COURSE TITLE: DATABASE SYSTEMS LAB

COURSE CODE: BCSE302P

EXPERIMENT NO: 7

NAME: Pratham Jethwani

REGISTER NO: 21BRS1593

SLOT: L21+L22

CLASS NUMBER: CH2022232501096

DATE: 15-APRIL, 2023

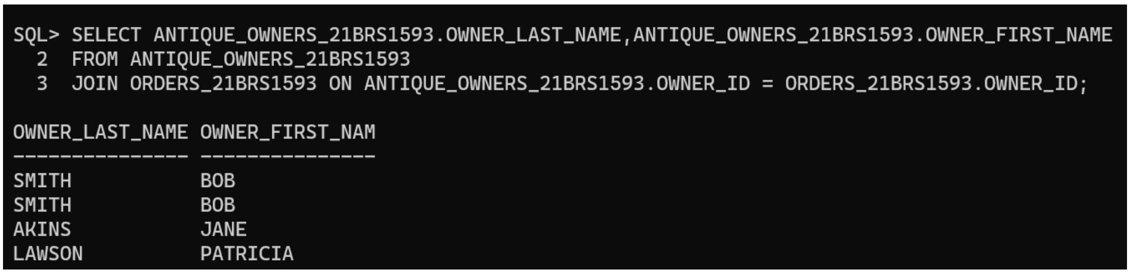
FACULTY NAME: L. JANI ANBARASI

1. Show each Antiques order and the last and first names of the person who ordered the item.

QUERY: -

SELECTANTIQUE_OWNERS_21BRS1593.OWNER_LAST_NAME, ANTIQUE_OWNERS_21BRS1593.OWNER_FIRST_NAME FROM ANTIQUE_OWNERS_21BRS1593 JOIN ORDERS_21BRS1593 ON ANTIQUE_OWNERS_21BRS1593.OWNER_ID = ORDERS_21BRS1593.OWNER_ID;

OUTPUT: -



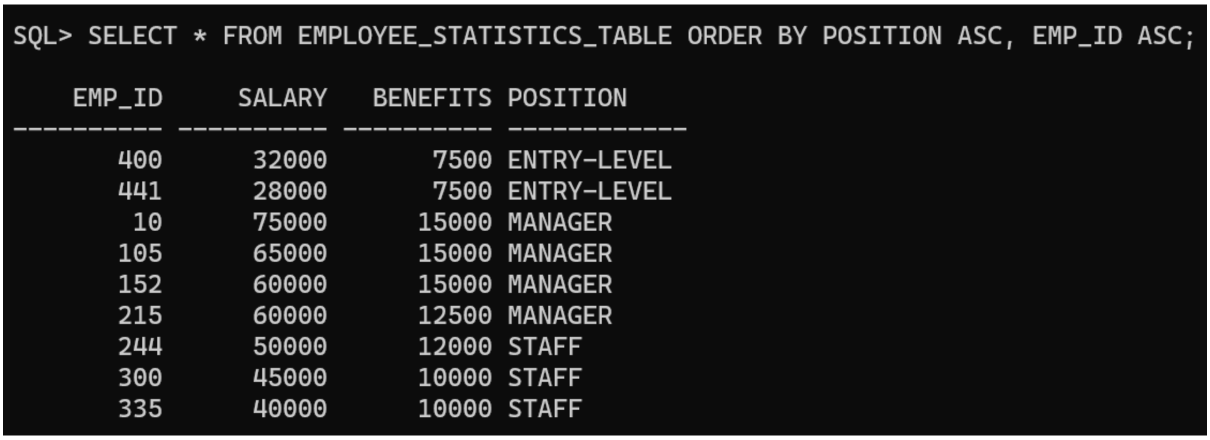
```
SQL> SELECT ANTIQUE_OWNERS_21BRS1593.OWNER_LAST_NAME,ANTIQUE_OWNERS_21BRS1593.OWNER_FIRST_NAME
  2  FROM ANTIQUE_OWNERS_21BRS1593
  3  JOIN ORDERS_21BRS1593 ON ANTIQUE_OWNERS_21BRS1593.OWNER_ID = ORDERS_21BRS1593.OWNER_ID;

OWNER_LAST_NAME OWNER_FIRST_NAM
--------------- ---------------
SMITH           BOB
SMITH           BOB
AKINS           JANE
LAWSON          PATRICIA
```

2. Show each column in the Employee Statistics Table in alphabetical order by Position, then by EmployeeIDNo.

QUERY: -

SELECT * FROM EMPLOYEE_STATISTICS_TABLE ORDER BY POSITION ASC, EMP_ID ASC;

