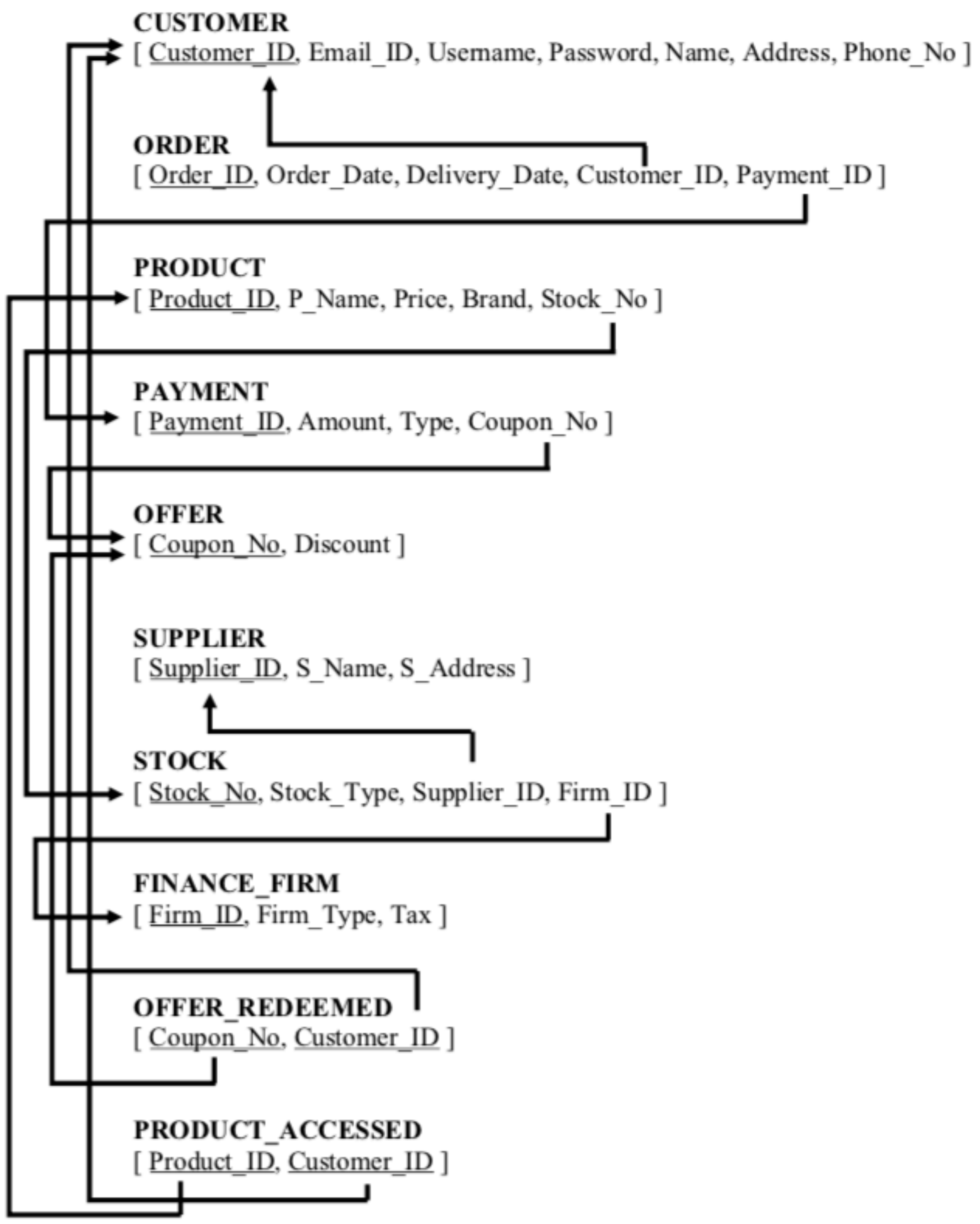
Data retrieval:

1. Customers can access products and view the details.
2. Customers can also view their order and payment details.
3. The list of products included in a stock can be retrieved.
4. The list of stocks that a supplier provides can be retrieved.
5. The list of orders place by a customer can be retrieved.
6. The list of offers availed by a particular customer can be retrieved.
7. The list of stocks that a finance firm facilitates can be retrieved.
8. The list of different payments that availed a particular offer can be retrieved.

E-R DIAGRAM



RELATIONAL MODEL

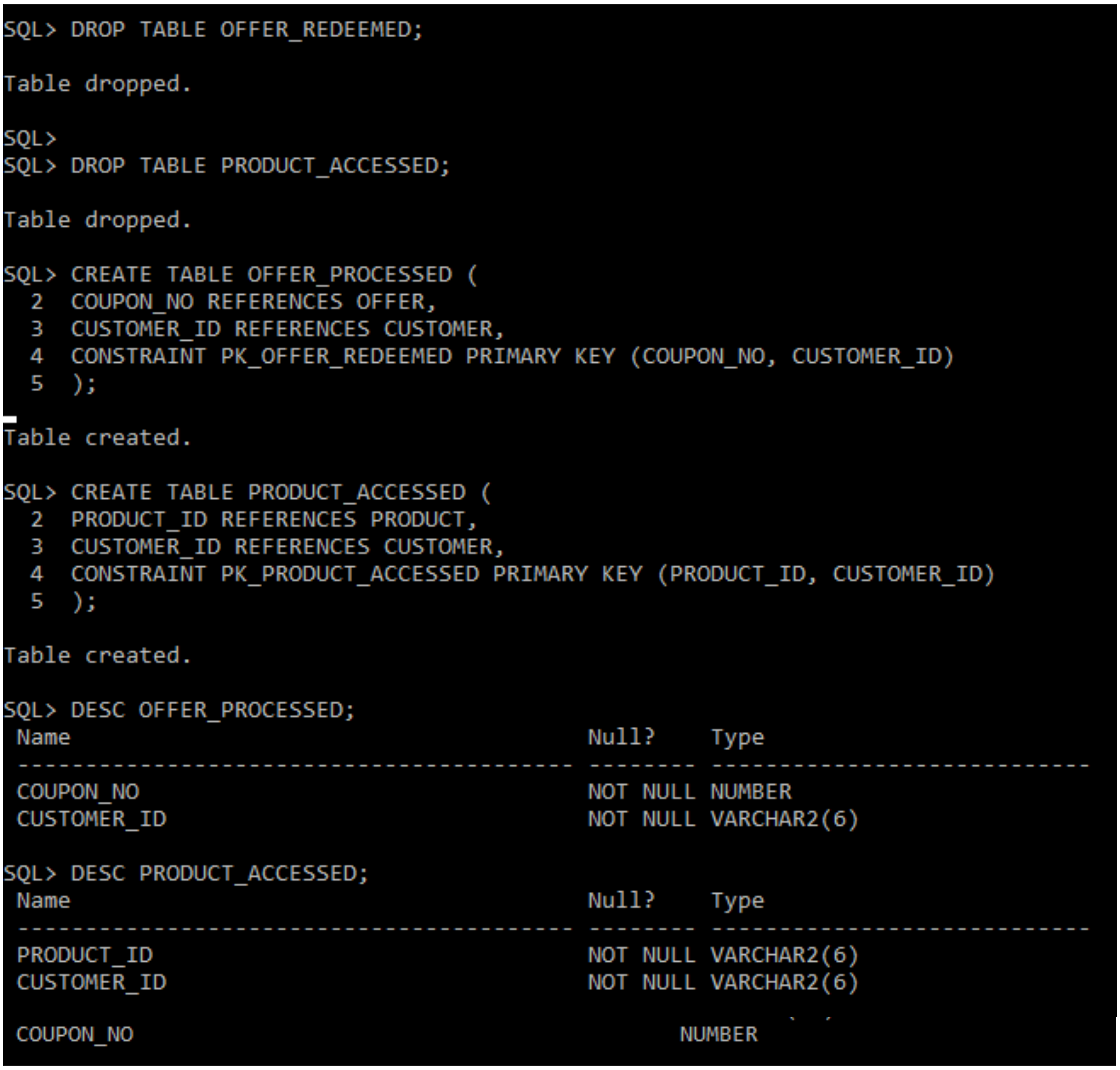

CUSTOMER
[Customer_ID, Email_ID, Username, Password, Name, Address, Phone_No]
ORDER
[Order_ID, Order_Date, Delivery_Date, Customer_ID, Payment_ID]
PRODUCT
[Product_ID, P_Name, Price, Brand, Stock_No]
PAYMENT
[Payment_ID, Amount, Type, Coupon_No]
OFFER
[Coupon_No, Discount]
SUPPLIER
[Supplier_ID, S_Name, S_Address]
STOCK
[Stock_No, Stock_Type, Supplier_ID, Firm_ID]
FINANCE_FIRM
[Firm_ID, Firm_Type, Tax]
OFFER_REDEEMED
[Coupon_No, Customer_ID]
PRODUCT_ACCESSED
[Product_ID, Customer_ID]

