

Course : CSE 2004 – Database Management Systems

Lab slot : L55+ L56

Faculty : Dr. B Saleena

Ex. No: 6

Date: 18/8/17

SQL - Commands revision and Views

Creating Table : Salesman

SQL> CREATE TABLE Salesman(Salesman_ID number, Salesman_name varchar(20), Salesman_city varchar(15), Salesman_comm number, CONSTRAINT SalesmanID_pk PRIMARY KEY(Salesman_ID));
Table created.

Creating Table : Customer

SQL> CREATE TABLE Customer(Customer_ID number, Customer_name varchar(20), Customer_city varchar(20), Customer_grade number, SalesmanID number, CONSTRAINT SalesmanID_fk FOREIGN KEY(SalesmanID) REFERENCES Salesman(Salesman_ID), CONSTRAINT CustomerID_pk PRIMARY KEY(Customer_ID));
Table created.

Creating Table : Orders

SQL> CREATE TABLE Orders(Order_num number, Purchase_amt number, Order_date date, CustomerID number, SalesmanID number, CONSTRAINT CustomerID_fk FOREIGN KEY(CustomerID) REFERENCES Customer(Customer_ID), CONSTRAINT OrderNo_pk PRIMARY KEY(Order_num));
Table created.

Table Salesman:

SALESMAN_ID	SALESMAN_NAME	SALESMAN_CITY	SALESMAN_COMM
-----	-----	-----	-----
1098	Ashok	Nashik	16
2789	Manoj	Noida	20
3428	Vishal	Amravati	11
4126	Anmol	Guwahati	14
5912	Varun	Ludhiana	10

Table Customer:

CUSTOMER_ID	CUSTOMER_NAME	CUSTOMER_CITY	CUSTOMER_GRADE	SALESMANID
-----	-----	-----	-----	-----
1190	Dhruv Garg	Mumbai	9	5912
1091	Aakash Tiwari	Delhi	7	3428
1382	Chahat Agarwal	Coimbatore	8	1098
1127	Arush Sharma	Dharamsala	7	4126
1143	Aditya Chitlangia	Jodhpur	7	2789
1199	Vishnu Nagpal	Delhi	6	5912
1331	Shivam Rathi	Pune	6	5912
1109	Shantanu Gupta	Bhopal	8	4126
1111	Swaraj Phadtare	Kolkata	6	1098
1081	Rachit Tiwary	Chennai	9	3428

Table Orders:

ORDER_NUM	PURCHASE_AMT	ORDER_DAT	CUSTOMERID	SALESMANID
-----	-----	-----	-----	-----
45281	1975	28-AUG-17	1190	5912
45282	742	28-AUG-17	1081	3428
45283	294	28-AUG-17	1091	3428
45271	2216	27-AUG-17	1382	1098
45272	660	27-AUG-17	1111	1098
45273	431	27-AUG-17	1109	4126
45261	800	26-AUG-17	1331	5912
45262	3999	26-AUG-17	1127	4126
45263	575	26-AUG-17	1143	2789
45251	11299	25-AUG-17	1199	5912

10 rows selected.

Revision queries

1. Write a query to display all the orders from the orders table issued by the salesman 'Manoj'

```
SQL> SELECT * FROM Orders WHERE SalesmanID = (SELECT Salesman_ID FROM Salesman WHERE
Salesman_name = 'Manoj');
```

ORDER_NUM	PURCHASE_AMT	ORDER_DAT	CUSTOMERID	SALESMANID
-----	-----	-----	-----	-----
45263	575	26-AUG-17	1143	2789

2. Write a query to display all the orders for the salesman who belongs to the city Amravati.

```
SQL> SELECT * FROM Orders WHERE SalesmanID = (SELECT Salesman_ID FROM Salesman WHERE
Salesman_city = 'Amravati');
```

ORDER_NUM	PURCHASE_AMT	ORDER_DAT	CUSTOMERID	SALESMANID
-----	-----	-----	-----	-----
45282	742	28-AUG-17	1081	3428
45283	294	28-AUG-17	1091	3428