OUTPUT: -



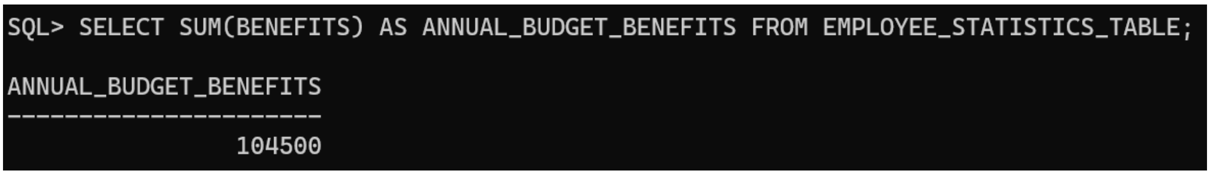
```
SQL> SELECT * FROM EMPLOYEE_STATISTICS_TABLE ORDER BY POSITION ASC, EMP_ID ASC;

    EMP_ID     SALARY   BENEFITS POSITION
---------- ---------- ---------- ------------
       400      32000       7500 ENTRY-LEVEL
       441      28000       7500 ENTRY-LEVEL
        10      75000      15000 MANAGER
       105      65000      15000 MANAGER
       152      60000      15000 MANAGER
       215      60000      12500 MANAGER
       244      50000      12000 STAFF
       300      45000      10000 STAFF
       335      40000      10000 STAFF
```

3. Show the annual budget for Benefits from the Employee Statistics Table.

QUERY: -

SELECT SUM(BENEFITS) AS ANNUAL_BUDGET_BENEFITS FROM EMPLOYEE_STATISTICS_TABLE;

OUTPUT: -



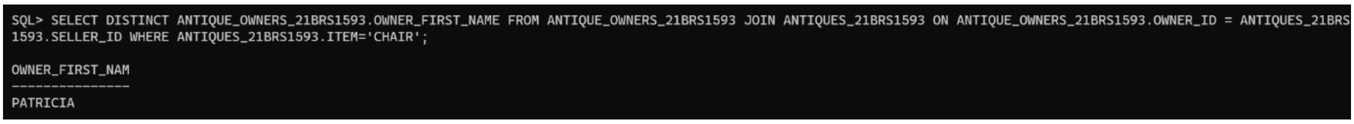
```
SQL> SELECT SUM(BENEFITS) AS ANNUAL_BUDGET_BENEFITS FROM EMPLOYEE_STATISTICS_TABLE;

ANNUAL_BUDGET_BENEFITS
----------------------
                104500
```

4. Using the IN Operator, show the names of the owners of Chairs.

QUERY: -

SELECT DISTINCT ANTIQUE_OWNERS_21BRS1593.OWNER_FIRST_NAME FROM ANTIQUE_OWNERS_21BRS1593 JOIN ANTIQUES_21BRS1593 ON ANTIQUE_OWNERS_21BRS1593.OWNER_ID = ANTIQUES_21BRS1593.SELLER_ID WHERE ANTIQUES_21BRS1593.ITEM='CHAIR';

OUTPUT: -



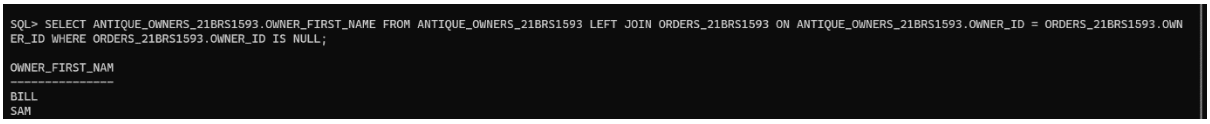
```
SQL> SELECT DISTINCT ANTIQUE_OWNERS_21BRS1593.OWNER_FIRST_NAME FROM ANTIQUE_OWNERS_21BRS1593 JOIN ANTIQUES_21BRS1593 ON ANTIQUE_OWNERS_21BRS1593.OWNER_ID = ANTIQUES_21BRS
1593.SELLER_ID WHERE ANTIQUES_21BRS1593.ITEM='CHAIR';

OWNER_FIRST_NAM
---------------
PATRICIA
```

5. Show the names of all Antiques Owners who have do not have an order placed.

QUERY: -

SELECT ANTIQUE_OWNERS_21BRS1593.OWNER_FIRST_NAME FROM ANTIQUE_OWNERS_21BRS1593 LEFT JOIN ORDERS_21BRS1593 ON ANTIQUE_OWNERS_21BRS1593.OWNER_ID = ORDERS_21BRS1593.OWNER_ID WHERE ORDERS_21BRS1593.OWNER_ID IS NULL;

OUTPUT: -



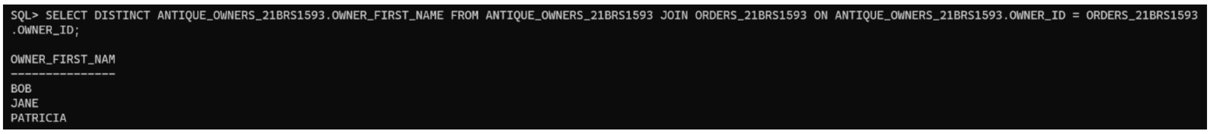
```
SQL> SELECT ANTIQUE_OWNERS_21BRS1593.OWNER_FIRST_NAME FROM ANTIQUE_OWNERS_21BRS1593 LEFT JOIN ORDERS_21BRS1593 ON ANTIQUE_OWNERS_21BRS1593.OWNER_ID = ORDERS_21BRS1593.OWN
ER_ID WHERE ORDERS_21BRS1593.OWNER_ID IS NULL;

OWNER_FIRST_NAM
---------------
BILL
SAM
```