IMPLEMENTATION

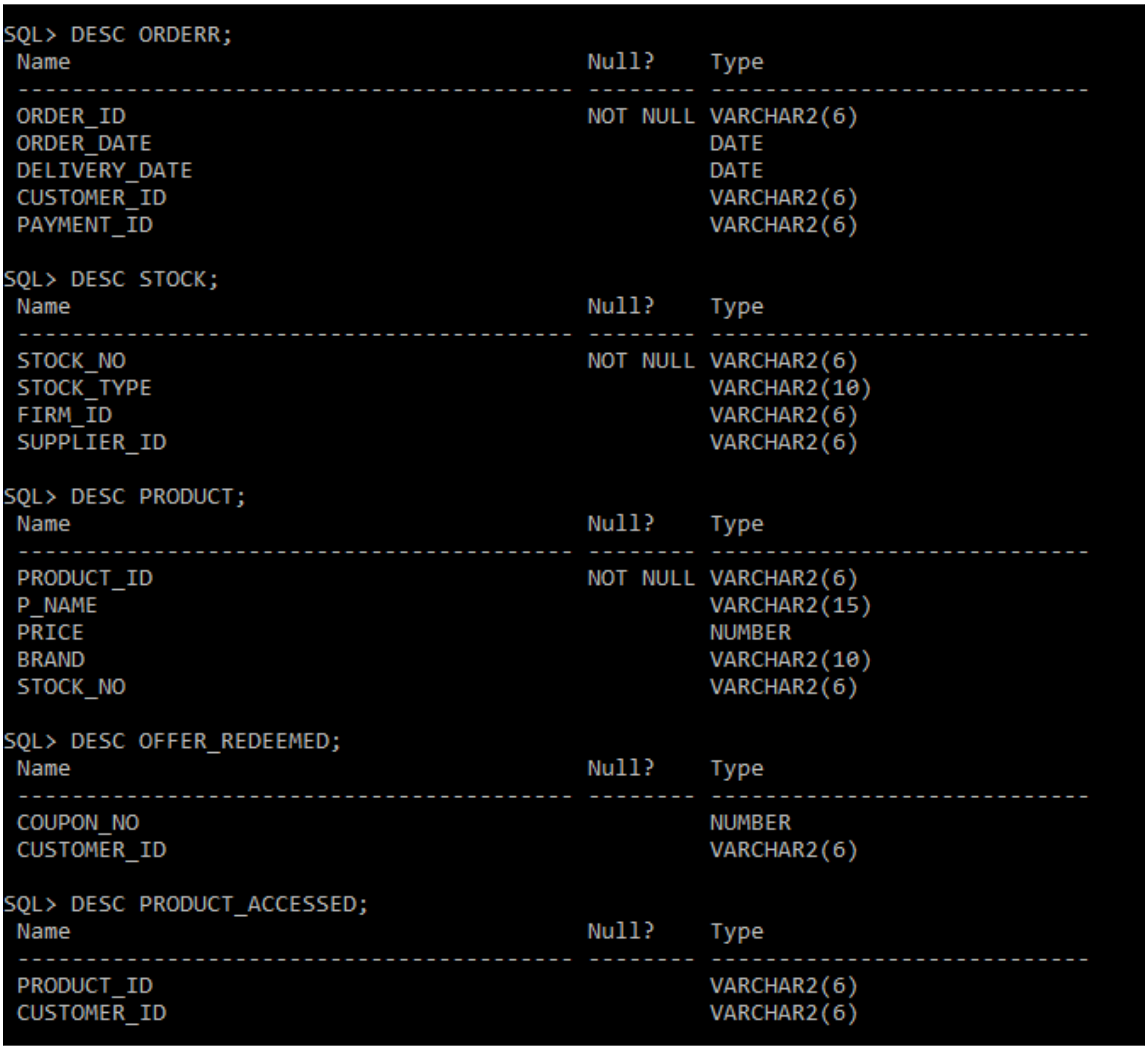
```
CREATE TABLE CUSTOMER
(
  CUSTOMER_ID VARCHAR(6),
  USERNAME VARCHAR(8),
  EMAIL_ID VARCHAR(25),
  PASSWORD VARCHAR(8),
  NAME VARCHAR(30),
  ADDRESS VARCHAR(50),
  PHONE_NO VARCHAR(10),
  CONSTRAINT PK_CUSTOMER PRIMARY KEY (CUSTOMER_ID)
);

CREATE TABLE OFFER
(
  COUPON_NO NUMBER,
  DISCOUNT NUMBER,
  CONSTRAINT PK_OFFER PRIMARY KEY (COUPON_NO)
);

CREATE TABLE SUPPLIER
(
  SUPPLIER_ID VARCHAR(6),
  S_NAME VARCHAR(15),
  S_ADDRESS VARCHAR(50),
  CONSTRAINT PK_SUPPLIER PRIMARY KEY (SUPPLIER_ID)
);

CREATE TABLE FINANCE_FIRM
(
  FIRM_ID VARCHAR(6),
  FIRM_TYPE VARCHAR(10),
  TAX NUMBER,
  CONSTRAINT PK_FINANCE_FIRM PRIMARY KEY (FIRM_ID)
);

CREATE TABLE PAYMENT
(
  PAYMENT_ID VARCHAR(6),
  AMOUNT NUMBER,
  TYPE VARCHAR(10),
  CONSTRAINT PK_PAYMENT PRIMARY KEY (PAYMENT_ID),
  CONSTRAINT FK_PAYMENT FOREIGN KEY (COUPON_NO) REFERENCES OFFER
     (COUPON_NO)
);
```

```
CREATE TABLE ORDERR
(
  ORDER_ID VARCHAR(6),
  ORDER_DATE DATE,
  DELIVERY_DATE DATE,
  CONSTRAINT PK_ORDERR PRIMARY KEY (ORDER_ID),
  CONSTRAINT FK1_ORDERR FOREIGN KEY (CUSTOMER_ID) REFERENCES
     CUSTOMER (CUSTOMER_ID),
  CONSTRAINT FK2_ORDERR FOREIGN KEY (PAYMENT_ID) REFERENCES
     PAYMENT (PAYMENT_ID)
);

CREATE TABLE STOCK
(
  STOCK_NO VARCHAR(6,
  STOCK_TYPE VARCHAR(10),
  CONSTRAINT PK_STOCK PRIMARY KEY (STOCK_NO),
  CONSTRAINT FK1_STOCK FOREIGN KEY (FIRM_ID) REFERENCES
     FINANCE_FIRM (FIRM_ID),
  CONSTRAINT FK2_STOCK FOREIGN KEY (SUPPLIER_ID) REFERENCES
     SUPPLIER (SUPPLIER_ID)
);

CREATE TABLE PRODUCT
(
  PRODUCT_ID VARCHAR(6),
  P_NAME VARCHAR(15),
  PRICE NUMBER,
  BRAND VARCHAR(10),
  CONSTRAINT PK_PRODUCT PRIMARY KEY (PRODUCT_ID),
  CONSTRAINT FK_PRODUCT FOREIGN KEY (STOCK_NO) REFERENCES STOCK
     (STOCK_NO)
);

CREATE TABLE OFFER_REDEEMED
(
  COUPON_NO REFERENCES OFFER,
  CUSTOMER_ID REFERENCES CUSTOMER,
  CONSTRAINT PK_OFFER_REDEEMED PRIMARY KEY (COUPON_NO,CUSTOMER_ID)
);

CREATE TABLE PRODUCT_ACCESSED
(
  PRODUCT_ID REFERENCES PRODUCT,
  CUSTOMER_ID REFERENCES CUSTOMER,
  CONSTRAINT PK_PRODUCT_ACCESSED PRIMARY KEY
     (PRODUCT_ID,CUSTOMER_ID)
);
```

Changes made for review 2:



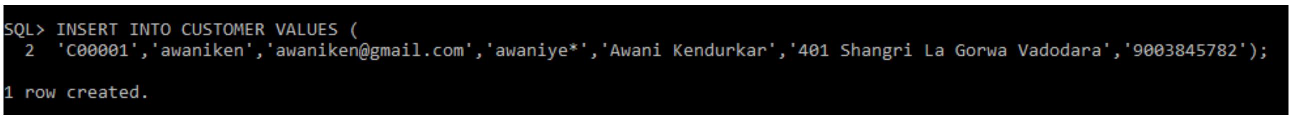
```
SQL> DROP TABLE OFFER_REDEEMED;

Table dropped.

SQL>
SQL> DROP TABLE PRODUCT_ACCESSED;

Table dropped.

SQL> CREATE TABLE OFFER_PROCESSED (
  2  COUPON_NO REFERENCES OFFER,
  3  CUSTOMER_ID REFERENCES CUSTOMER,
  4  CONSTRAINT PK_OFFER_REDEEMED PRIMARY KEY (COUPON_NO, CUSTOMER_ID)
  5  );

Table created.

SQL> CREATE TABLE PRODUCT_ACCESSED (
  2  PRODUCT_ID REFERENCES PRODUCT,
  3  CUSTOMER_ID REFERENCES CUSTOMER,
  4  CONSTRAINT PK_PRODUCT_ACCESSED PRIMARY KEY (PRODUCT_ID, CUSTOMER_ID)
  5  );

Table created.

SQL> DESC OFFER_PROCESSED;
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 COUPON_NO                                 NOT NULL NUMBER
 CUSTOMER_ID                               NOT NULL VARCHAR2(6)

SQL> DESC PRODUCT_ACCESSED;
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 PRODUCT_ID                                NOT NULL VARCHAR2(6)
 CUSTOMER_ID                               NOT NULL VARCHAR2(6)

 COUPON_NO                                                 NUMBER
```



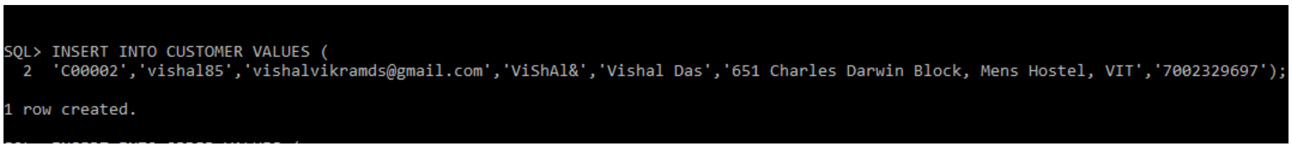
```
SQL> DESC ORDERR;
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 ORDER_ID                                  NOT NULL VARCHAR2(6)
 ORDER_DATE                                         DATE
 DELIVERY_DATE                                      DATE
 CUSTOMER_ID                                        VARCHAR2(6)
 PAYMENT_ID                                         VARCHAR2(6)

SQL> DESC STOCK;
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 STOCK_NO                                  NOT NULL VARCHAR2(6)
 STOCK_TYPE                                         VARCHAR2(10)
 FIRM_ID                                            VARCHAR2(6)
 SUPPLIER_ID                                        VARCHAR2(6)

SQL> DESC PRODUCT;
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 PRODUCT_ID                                NOT NULL VARCHAR2(6)
 P_NAME                                             VARCHAR2(15)
 PRICE                                              NUMBER
 BRAND                                              VARCHAR2(10)
 STOCK_NO                                           VARCHAR2(6)

SQL> DESC OFFER_REDEEMED;
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 COUPON_NO                                          NUMBER
 CUSTOMER_ID                                        VARCHAR2(6)

SQL> DESC PRODUCT_ACCESSED;
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 PRODUCT_ID                                         VARCHAR2(6)
 CUSTOMER_ID                                        VARCHAR2(6)
```

```
INSERT INTO CUSTOMER VALUES
('C00001','awaniken','awaniken@gmail.com','awaniye*','Awani
Kendurkar','401 Shangri La Gorwa Vadodara','9003845782');

INSERT INTO CUSTOMER VALUES
('C00002','vishal85','vishalvikramds@gmail.com','ViShAl&','Vishal
Das','651 Charles Darwin Block, Mens Hostel, VIT','7002329697');

INSERT INTO OFFER VALUES
(23891,15);

INSERT INTO OFFER VALUES
(71629,22);

INSERT INTO SUPPLIER VALUES
('S00001','Adidas','23 MG Road, Gurgaon');

INSERT INTO SUPPLIER VALUES
('S00002','Nike','2 Dutt Nagar, Noida');
```