3. Write a query to find all the orders issued against the salesman who works for customer whose id is 1382.

```
SQL> SELECT * FROM Orders WHERE SalesmanID = (SELECT SalesmanID FROM Customer WHERE
Customer_ID = 1382);
```

ORDER_NUM	PURCHASE_AMT	ORDER_DAT	CUSTOMERID	SALESMANID
-----	-----	-----	-----	-----
45271	2216	27-AUG-17	1382	1098
45272	660	27-AUG-17	1111	1098

4. Write a query to display all the orders which values are greater than the average order value for 28th August 2017.

```
SQL> SELECT * FROM Orders WHERE Purchase_amt > (SELECT AVG(Purchase_amt) FROM Orders
WHERE Order_date = '28-Aug-17');
```

ORDER_NUM	PURCHASE_AMT	ORDER_DAT	CUSTOMERID	SALESMANID
45281	1975	28-AUG-17	1190	5912
45271	2216	27-AUG-17	1382	1098
45262	3999	26-AUG-17	1127	4126
45251	11299	25-AUG-17	1199	5912

```
SQL> SELECT AVG(Purchase_amt) AS Avg_Purchase_28Aug FROM Orders WHERE Order_date = '28-
Aug-17';
```

AVG_PURCHASE_28AUG
1003.66667

5. Write a query to display the commission of all the salesmen servicing customers in 'Delhi'.

```
SQL> SELECT Salesman_ID, Salesman_name, Salesman_comm FROM Salesman WHERE
Salesman_ID IN (SELECT SalesmanID FROM Customer WHERE Customer_city = 'Delhi');
```

SALESMAN_ID	SALESMAN_NAME	SALESMAN_COMM
3428	Vishal	11
5912	Varun	10

6. Write a query to display all the customers whose id is below the salesman ID of 'Ashok'.

```
SQL> SELECT * FROM Customer WHERE Customer_ID < (SELECT Salesman_ID FROM Salesman
WHERE Salesman_name = 'Ashok');
```

CUSTOMER_ID	CUSTOMER_NAME	CUSTOMER_CITY	CUSTOMER_GRADE	SALESMANID
1081	Rachit Tiwary	Chennai	9	3428
1091	Aakash Tiwari	Delhi	7	3428

```
SQL> SELECT Salesman_ID FROM Salesman WHERE Salesman_name = 'Ashok';
```

SALESMAN_ID
1098

7. Write a query to counts the customers with grades above Delhi's average.

```
SQL> SELECT COUNT(*) AS Grade_greater_than_delhi FROM Customer WHERE Customer_grade >
(SELECT AVG(Customer_grade) FROM Customer WHERE Customer_city = 'Delhi');
```

GRADE_GREATER_THAN_DELHI
7

8. Write a query to display all customers with orders on August 26, 2017.

```
SQL> SELECT * FROM Customer WHERE Customer_ID IN (SELECT CustomerID FROM Orders WHERE
Order_date = '26-Aug-17');
```

CUSTOMER_ID	CUSTOMER_NAME	CUSTOMER_CITY	CUSTOMER_GRADE	SALESMANID
1331	Shivam Rathi	Pune	6	5912
1127	Arush Sharma	Dharamsala	7	4126
1143	Aditya Chitlangia	Jodhpur	7	2789

9. Write a query to display all the customers with orders issued on date 28th August, 2017.

```
SQL> SELECT * FROM Customer WHERE Customer_ID IN (SELECT CustomerID FROM Orders WHERE
Order_date = '28-Aug-17');
```

CUSTOMER_ID	CUSTOMER_NAME	CUSTOMER_CITY	CUSTOMER_GRADE	SALESMANID
1190	Dhruv Garg	Mumbai	9	5912
1081	Rachit Tiwary	Chennai	9	3428
1091	Aakash Tiwari	Delhi	7	3428

