Teacher’s Assessment 05

Aim:

Consider the SimpleCo databases relation schema comprising of database tables – INVOICE,CUSTOMER and CUSTOMER2. The primary keys are highlighted in red color.

You may apply your domain understanding to assign appropriate constraints additionally.

Create the tables and populate as mentioned.

Problem Statement:

```
-----------------------------------------------------------------------------------
             Author       : Pravesh Dholwani
             Semester     : 5 CSE
             Roll No.     : 108
             Date         : 15-OCT-21
-----------------------------------------------------------------------------------
```

Queries &Outcomes:

```
==============================================
*********************************************************************************
Table Creation & Population
*********************************************************************************
CREATE TABLE CUSTOMER(
      CUST_NUM NUMBER(4) ,
      CUST_LNAME VARCHAR2(20) NOT NULL,
      CUST_FNAME VARCHAR2(20) NOT NULL,
      CUST_BALANCE NUMBER(10,2),
      PRIMARY KEY(CUST_NUM)
);

Table created.

CREATE TABLE INVOICE(
      INV_NUM NUMBER(4),
      CUST_NUM NUMBER(4) REFERENCES CUSTOMER(CUST_NUM),
      INV_DATE DATE,
      INV_AMOUNT NUMBER(10,2) NOT NULL,
      PRIMARY KEY(INV_NUM,CUST_NUM)
);

Table created.
```

```
CREATE TABLE CUSTOMER2(
       CUST_NUM NUMBER(4),
       CUST_LNAME VARCHAR2(20),
       CUST_FNAME VARCHAR2(20),
       PRIMARY KEY(CUST_NUM)
);

Table created.

INSERT INTO CUSTOMER(CUST_NUM,CUST_LNAME,CUST_FNAME,CUST_BALANCE)
       VALUES(1000,'Sharma','Suman',2200.65);

1 row created.

INSERT INTO CUSTOMER(CUST_NUM,CUST_LNAME,CUST_FNAME,CUST_BALANCE)
       VALUES(1001,'Mantha','Mohan',1440.55);

1 row created.

INSERT INTO INVOICE(INV_NUM,CUST_NUM,INV_DATE,INV_AMOUNT)
       VALUES(8000,1000,'24-AUG-21',765.92);

1 row created.

INSERT INTO INVOICE(INV_NUM,CUST_NUM,INV_DATE,INV_AMOUNT)
       VALUES(8001,1001,'28-AUG-21',456.30);

1 row created.

INSERT INTO INVOICE(INV_NUM,CUST_NUM,INV_DATE,INV_AMOUNT)
       VALUES(8002,1001,'30-AUG-21',984.25);

1 row created.

INSERT INTO INVOICE(INV_NUM,CUST_NUM,INV_DATE,INV_AMOUNT)
       VALUES(8003,1000,'12-SEP-21',891.23);

1 row created.

INSERT INTO INVOICE(INV_NUM,CUST_NUM,INV_DATE,INV_AMOUNT)
       VALUES(8004,1000,'15-SEP-21',543.50);
```

```
1 row created.

INSERT INTO CUSTOMER2(CUST_NUM,CUST_LNAME,CUST_FNAME)
      VALUES(2000,'Mantha','Mohan');

1 row created.

INSERT INTO CUSTOMER2(CUST_NUM,CUST_LNAME,CUST_FNAME)
      VALUES(2001,'Bambani','Basav');

1 row created.

INSERT INTO CUSTOMER2(CUST_NUM,CUST_LNAME,CUST_FNAME)
      VALUES(2002,'Kaswa','Ketki');

1 row created.

INSERT INTO CUSTOMER2(CUST_NUM,CUST_LNAME,CUST_FNAME)
      VALUES(2003,'Chawla','Chanchal');

1 row created.
```

****

Query 1: Write the query that will generate a combined list of customers (from the tables CUSTOMER and CUSTOMER_2) that do not include the duplicate customer records.

****

QUERY:

```
      SELECT CUST_LNAME,CUST_FNAME
      FROM CUSTOMER
            UNION
      SELECT CUST_LNAME,CUST_FNAME
       FROM CUSTOMER2;
```

OUTPUT:

```
      CUST_LNAME           CUST_FNAME
      -------------------- --------------------
      Bambani              Basav
      Chawla               Chanchal
      Kaswa                Ketki
      Mantha               Mohan
      Sharma               Suman
```

****

Query 2: Write the query that will generate a combined list of customers to include the duplicate customer records. Also write the query that will show only the duplicate customer records.

****

QUERY:

```
      SELECT CUST_LNAME,CUST_FNAME
      FROM CUSTOMER
            UNION ALL
      SELECT CUST_LNAME,CUST_FNAME
       FROM CUSTOMER2;
```

OUTPUT:

```
      CUST_LNAME           CUST_FNAME
      -------------------- --------------------
      Sharma               Suman
      Mantha               Mohan
      Mantha               Mohan
      Bambani              Basav
      Kaswa                Ketki
      Chawla               Chanchal
```

Query:

```
      SELECT CUST_LNAME,CUST_FNAME
      FROM CUSTOMER
            INTERSECT
      SELECT CUST_LNAME,CUST_FNAME
       FROM CUSTOMER2;
```

OUTPUT:

```
      CUST_LNAME           CUST_FNAME
      -------------------- --------------------
      Mantha               Mohan
```

****

Query 3: Write the query that will generate only the records that are unique to the CUSTOMER_2 table.