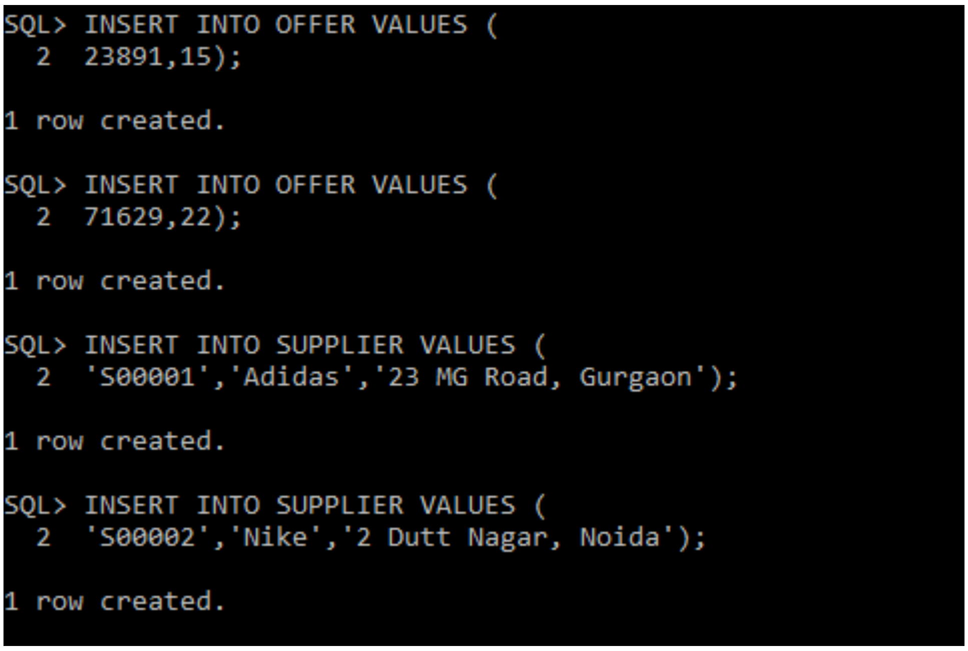
```
INSERT INTO FINANCE_FIRM VALUES
(‘F00001’,’XYZ’,10);

INSERT INTO FINANCE_FIRM VALUES
(‘F00002’,’ABC’,10);

INSERT INTO PAYMENT VALUES
(‘P00001’,4599,’COD’,23891);

INSERT INTO PAYMENT VALUES
(‘P00002’,2850,’PayTM’,71629);

INSERT INTO ORDERR VALUES
(‘O00001’,’25-JUN-2019’,’30-JUN-2019’,’C00001’,‘P00001’);

INSERT INTO ORDERR VALUES
(‘O00002’,’25-JUL-2019’,’30-JUL-2019’,’C00002’,‘P00002’);

INSERT INTO STOCK VALUES
(‘8912’,’Clothes’,’F00001’,‘S00001’);

INSERT INTO STOCK VALUES
(‘8916’,’Shoes’,’F00002’,’S00002’);

INSERT INTO PRODUCT VALUES
(‘PR0001’,’Grey T-shirt’,4599,’ADIDAS’,’8912’);

INSERT INTO PRODUCT VALUES
(‘PR0002’,’Black Ace’,2850,’NIKE’,‘8916’);

INSERT INTO OFFER_REDEEMED VALUES
(23891,’C00001’);

INSERT INTO OFFER_REDEEMED VALUES
(71629,’C00002’);

INSERT INTO PRODUCT_ACCESSED VALUES
(‘PR0001’,’C00001’);

INSERT INTO PRODUCT_ACCESSED VALUES
(‘PR0002’,’C00002’);
```



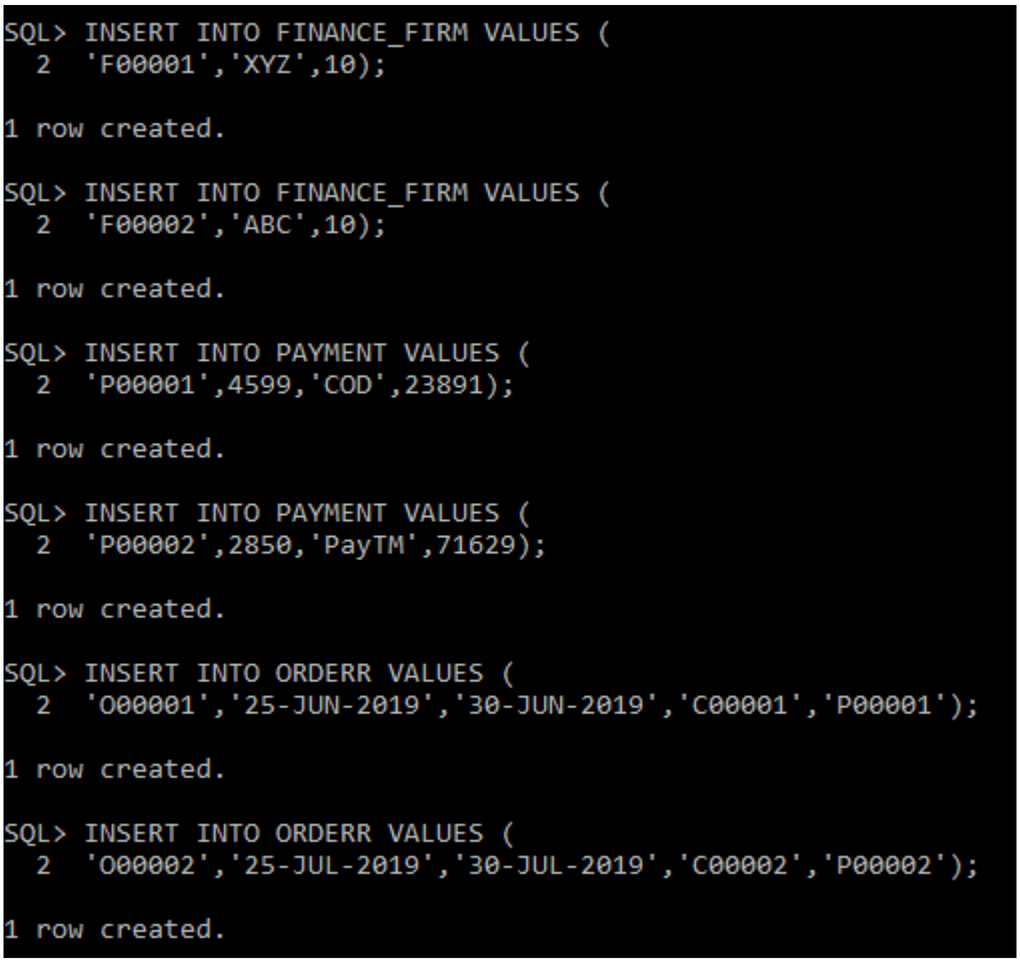
```
SQL> INSERT INTO CUSTOMER VALUES (
  2  'C00001','awaniken','awaniken@gmail.com','awaniye*','Awani Kendurkar','401 Shangri La Gorwa Vadodara','9003845782');

1 row created.
```



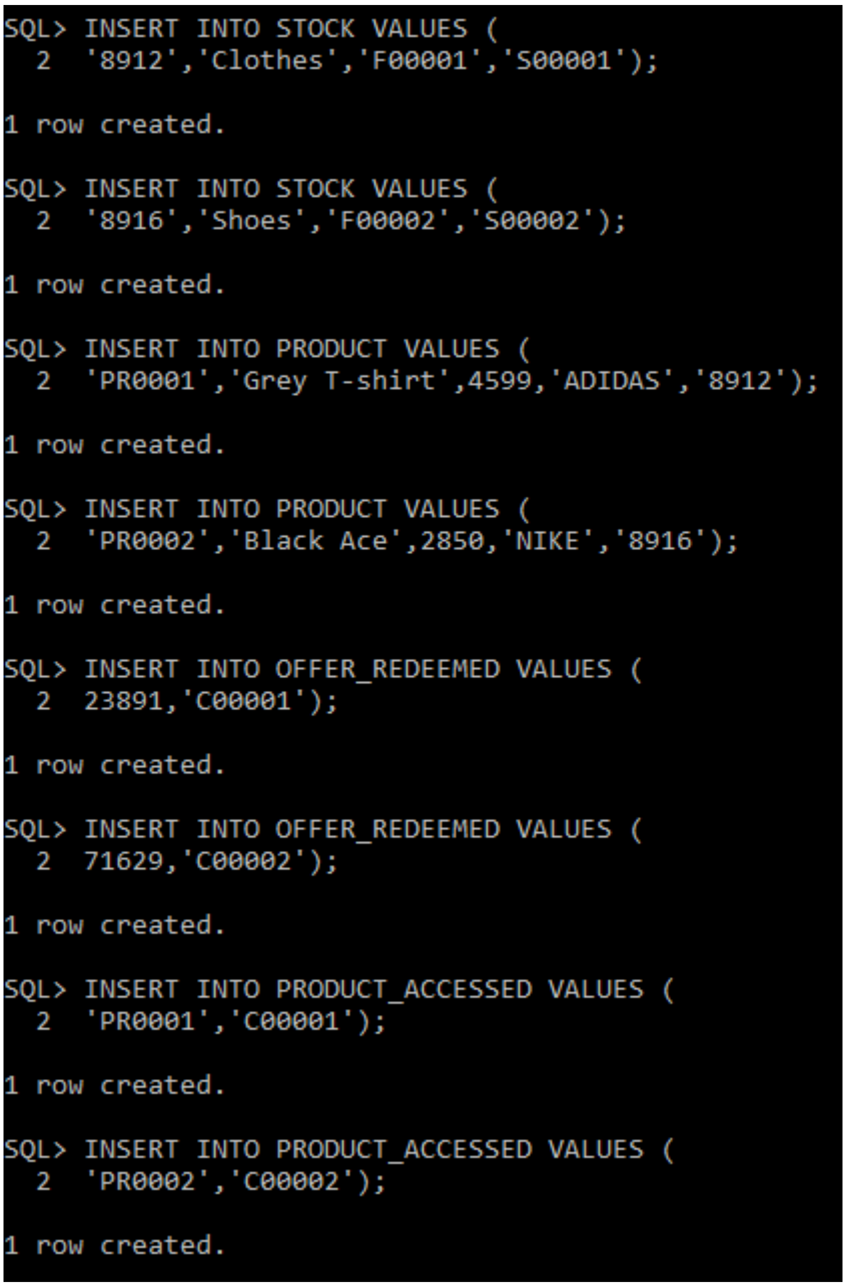
```
SQL> INSERT INTO CUSTOMER VALUES (
  2  'C00002','vishal85','vishalvikramds@gmail.com','ViShAl&','Vishal Das','651 Charles Darwin Block, Mens Hostel, VIT','7002329697');

1 row created.
```

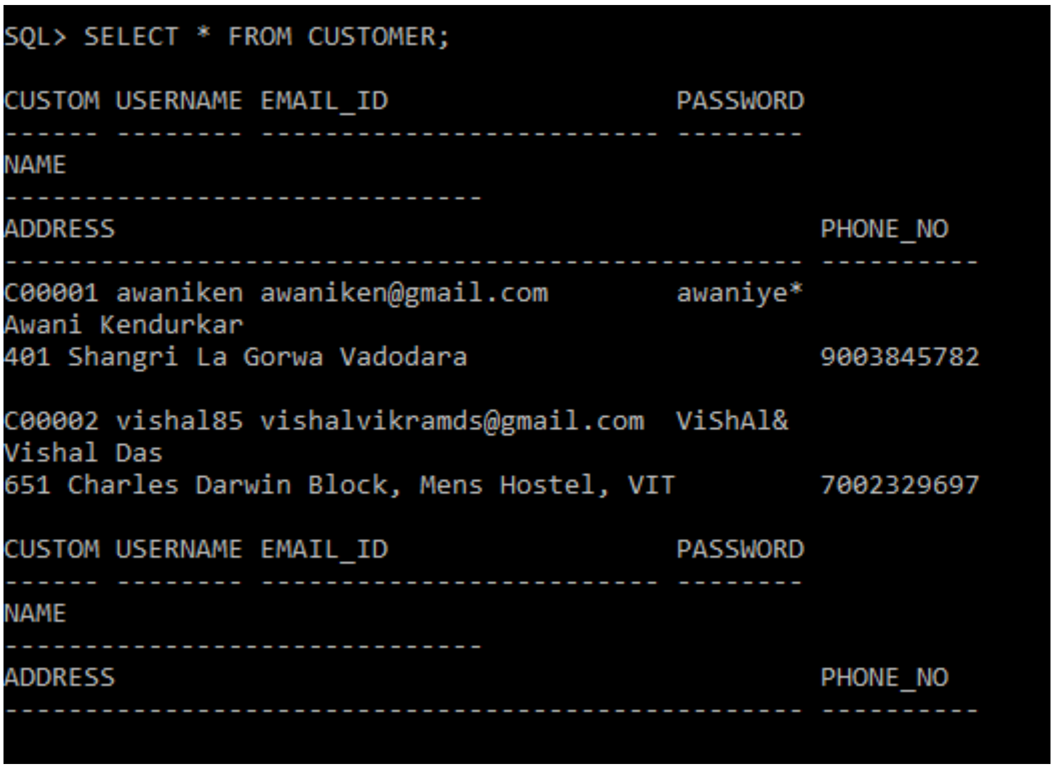


```
SQL> INSERT INTO OFFER VALUES (
  2  23891,15);

1 row created.

SQL> INSERT INTO OFFER VALUES (
  2  71629,22);

1 row created.

SQL> INSERT INTO SUPPLIER VALUES (
  2  'S00001','Adidas','23 MG Road, Gurgaon');

1 row created.

SQL> INSERT INTO SUPPLIER VALUES (
  2  'S00002','Nike','2 Dutt Nagar, Noida');

1 row created.
```

