

## Assignment – 11 Subqueries

- 1) Write a query that uses a subquery to obtain all orders for the customer named Cisneros. Assume you do not know his customer number (cnum).

```
SELECT * FROM ORDERS  
WHERE Cnum = (SELECT Cnum FROM CUSTOMERS WHERE Cname = 'Cisneros');
```

- 2) Write a query that produces the names and ratings of all customers who have above-average orders.

```
SELECT Cname, Rating  
FROM CUSTOMERS  
WHERE Cnum IN (SELECT Cnum FROM ORDERS GROUP BY Cnum HAVING SUM(Amt) > (SELECT  
AVG(Amt) FROM ORDERS));
```

- 3) Write a query that selects the total amount in orders for each salesperson for whom this total is greater than the amount of the largest order in the table.

```
SELECT Snum, SUM(Amt) AS TotalOrders  
FROM ORDERS  
GROUP BY Snum  
HAVING TotalOrders > (SELECT MAX(Amt) FROM ORDERS);
```

## Assignment – 12 Using the operators IN, ANY, and ALL.

1) Write a query that selects all customers whose ratings are equal to or greater than ANY of Serres.

```
SELECT * FROM CUSTOMERS  
WHERE Rating >= (SELECT Rating FROM SALESPeOPLE WHERE Sname = 'Serres');
```

2) Write a query using ANY or ALL that will find all salespeople who have no customers located in their city.

```
SELECT Sname  
FROM SALESPeOPLE  
WHERE Snum NOT IN (SELECT Snum FROM CUSTOMERS WHERE SALESPeOPLE.City =  
CUSTOMERS.City);
```

3) Write a query that selects all orders for amounts greater than any for the customers in London.

```
SELECT * FROM ORDERS  
WHERE Amt > ANY (SELECT Amt FROM ORDERS JOIN CUSTOMERS ON ORDERS.Cnum =  
CUSTOMERS.Cnum WHERE CUSTOMERS.City = 'London');
```

4) Write the above query using MIN or MAX.

```
SELECT * FROM ORDERS  
WHERE Amt > (SELECT MAX(Amt) FROM ORDERS JOIN CUSTOMERS ON ORDERS.Cnum =  
CUSTOMERS.Cnum WHERE CUSTOMERS.City = 'London');
```

## Assignment – 13 Using the UNION clause

- 1) Create a union of two queries that shows the names, cities, and ratings of all customers. Those with rating of 200 or greater will also have the words “High Rating”, while the others will have the words “Low Rating”.

```
SELECT Cname, City, Rating, 'High Rating' AS Status FROM CUSTOMERS WHERE Rating >= 200
```

```
UNION
```

```
SELECT Cname, City, Rating, 'Low Rating' AS Status FROM CUSTOMERS WHERE Rating < 200;
```

- 2) Write a command that produces the name and number of each salesperson and each customer with more than one current order. Put the results in alphabetical order.

```
SELECT Sname, Snum FROM SALESPeOPLE WHERE Snum IN (SELECT Snum FROM ORDERS GROUP BY Snum HAVING COUNT(*) > 1)
```

```
UNION
```

```
SELECT Cname, Cnum FROM CUSTOMERS WHERE Cnum IN (SELECT Cnum FROM ORDERS GROUP BY Cnum HAVING COUNT(*) > 1)
```

```
ORDER BY Sname, Cname;
```

- 3) Form a union of three queries. Have the first select the snums of all salespeople in San Jose; the second, the cnums of all customers in San Jose; and the third the onums of all orders on October 3. Retain duplicates between the last two queries but eliminate any redundancies between either of them and the first. (Note: in the sample tables as given, there would be no such redundancy. This is besides the point.)

```
SELECT Snum FROM SALESPeOPLE WHERE City = 'San Jose'
```

```
UNION
```

```
SELECT Cnum FROM CUSTOMERS WHERE City = 'San Jose'
```

```
UNION ALL
```

```
SELECT Onum FROM ORDERS WHERE Odate = '1990-10-03';
```

## Assignment – 14 Entering, Deleting, and Changing Field Values

- 1) Write a command that puts the following values, in their given order, into the salespeople table:  
city – San Jose, name – Blanco, comm – NULL, cnum – 1100.

```
INSERT INTO SALESPEOPLE (Snum, Sname, City, Comm)
VALUES (1100, 'Blanco', 'San Jose', NULL);
```

- 2) Write a command that removes all orders from customer Clemens from the Orders table.

```
DELETE FROM ORDERS
WHERE Cnum = (SELECT Cnum FROM CUSTOMERS WHERE Cname = 'Clemens');
```

- 3) Write a command that increases the rating of all customers in Rome by 100.

```
UPDATE CUSTOMERS
SET Rating = Rating + 100
WHERE City = 'Rome';
```

- 4) Salesperson Serres has left the company. Assign her customers to Motika.

```
UPDATE CUSTOMERS
SET Snum = (SELECT Snum FROM SALESPEOPLE WHERE Sname = 'Motika')
WHERE Snum = (SELECT Snum FROM SALESPEOPLE WHERE Sname = 'Serres');
```

## Assignment – 15 Using Subqueries with DML Commands.

- 1) Assume there is a table called Multicust, with all of the same column definitions as Salespeople. Write a command that inserts all salespeople with more than one customer into this table.

```
INSERT INTO Multicust (Snum, Sname, City, Comm)
SELECT Snum, Sname, City, Comm
FROM SALESPEOPLE
WHERE Snum IN (SELECT Snum FROM CUSTOMERS GROUP BY Snum HAVING COUNT(*) > 1);
```

- 2) Write a command that deletes all customers with no current orders.

```
DELETE FROM CUSTOMERS
WHERE Cnum NOT IN (SELECT Cnum FROM ORDERS);
```

- 3) Write a command that increases by twenty percent the commissions of all salespeople with total orders above Rs. 3,000.

```
UPDATE SALESPEOPLE
SET Comm = Comm * 1.20
WHERE Snum IN (SELECT Snum FROM ORDERS GROUP BY Snum HAVING SUM(Amt) > 3000);
```