6. Show the names of those who have placed Antique orders, with no duplicates (Hint: consider the order of tables in the From clauses)

QUERY: -

SELECT DISTINCT ANTIQUE_OWNERS_21BRS1593.OWNER_FIRST_NAME FROM ANTIQUE_OWNERS_21BRS1593 JOIN ORDERS_21BRS1593 ON ANTIQUE_OWNERS_21BRS1593.OWNER_ID = ORDERS_21BRS1593.OWNER_ID;

OUTPUT: -



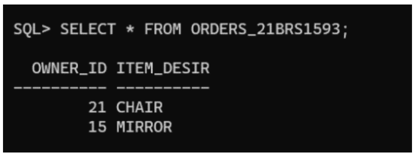
```
SQL> SELECT DISTINCT ANTIQUE_OWNERS_21BRS1593.OWNER_FIRST_NAME FROM ANTIQUE_OWNERS_21BRS1593 JOIN ORDERS_21BRS1593 ON ANTIQUE_OWNERS_21BRS1593.OWNER_ID = ORDERS_21BRS1593
.OWNER_ID;

OWNER_FIRST_NAM
---------------
BOB
JANE
PATRICIA
```

7. Delete all of Bob Smith's Antique orders

QUERY: -

DELETE FROM ORDERS_21BRS1593 WHERE OWNER_ID=02;

OUTPUT: -



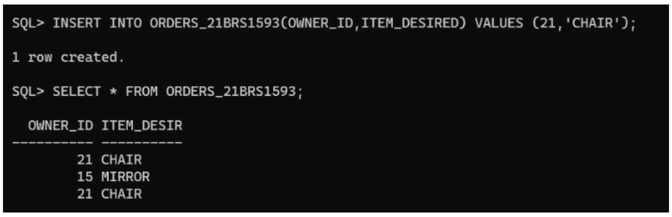
```
SQL> SELECT * FROM ORDERS_21BRS1593;

  OWNER_ID ITEM_DESIR
---------- ----------
        21 CHAIR
        15 MIRROR
```

8. Create an Antique order for a Rocking Chair for Jane Akins (Hint: Jane's ID Number is 21).

QUERY: -

INSERT INTO ORDERS_21BRS1593(OWNER_ID,ITEM_DESIRED) VALUES (21,'CHAIR');

OUTPUT: -



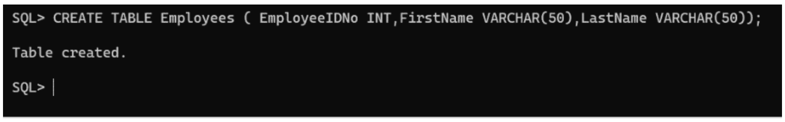
```
SQL> INSERT INTO ORDERS_21BRS1593(OWNER_ID,ITEM_DESIRED) VALUES (21,'CHAIR');

1 row created.

SQL> SELECT * FROM ORDERS_21BRS1593;

  OWNER_ID ITEM_DESIR
---------- ----------
        21 CHAIR
        15 MIRROR
        21 CHAIR
```

9. Create a table called Employees, with columns EmployeeIDNo (don't worry about trailing zeroes), FirstName, and LastName).

QUERY: -

CREATE TABLE Employees (EmployeeIDNo INT,FirstName VARCHAR(50),LastName VARCHAR(50));

OUTPUT: -



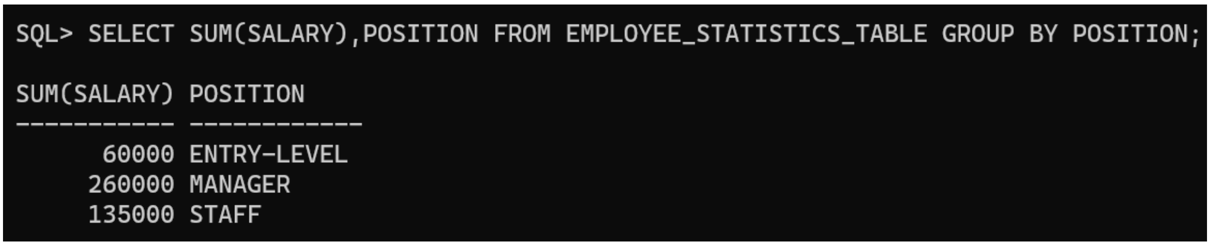
```
SQL> CREATE TABLE Employees ( EmployeeIDNo INT,FirstName VARCHAR(50),LastName VARCHAR(50));

Table created.

SQL> |
```

10. Show the annual budget for Salary by each position from the Employee Statistics Table.

QUERY: -

SELECT SUM(SALARY),POSITION FROM EMPLOYEE_STATISTICS_TABLE GROUP BY POSITION;

OUTPUT: -



```
SQL> SELECT SUM(SALARY),POSITION FROM EMPLOYEE_STATISTICS_TABLE GROUP BY POSITION;

SUM(SALARY) POSITION
----------- ------------
      60000 ENTRY-LEVEL
     260000 MANAGER
     135000 STAFF
```