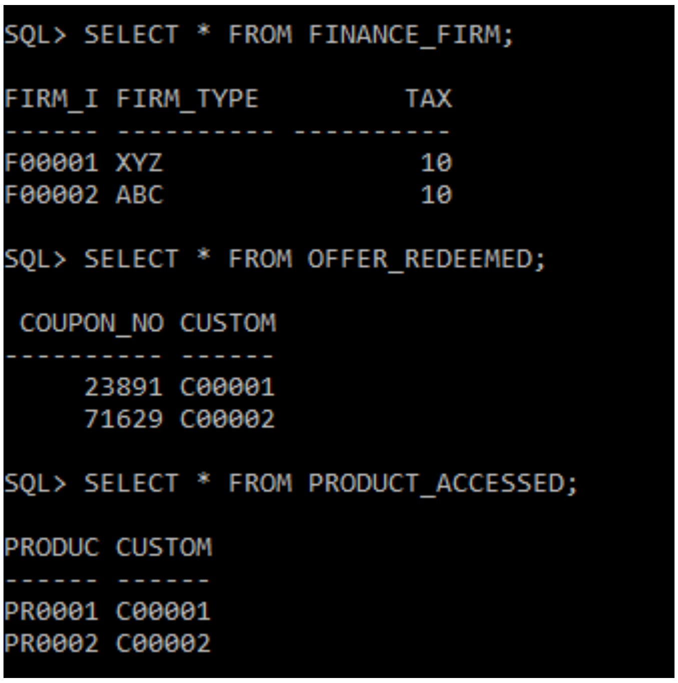


```
SQL> INSERT INTO FINANCE_FIRM VALUES (
  2  'F00001','XYZ',10);

1 row created.

SQL> INSERT INTO FINANCE_FIRM VALUES (
  2  'F00002','ABC',10);

1 row created.

SQL> INSERT INTO PAYMENT VALUES (
  2  'P00001',4599,'COD',23891);

1 row created.

SQL> INSERT INTO PAYMENT VALUES (
  2  'P00002',2850,'PayTM',71629);

1 row created.

SQL> INSERT INTO ORDERR VALUES (
  2  'O00001','25-JUN-2019','30-JUN-2019','C00001','P00001');

1 row created.

SQL> INSERT INTO ORDERR VALUES (
  2  'O00002','25-JUL-2019','30-JUL-2019','C00002','P00002');

1 row created.
```

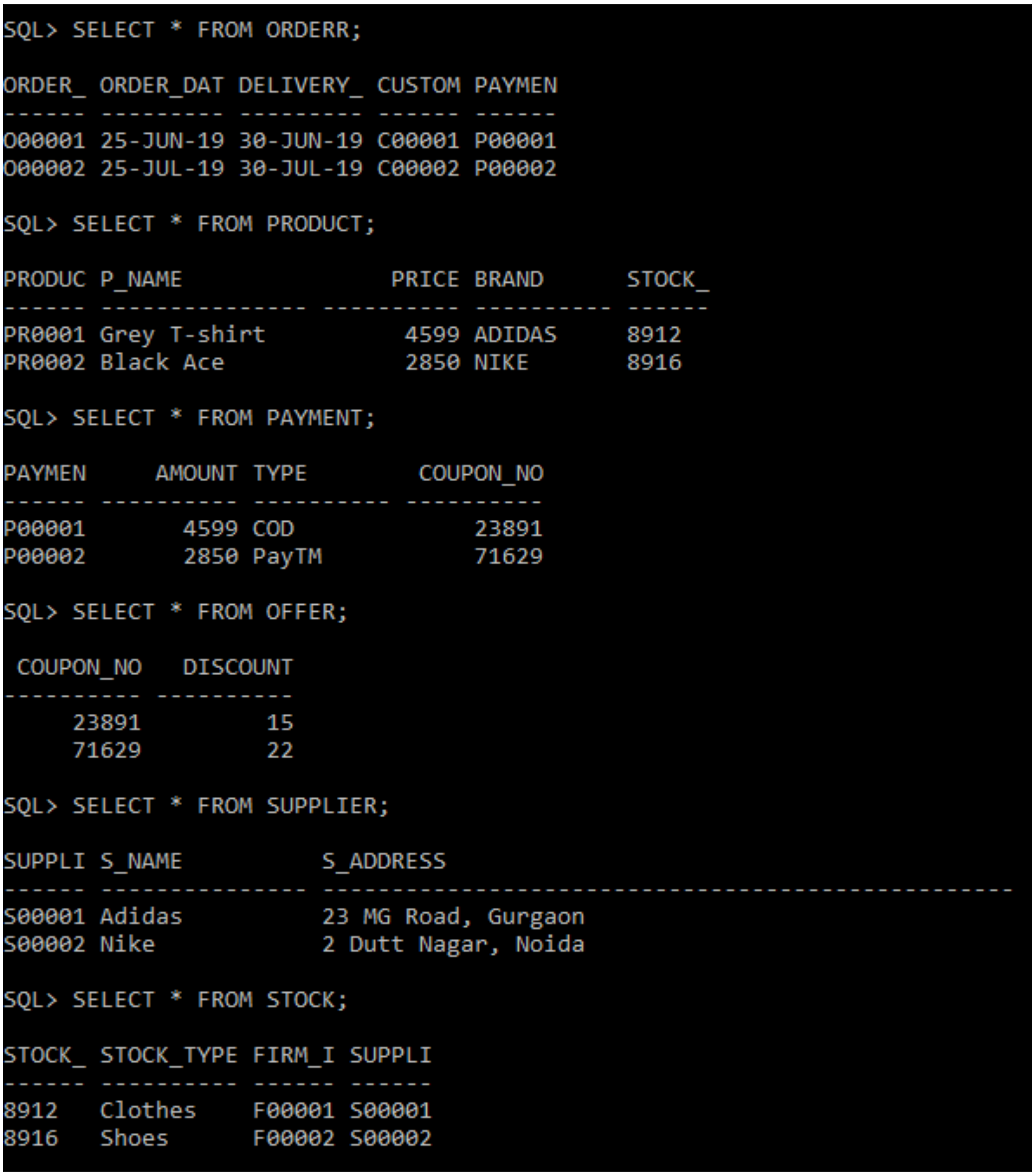

```
SQL> INSERT INTO STOCK VALUES (
  2  '8912','Clothes','F00001','S00001');

1 row created.

SQL> INSERT INTO STOCK VALUES (
  2  '8916','Shoes','F00002','S00002');

1 row created.

SQL> INSERT INTO PRODUCT VALUES (
  2  'PR0001','Grey T-shirt',4599,'ADIDAS','8912');

1 row created.

SQL> INSERT INTO PRODUCT VALUES (
  2  'PR0002','Black Ace',2850,'NIKE','8916');

1 row created.

SQL> INSERT INTO OFFER_REDEEMED VALUES (
  2  23891,'C00001');

1 row created.

SQL> INSERT INTO OFFER_REDEEMED VALUES (
  2  71629,'C00002');

1 row created.

SQL> INSERT INTO PRODUCT_ACCESSED VALUES (
  2  'PR0001','C00001');

1 row created.

SQL> INSERT INTO PRODUCT_ACCESSED VALUES (
  2  'PR0002','C00002');

1 row created.
```



```
SQL> SELECT * FROM CUSTOMER;

CUSTOM USERNAME EMAIL_ID                  PASSWORD
------ -------- ------------------------- --------
NAME
------------------------------
ADDRESS                                             PHONE_NO
-------------------------------------------------- ----------
C00001 awaniken awaniken@gmail.com        awaniye*
Awani Kendurkar
401 Shangri La Gorwa Vadodara                       9003845782

C00002 vishal85 vishalvikramds@gmail.com  ViShAl&
Vishal Das
651 Charles Darwin Block, Mens Hostel, VIT          7002329697

CUSTOM USERNAME EMAIL_ID                  PASSWORD
------ -------- ------------------------- --------
NAME
------------------------------
ADDRESS                                             PHONE_NO
-------------------------------------------------- ----------
```



```
SQL> SELECT * FROM FINANCE_FIRM;

FIRM_I FIRM_TYPE         TAX
------ ---------- ----------
F00001 XYZ                10
F00002 ABC                10

SQL> SELECT * FROM OFFER_REDEEMED;

 COUPON_NO CUSTOM
---------- ------
     23891 C00001
     71629 C00002

SQL> SELECT * FROM PRODUCT_ACCESSED;

PRODUC CUSTOM
------ ------
PR0001 C00001
PR0002 C00002
```

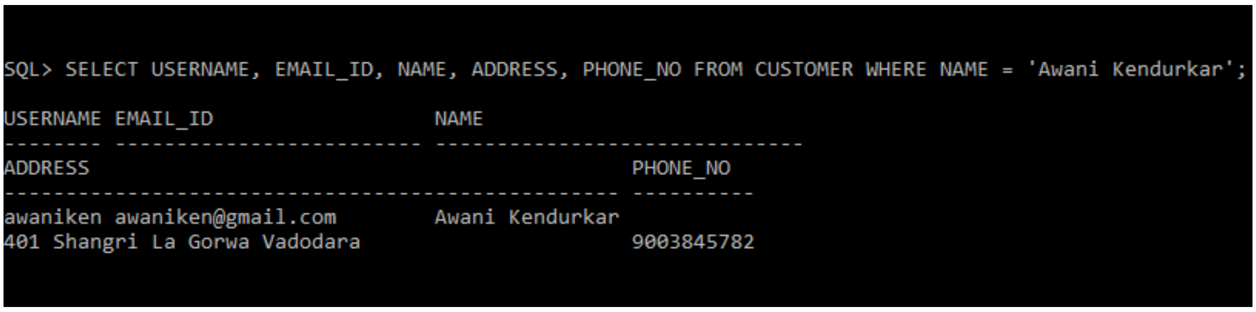


```
SQL> SELECT * FROM ORDERR;

ORDER_ ORDER_DAT DELIVERY_ CUSTOM PAYMEN
------ --------- --------- ------ ------
O00001 25-JUN-19 30-JUN-19 C00001 P00001
O00002 25-JUL-19 30-JUL-19 C00002 P00002

SQL> SELECT * FROM PRODUCT;

PRODUC P_NAME               PRICE BRAND      STOCK_
------ --------------- ---------- ---------- ------
PR0001 Grey T-shirt          4599 ADIDAS     8912
PR0002 Black Ace             2850 NIKE       8916

SQL> SELECT * FROM PAYMENT;

PAYMEN     AMOUNT TYPE        COUPON_NO
------ ---------- ---------- ----------
P00001       4599 COD             23891
P00002       2850 PayTM           71629

SQL> SELECT * FROM OFFER;

 COUPON_NO   DISCOUNT
---------- ----------
     23891         15
     71629         22

SQL> SELECT * FROM SUPPLIER;

SUPPLI S_NAME          S_ADDRESS
------ --------------- --------------------------------------------------
S00001 Adidas          23 MG Road, Gurgaon
S00002 Nike            2 Dutt Nagar, Noida

SQL> SELECT * FROM STOCK;

STOCK_ STOCK_TYPE FIRM_I SUPPLI
------ ---------- ------ ------
8912   Clothes    F00001 S00001
8916   Shoes      F00002 S00002
```