****

QUERY:

```
      SELECT DISTINCT CUST_NUM,CUST_FNAME,CUST_LNAME
            FROM CUSTOMER2;
```

OUTPUT:

```
        CUST_NUM CUST_FNAME           CUST_LNAME
```

```
      ---------- ------------------- -------------------
            2000 Mohan               Mantha
            2002 Ketki               Kaswa
            2003 Chanchal            Chawla
            2001 Basav               Bambani
```

QUERY:

```
      SELECT INV_NUM,(SELECT (CUST_FNAME ||' '|| CUST_LNAME)
       FROM CUSTOMER WHERE CUST_NUM=I.CUST_NUM) AS NAME ,INV_DATE,INV_AMOUNT
            FROM INVOICE I;
```

OUTPUT:

```
   INV_NUM NAME            INV_DATE  INV_AMOUNT
---------- --------------- --------- ----------
      8000 Suman Sharma    24-AUG-21     765.92
      8001 Mohan Mantha    28-AUG-21      456.3
      8002 Mohan Mantha    30-AUG-21     984.25
      8003 Suman Sharma    12-SEP-21     891.23
      8004 Suman Sharma    15-SEP-21      543.5
```

****

Query 4: Write the query to show the invoice number, the customer number, the customer name, the invoice date, and the invoice amount for all customers with a customer balance of $1,500 or more.

****

QUERY:

```
      SELECT INV_NUM,(SELECT (CUST_FNAME ||' '|| CUST_LNAME)
       FROM CUSTOMER WHERE CUST_NUM=I.CUST_NUM) AS NAME ,INV_DATE,INV_AMOUNT
            FROM INVOICE I
            WHERE CUST_NUM IN (SELECT CUST_NUM FROM CUSTOMER
            WHERE CUST_BALANCE>=1500);
```

OUTPUT:

```
   INV_NUM NAME            INV_DATE  INV_AMOUNT
---------- --------------- --------- ----------
      8000 Suman Sharma    24-AUG-21     765.92
      8003 Suman Sharma    12-SEP-21     891.23
      8004 Suman Sharma    15-SEP-21      543.5
```

****

Query 5: Write the query that will show (for all the invoices) the invoice number, the invoice amount, the average invoice amount, and the difference between the average invoice amount and the actual invoice amount.

****

QUERY:

```
SELECT INV_NUM,INV_AMOUNT,
      (SELECT AVG(INV_AMOUNT) FROM INVOICE)AVEG,
      (INV_AMOUNT-(SELECT AVG(INV_AMOUNT) FROM INVOICE))DIFFER
      FROM INVOICE;
```

OUTPUT:

```
   INV_NUM INV_AMOUNT       AVEG     DIFFER
---------- ---------- ---------- ----------
      8003     891.23     728.24     162.99
      8001      456.3     728.24    -271.94
      8000     765.92     728.24      37.68
      8002     984.25     728.24     256.01
      8004      543.5     728.24    -184.74
```

****

Query 6: Write the query that will write Oracle sequences to produce automatic customer

number and invoice number values. Start the customer numbers at 1000 and the invoice

numbers at 5000.

****

QUERY:

```
CREATE SEQUENCE INV_SEQ
      START WITH 5000
      INCREMENT BY 1
      NOCACHE
      NOCYCLE;
```

OUTPUT:

```
Sequence created.
```

QUERY:

```
CREATE SEQUENCE CUST_SEQ
      START WITH 1000
      INCREMENT BY 1
      NOCACHE
      NOCYCLE;
```

OUTPUT:

```
Sequence created.
```

****

Query 7: Modify the CUSTOMER table to include two new attributes: CUST_DOB and CUST_AGE. Customer 1000 was born on March 02, 1980, and customer 1001 was born on January 30, 1979.

****

QUERY:

```
ALTER TABLE CUSTOMER ADD CUST_DOB DATE;
ALTER TABLE CUSTOMER ADD CUST_AGE NUMBER(3);
UPDATE CUSTOMER SET CUST_DOB='02-MAR-1980' WHERE CUST_NUM=1000;
UPDATE CUSTOMER SET CUST_DOB='30-JAN-1979' WHERE CUST_NUM=1001;
```

OUTPUT:

```
Table altered.
Table altered.
1 row updated.
1 row updated.
```

****

Query 8: Write the query that will list the names and ages of your customers (Precondition:

Query 7 executed).

****

QUERY:

```
SELECT (CUST_FNAME||' '||CUST_LNAME)NAME,((SYSDATE-CUST_DOB)/365)CUST_AGE
      FROM CUSTOMER;
```

OUTPUT:

```
NAME             CUST_AGE
--------------- ----------
Suman Sharma           42
Mohan Mantha           43
```

****

Query 9: Assuming the CUSTOMER table contains a CUST_AGE attribute, write the query to

update the values in that attribute. (Hint: Use the results of the previous query.)

****

QUERY:

```
UPDATE CUSTOMER SET CUST_AGE=((SYSDATE-CUST_DOB)/365);
```

OUTPUT:

```
      2 rows updated.
```

****

Query 10: Write the query that lists the average age of all customers. (Assume that the CUSTOMER table has been modified to include the CUST_DOB and the derived CUST_AGE attribute.)

****

QUERY:

```
      SELECT AVG(CUST_AGE)AVERAGE_CUSTOMER_AGE FROM CUSTOMER;
```

OUTPUT:

```
AVERAGE_CUSTOMER_AGE
--------------------
                42.5
```

****