QUERY, DELETE & UPDATE

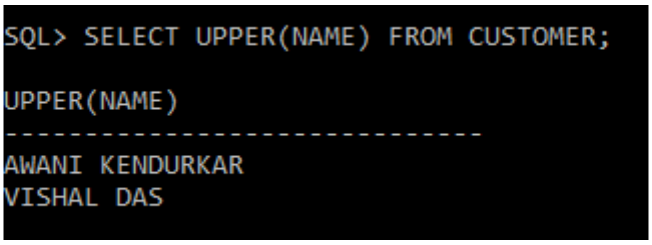
Data retrieval:

1. Select USERNAME, EMAIL_ID, NAME, ADDRESS and PHONE_NO from table CUSTOMER where NAME = ‘Awani Kendurkar’.



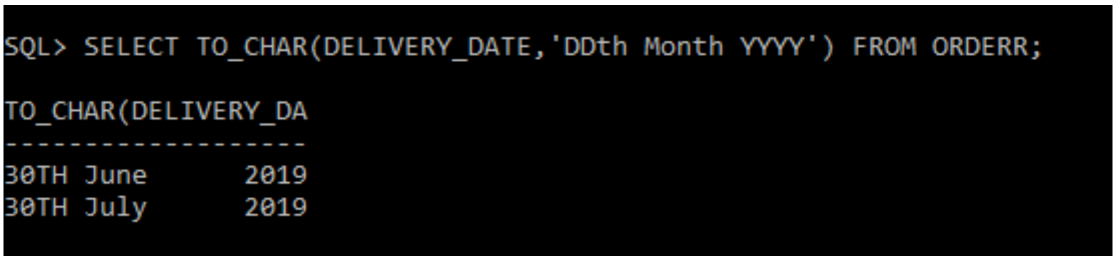
```
SQL> SELECT USERNAME, EMAIL_ID, NAME, ADDRESS, PHONE_NO FROM CUSTOMER WHERE NAME = 'Awani Kendurkar';

USERNAME EMAIL_ID                 NAME
-------- ------------------------ ------------------------------
ADDRESS                                         PHONE_NO
----------------------------------------------- ---------
awaniken awaniken@gmail.com       Awani Kendurkar
401 Shangri La Gorwa Vadodara                   9003845782
```

2. Select customer names in uppercase.



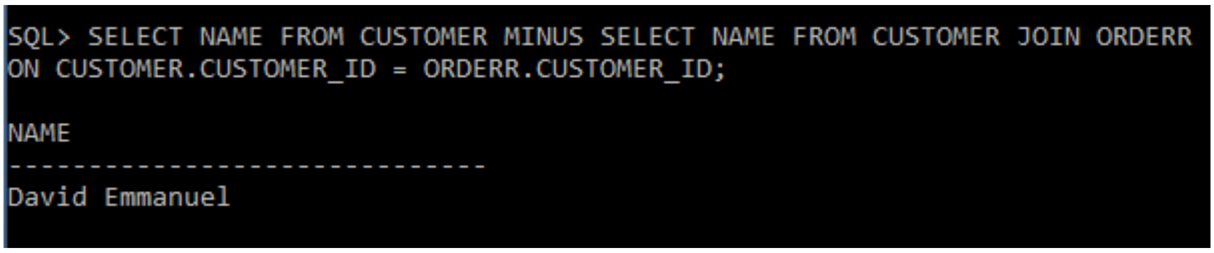
```
SQL> SELECT UPPER(NAME) FROM CUSTOMER;

UPPER(NAME)
------------------------------
AWANI KENDURKAR
VISHAL DAS
```

3. Select DELIVERY_DATE from ORDER in the format ‘30th July 2019’.



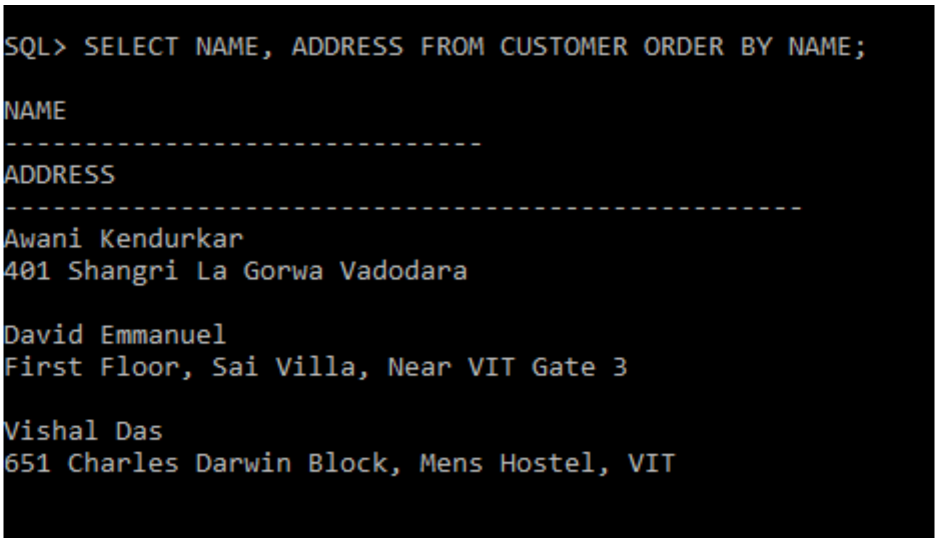
```
SQL> SELECT TO_CHAR(DELIVERY_DATE,'DDth Month YYYY') FROM ORDERR;

TO_CHAR(DELIVERY_DA
-------------------
30TH June      2019
30TH July      2019
```

4. Select the name of customer(s) who have not yet placed any order.



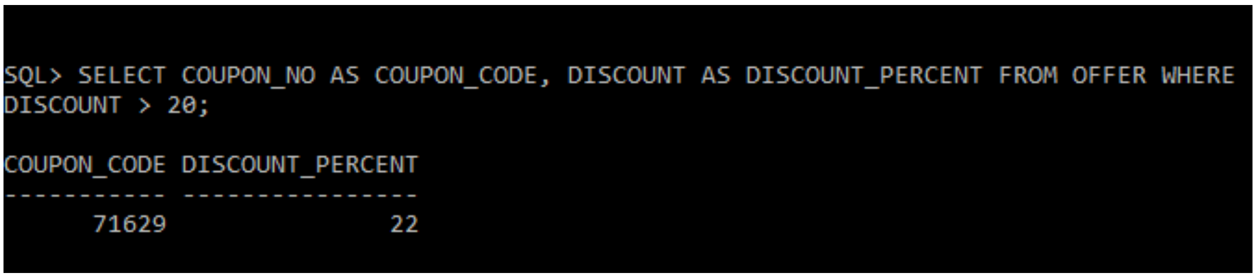
```
SQL> SELECT NAME FROM CUSTOMER MINUS SELECT NAME FROM CUSTOMER JOIN ORDERR
ON CUSTOMER.CUSTOMER_ID = ORDERR.CUSTOMER_ID;

NAME
------------------------------
David Emmanuel
```

5. Select names and addresses of customers in alphabetical order of their names.



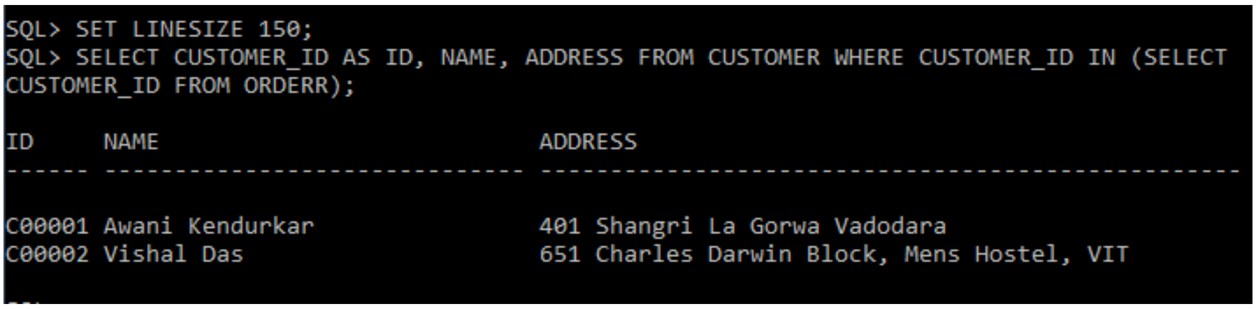
```
SQL> SELECT NAME, ADDRESS FROM CUSTOMER ORDER BY NAME;

NAME
------------------------------
ADDRESS
--------------------------------------------------
Awani Kendurkar
401 Shangri La Gorwa Vadodara

David Emmanuel
First Floor, Sai Villa, Near VIT Gate 3

Vishal Das
651 Charles Darwin Block, Mens Hostel, VIT
```

6. Select COUPON_NO and DISCOUNT from OFFER where the discount offered is greater than 20%.



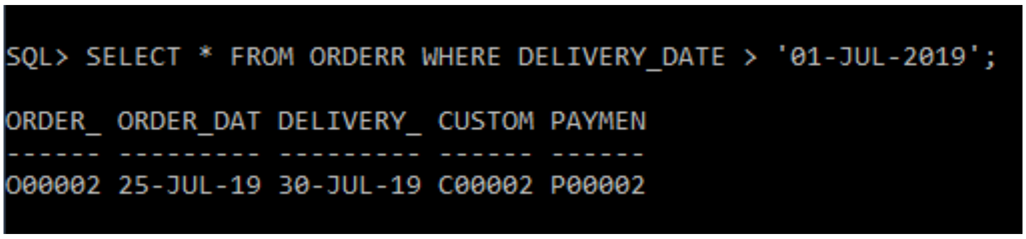
```
SQL> SELECT COUPON_NO AS COUPON_CODE, DISCOUNT AS DISCOUNT_PERCENT FROM OFFER WHERE
DISCOUNT > 20;

COUPON_CODE DISCOUNT_PERCENT
----------- ----------------
      71629               22
```

7. Select the ID, name and address of the customer who have placed an order.



```
SQL> SET LINESIZE 150;
SQL> SELECT CUSTOMER_ID AS ID, NAME, ADDRESS FROM CUSTOMER WHERE CUSTOMER_ID IN (SELECT
CUSTOMER_ID FROM ORDERR);

ID     NAME                           ADDRESS
------ ------------------------------ --------------------------------------------------

C00001 Awani Kendurkar                401 Shangri La Gorwa Vadodara
C00002 Vishal Das                     651 Charles Darwin Block, Mens Hostel, VIT
```

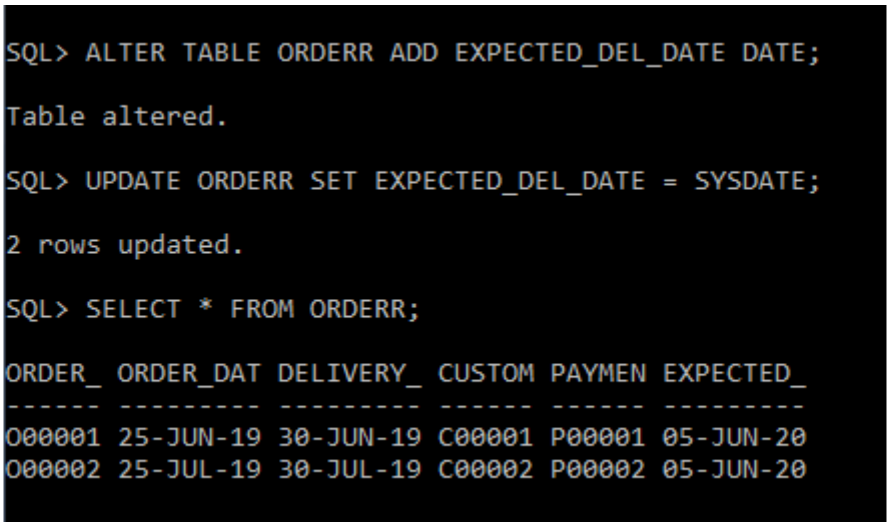
8. Select details of the order from ORDERR where DELIVERY_DATE is after 1st July 2019.



```
SQL> SELECT * FROM ORDERR WHERE DELIVERY_DATE > '01-JUL-2019';

ORDER_ ORDER_DAT DELIVERY_ CUSTOM PAYMEN
------ --------- --------- ------ ------
O00002 25-JUL-19 30-JUL-19 C00002 P00002
```

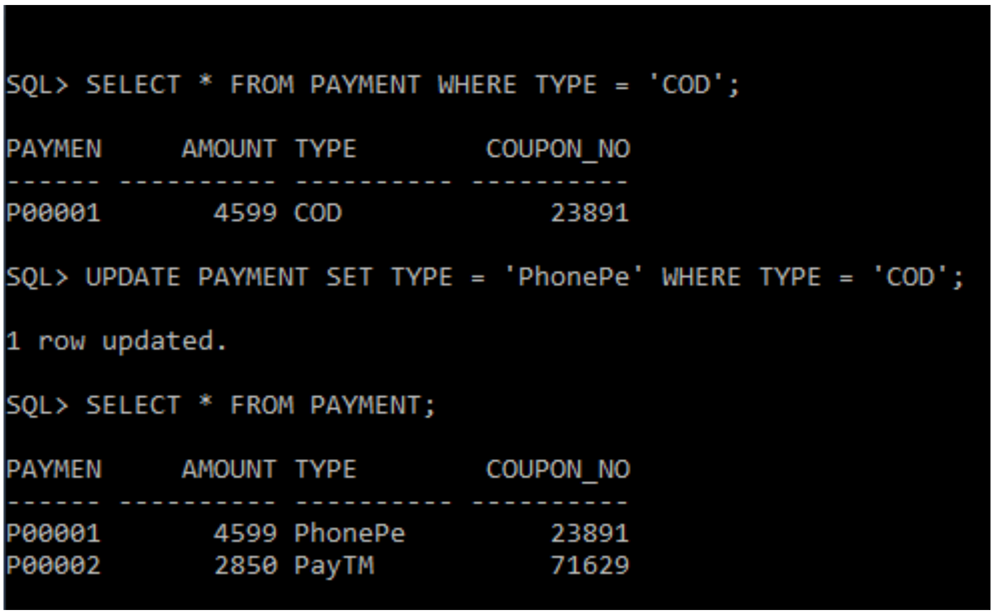
Data modification:

1. Since the lockdown has been imposed in the country due to COVID-19, the delivery date of the orders has been postponed. Add a new column EXPECTED_DEL_DATE to the table ORDERR.



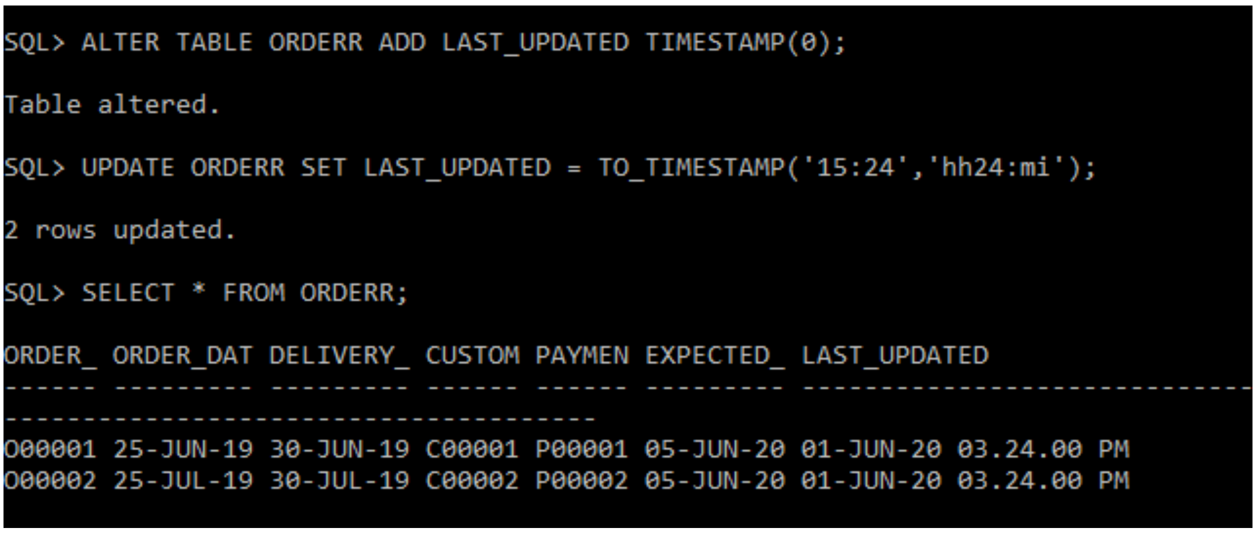
```
SQL> ALTER TABLE ORDERR ADD EXPECTED_DEL_DATE DATE;

Table altered.

SQL> UPDATE ORDERR SET EXPECTED_DEL_DATE = SYSDATE;

2 rows updated.

SQL> SELECT * FROM ORDERR;

ORDER_ ORDER_DAT DELIVERY_ CUSTOM PAYMEN EXPECTED_
------ --------- --------- ------ ------ ---------
O00001 25-JUN-19 30-JUN-19 C00001 P00001 05-JUN-20
O00002 25-JUL-19 30-JUL-19 C00002 P00002 05-JUN-20
```

2. Due to the spread of COVID-19, cash on delivery is no longer accepted as a payment type. If there is a TYPE in PAYMENT which is ‘COD’, update it to ‘PhonePe’.



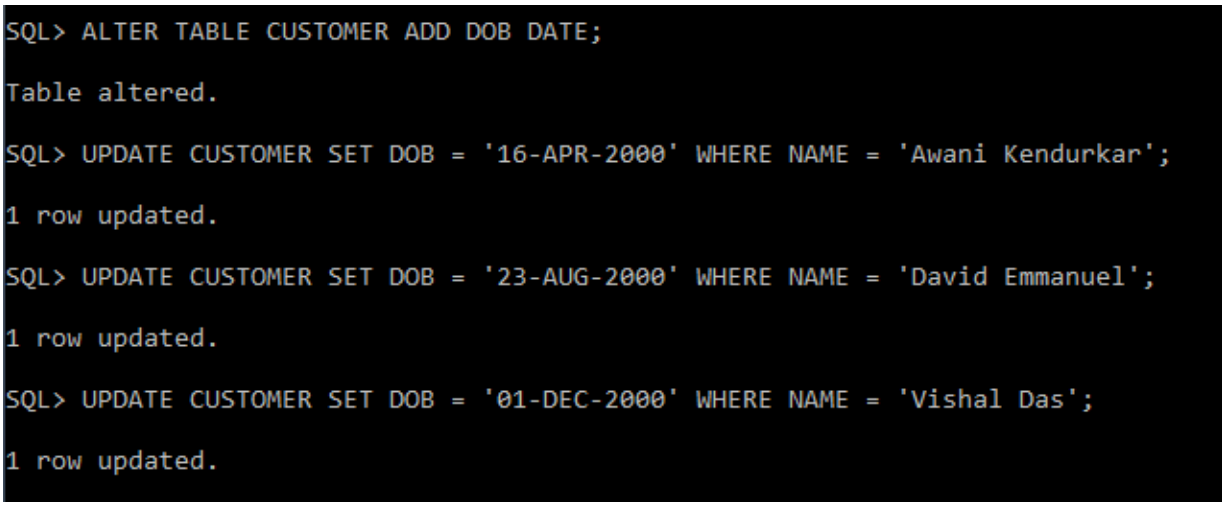
```
SQL> SELECT * FROM PAYMENT WHERE TYPE = 'COD';

PAYMEN     AMOUNT TYPE        COUPON_NO
------ ---------- ---------- ----------
P00001       4599 COD             23891

SQL> UPDATE PAYMENT SET TYPE = 'PhonePe' WHERE TYPE = 'COD';

1 row updated.

SQL> SELECT * FROM PAYMENT;

PAYMEN     AMOUNT TYPE        COUPON_NO
------ ---------- ---------- ----------
P00001       4599 PhonePe         23891
P00002       2850 PayTM           71629
```

3. Add a new column to ORDERR, LAST_UPDATED which shows the time at which the order was last updated/tracked.

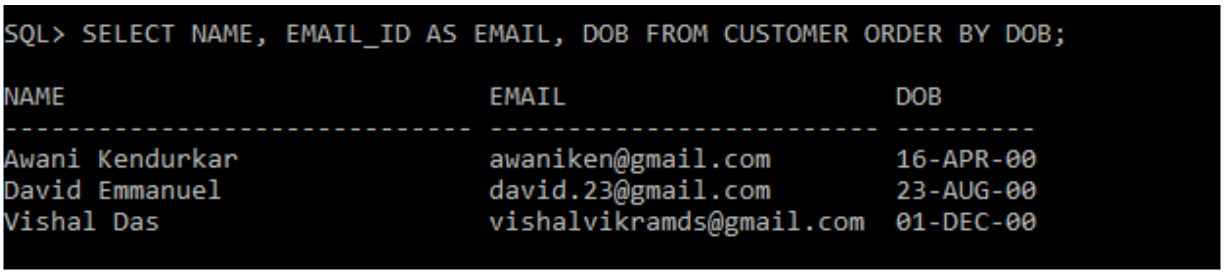


```
SQL> ALTER TABLE ORDERR ADD LAST_UPDATED TIMESTAMP(0);

Table altered.

SQL> UPDATE ORDERR SET LAST_UPDATED = TO_TIMESTAMP('15:24','hh24:mi');

2 rows updated.

SQL> SELECT * FROM ORDERR;

ORDER_ ORDER_DAT DELIVERY_ CUSTOM PAYMEN EXPECTED_ LAST_UPDATED
------ --------- --------- ------ ------ --------- -----------------------------
-----------------------------------
O00001 25-JUN-19 30-JUN-19 C00001 P00001 05-JUN-20 01-JUN-20 03.24.00 PM
O00002 25-JUL-19 30-JUL-19 C00002 P00002 05-JUN-20 01-JUN-20 03.24.00 PM
```

4. Add column DOB to CUSTOMER.



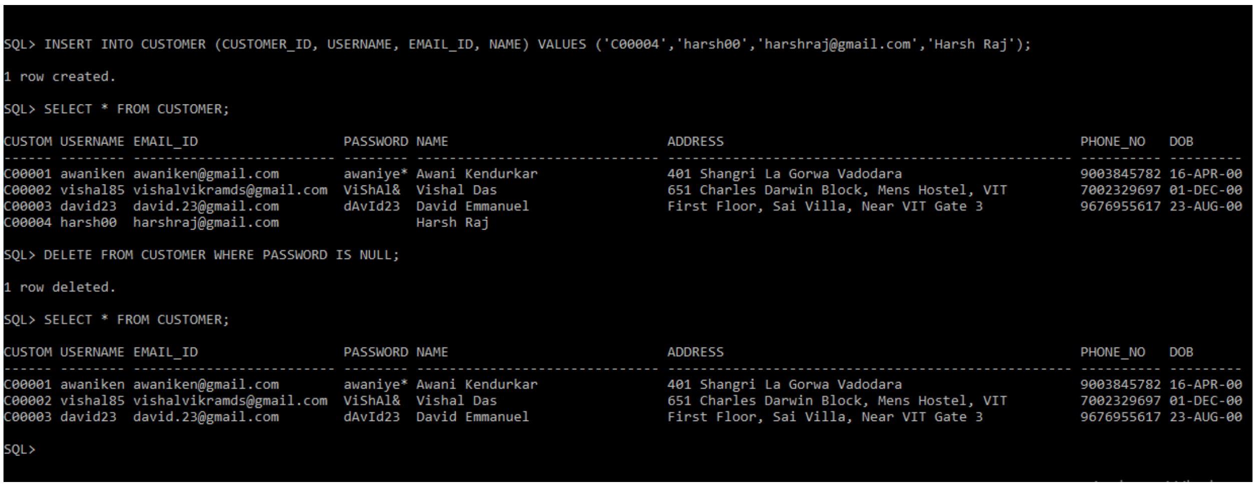
```
SQL> ALTER TABLE CUSTOMER ADD DOB DATE;

Table altered.

SQL> UPDATE CUSTOMER SET DOB = '16-APR-2000' WHERE NAME = 'Awani Kendurkar';

1 row updated.

SQL> UPDATE CUSTOMER SET DOB = '23-AUG-2000' WHERE NAME = 'David Emmanuel';

1 row updated.

SQL> UPDATE CUSTOMER SET DOB = '01-DEC-2000' WHERE NAME = 'Vishal Das';

1 row updated.
```



```
SQL> SELECT NAME, EMAIL_ID AS EMAIL, DOB FROM CUSTOMER ORDER BY DOB;

NAME                           EMAIL                     DOB
------------------------------ ------------------------- ---------
Awani Kendurkar                awaniken@gmail.com        16-APR-00
David Emmanuel                 david.23@gmail.com        23-AUG-00
Vishal Das                     vishalvikramds@gmail.com  01-DEC-00
```

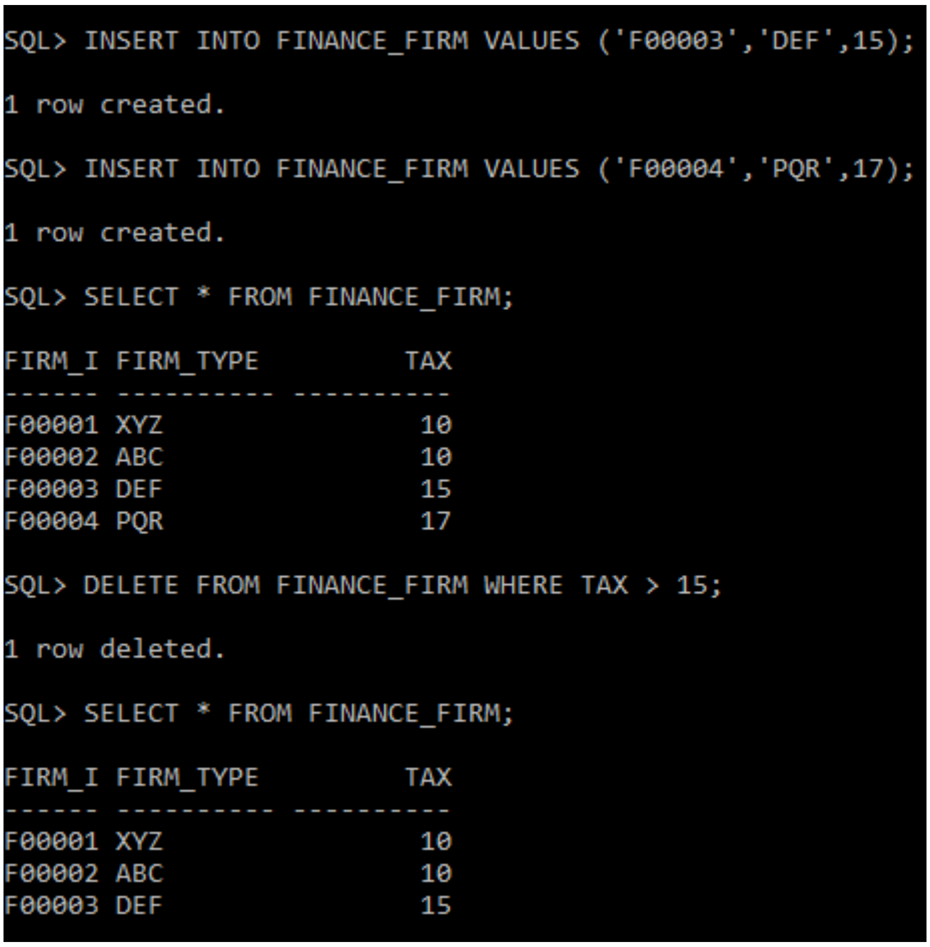
Data removal:

1. Delete customer entries that does not have a password.



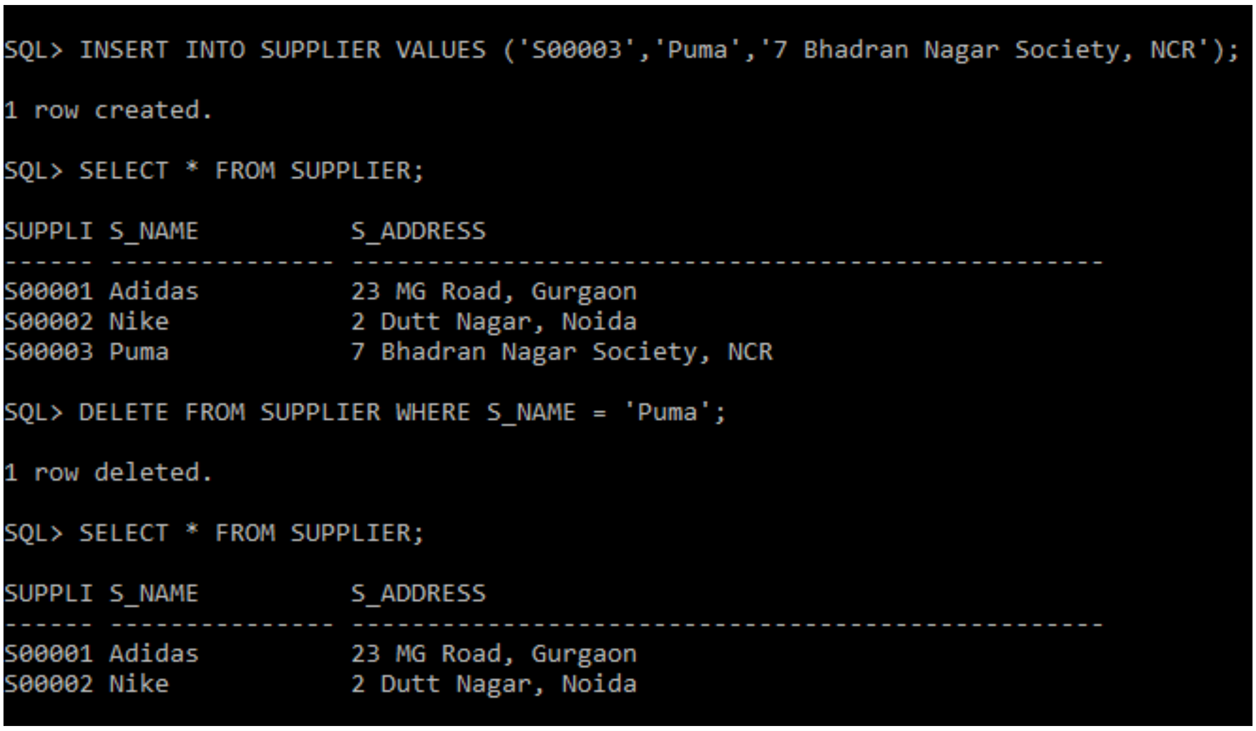
```
SQL> INSERT INTO CUSTOMER (CUSTOMER_ID, USERNAME, EMAIL_ID, NAME) VALUES ('C00004','harsh00','harshraj@gmail.com','Harsh Raj');

1 row created.

SQL> SELECT * FROM CUSTOMER;

CUSTOM USERNAME EMAIL_ID                  PASSWORD NAME                           ADDRESS                                            PHONE_NO   DOB
------ -------- ------------------------- -------- ------------------------------ -------------------------------------------------- ---------- ---------
C00001 awaniken awaniken@gmail.com        awaniye* Awani Kendurkar                401 Shangri La Gorwa Vadodara                      9003845782 16-APR-00
C00002 vishal85 vishalvikramds@gmail.com  ViShAl&  Vishal Das                     651 Charles Darwin Block, Mens Hostel, VIT         7002329697 01-DEC-00
C00003 david23  david.23@gmail.com        dAvId23  David Emmanuel                 First Floor, Sai Villa, Near VIT Gate 3            9676955617 23-AUG-00
C00004 harsh00  harshraj@gmail.com                 Harsh Raj

SQL> DELETE FROM CUSTOMER WHERE PASSWORD IS NULL;

1 row deleted.

SQL> SELECT * FROM CUSTOMER;

CUSTOM USERNAME EMAIL_ID                  PASSWORD NAME                           ADDRESS                                            PHONE_NO   DOB
------ -------- ------------------------- -------- ------------------------------ -------------------------------------------------- ---------- ---------
C00001 awaniken awaniken@gmail.com        awaniye* Awani Kendurkar                401 Shangri La Gorwa Vadodara                      9003845782 16-APR-00
C00002 vishal85 vishalvikramds@gmail.com  ViShAl&  Vishal Das                     651 Charles Darwin Block, Mens Hostel, VIT         7002329697 01-DEC-00
C00003 david23  david.23@gmail.com        dAvId23  David Emmanuel                 First Floor, Sai Villa, Near VIT Gate 3            9676955617 23-AUG-00

SQL>
```

2. Delete finance firms where tax is greater than 15.



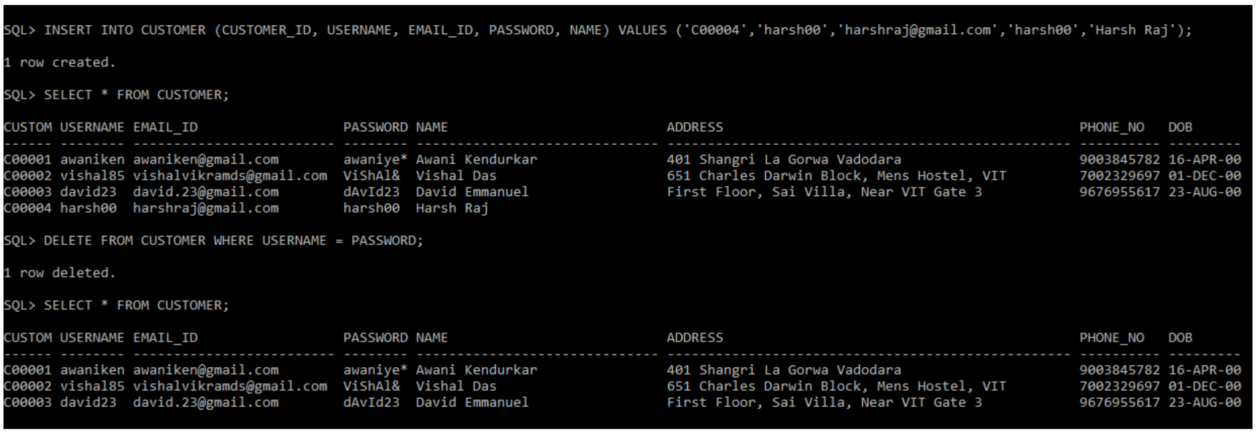
```
SQL> INSERT INTO FINANCE_FIRM VALUES ('F00003','DEF',15);

1 row created.

SQL> INSERT INTO FINANCE_FIRM VALUES ('F00004','PQR',17);

1 row created.

SQL> SELECT * FROM FINANCE_FIRM;

FIRM_I FIRM_TYPE         TAX
------ ---------- ----------
F00001 XYZ                10
F00002 ABC                10
F00003 DEF                15
F00004 PQR                17

SQL> DELETE FROM FINANCE_FIRM WHERE TAX > 15;

1 row deleted.

SQL> SELECT * FROM FINANCE_FIRM;

FIRM_I FIRM_TYPE         TAX
------ ---------- ----------
F00001 XYZ                10
F00002 ABC                10
F00003 DEF                15
```

3. Delete Puma supplier.



```
SQL> INSERT INTO SUPPLIER VALUES ('S00003','Puma','7 Bhadran Nagar Society, NCR');

1 row created.

SQL> SELECT * FROM SUPPLIER;

SUPPLI S_NAME          S_ADDRESS
------ --------------- --------------------------------------------------
S00001 Adidas          23 MG Road, Gurgaon
S00002 Nike            2 Dutt Nagar, Noida
S00003 Puma            7 Bhadran Nagar Society, NCR

SQL> DELETE FROM SUPPLIER WHERE S_NAME = 'Puma';

1 row deleted.

SQL> SELECT * FROM SUPPLIER;

SUPPLI S_NAME          S_ADDRESS
------ --------------- --------------------------------------------------
S00001 Adidas          23 MG Road, Gurgaon
S00002 Nike            2 Dutt Nagar, Noida
```

4. Delete customer whose password is same as username.



```
SQL> INSERT INTO CUSTOMER (CUSTOMER_ID, USERNAME, EMAIL_ID, PASSWORD, NAME) VALUES ('C00004','harsh00','harshraj@gmail.com','harsh00','Harsh Raj');

1 row created.

SQL> SELECT * FROM CUSTOMER;

CUSTOM USERNAME EMAIL_ID                  PASSWORD NAME                           ADDRESS                                            PHONE_NO   DOB
------ -------- ------------------------- -------- ------------------------------ -------------------------------------------------- ---------- ---------
C00001 awaniken awaniken@gmail.com        awaniye* Awani Kendurkar                401 Shangri La Gorwa Vadodara                      9003845782 16-APR-00
C00002 vishal85 vishalvikramds@gmail.com  ViShAl&  Vishal Das                     651 Charles Darwin Block, Mens Hostel, VIT         7002329697 01-DEC-00
C00003 david23  david.23@gmail.com        dAvId23  David Emmanuel                 First Floor, Sai Villa, Near VIT Gate 3            9676955617 23-AUG-00
C00004 harsh00  harshraj@gmail.com        harsh00  Harsh Raj

SQL> DELETE FROM CUSTOMER WHERE USERNAME = PASSWORD;

1 row deleted.

SQL> SELECT * FROM CUSTOMER;

CUSTOM USERNAME EMAIL_ID                  PASSWORD NAME                           ADDRESS                                            PHONE_NO   DOB
------ -------- ------------------------- -------- ------------------------------ -------------------------------------------------- ---------- ---------
C00001 awaniken awaniken@gmail.com        awaniye* Awani Kendurkar                401 Shangri La Gorwa Vadodara                      9003845782 16-APR-00
C00002 vishal85 vishalvikramds@gmail.com  ViShAl&  Vishal Das                     651 Charles Darwin Block, Mens Hostel, VIT         7002329697 01-DEC-00
C00003 david23  david.23@gmail.com        dAvId23  David Emmanuel                 First Floor, Sai Villa, Near VIT Gate 3            9676955617 23-AUG-00
```

PL/SQL

Functions:

1. Function to return name of product whose ID is passed

```
CREATE OR REPLACE FUNCTION PRODUCT_NAME(ID VARCHAR)
  RETURN VARCHAR IS NAME PRODUCT.P_NAME%TYPE;
  BEGIN
    SELECT P_NAME INTO NAME FROM PRODUCT WHERE PRODUCT_ID=ID;
  END;
  RETURN NAME;
END PRODUCT_NAME;
```

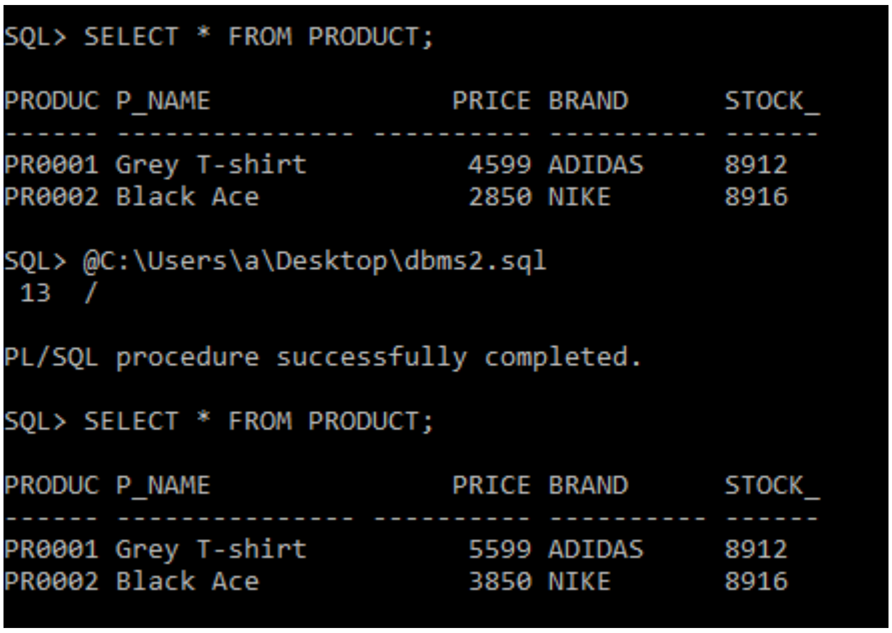
2. Function to find age of a customer by providing date of birth

```
CREATE OR REPLACE FUNCTION FIND_AGE (C_DATE DATE)
RETURN NUMBER AS V_AGE NUMBER;
CURSOR C1 IS SELECT AGE FROM CUSTOMER WHERE DOB=C_DATE;
BEGIN
  OPEN C1;
  IF C1%FOUND THEN
    FETCH C1 INTO V_AGE;
  END IF;
RETURN V_AGE;
END FIND_AGE;
```

Procedures:

1. To increase PRICE in the PRODUCT table by 1000.

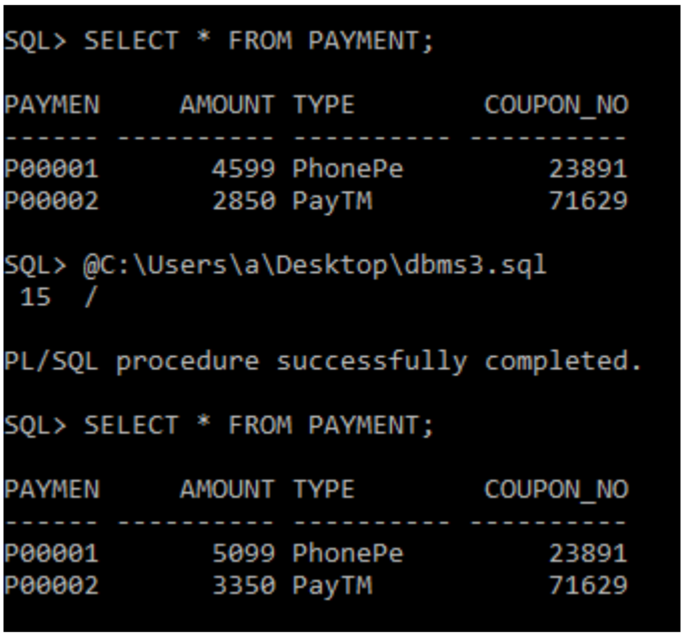
```
CREATE OR REPLACE PROCEDURE INC_PRICE
DECLARE
  total_rows number(2);
BEGIN
    UPDATE product
    SET price = price + 1000;
    IF sql%found THEN
      total_rows := sql%rowcount;
    END IF;
END;
```



```
SQL> SELECT * FROM PRODUCT;

PRODUC P_NAME               PRICE BRAND      STOCK_
------ -------------- ---------- ---------- ------
PR0001 Grey T-shirt          4599 ADIDAS     8912
PR0002 Black Ace             2850 NIKE       8916

SQL> @C:\Users\a\Desktop\dbms2.sql
 13  /

PL/SQL procedure successfully completed.

SQL> SELECT * FROM PRODUCT;

PRODUC P_NAME               PRICE BRAND      STOCK_
------ -------------- ---------- ---------- ------
PR0001 Grey T-shirt          5599 ADIDAS     8912
PR0002 Black Ace             3850 NIKE       8916
```

2. To increase AMOUNT in the PAYMENT table by 500.

```
CREATE OR REPLACE PROCEDURE INC_AMOUNT
DECLARE
  total_rows number(2);
BEGIN
    UPDATE payment
    SET amount = amount + 500;
    IF sql%found THEN
      total_rows := sql%rowcount;
    END IF;
END;
```



```
SQL> SELECT * FROM PAYMENT;

PAYMEN     AMOUNT TYPE       COUPON_NO
------ ---------- ---------- ----------
P00001       4599 PhonePe         23891
P00002       2850 PayTM           71629

SQL> @C:\Users\a\Desktop\dbms3.sql
 15  /

PL/SQL procedure successfully completed.

SQL> SELECT * FROM PAYMENT;

PAYMEN     AMOUNT TYPE       COUPON_NO
------ ---------- ---------- ----------
P00001       5099 PhonePe         23891
P00002       3350 PayTM           71629
```