10. Write a query to find the name and numbers of all salesmen who had more than one customer.

```
SQL> SELECT Salesman_ID, Salesman_name FROM Salesman WHERE Salesman_ID IN (SELECT
SalesmanID FROM Orders HAVING COUNT(*) > 1 GROUP BY SalesmanID);
```

SALESMAN_ID	SALESMAN_NAME
1098	Ashok
4126	Anmol
5912	Varun
3428	Vishal

```
SQL> SELECT Salesman_ID, Salesman_name FROM Salesman WHERE Salesman_ID IN (SELECT
SalesmanID FROM Orders HAVING COUNT(*) > 2 GROUP BY SalesmanID);
```

SALESMAN_ID	SALESMAN_NAME
5912	Varun

- 11. Write a query to find the sums of the amounts from the orders table, grouped by date, eliminating all those dates where the sum was not greater than 6000.00 below the maximum amount.**

```
SQL> SELECT SUM(Purchase_amt) AS Total, Order_date FROM Orders HAVING SUM(Purchase_amt)
> (SELECT MAX(Purchase_amt)-6000 FROM Orders) GROUP BY Order_date;
```

TOTAL	ORDER_DAT
11299	25-AUG-17
5374	26-AUG-17

- 12. Write a query to find all orders with amount smaller than any amount for a customer in Pune.**

```
SQL> SELECT * FROM Orders WHERE Purchase_amt < (SELECT MIN(Purchase_amt) FROM Orders
WHERE CustomerID IN (SELECT Customer_ID FROM Customer WHERE Customer_city = 'Pune'));
```

ORDER_NUM	PURCHASE_AMT	ORDER_DAT	CUSTOMERID	SALESMANID
45282	742	28-AUG-17	1081	3428
45283	294	28-AUG-17	1091	3428
45272	660	27-AUG-17	1111	1098
45273	431	27-AUG-17	1109	4126
45263	575	26-AUG-17	1143	2789

Exercise for views

1. Write a query to create a view for all salesmen with columns salesman_id, salesman_name, and salesman_city.

```
SQL> CREATE VIEW Salesman_details AS
```

```
2 SELECT Salesman_ID, Salesman_name, Salesman_city
```

```
3 FROM Salesman;
```

View created.

```
SQL> SELECT * FROM Salesman_details;
```

SALESMAN_ID	SALESMAN_NAME	SALESMAN_CITY
-----	-----	-----
1098	Ashok	Nashik
2789	Manoj	Noida
3428	Vishal	Amravati
4126	Anmol	Guwahati
5912	Varun	Ludhiana

2. Write a query to find the salesmen of the city 'Noida' or 'Guwahati' who achieved the commission more than 15%.

```
SQL> CREATE VIEW Salesman_comm_cities AS
```

```
2 SELECT Salesman_ID, Salesman_name, Salesman_comm
```

```
3 FROM Salesman
```

```
4 WHERE (Salesman_city = 'Noida' OR Salesman_city = 'Guwahati') AND Salesman_comm > 15;
```

View created.

```
SQL> SELECT * FROM Salesman_comm_cities;
```

SALESMAN_ID	SALESMAN_NAME	SALESMAN_COMM
-----	-----	-----
2789	Manoj	20

3. Write a query to create a view to keeping track the number of customers ordering, number of salesmen attached, average amount of orders and the total amount of orders in a day.

```
SQL> CREATE VIEW Order_details AS
```

```
2 SELECT Order_date, COUNT(DISTINCT CustomerID) AS Num_of_cust, COUNT(DISTINCT
SalesmanID) AS Salesman_attatched, AVG(Purchase_amt) AS Avg_amt, SUM(Purchase_amt) AS
Total_amt
```

```
3 FROM Orders
```

```
4 GROUP BY Order_date;
```

View created.

```
SQL> SELECT * FROM Order_details;
```

ORDER_DAT	NUM_OF_CUST	SALESMAN_ATTATCHED	AVG_AMT	TOTAL_AMT
-----	-----	-----	-----	-----
25-AUG-17	1	1	11299	11299
26-AUG-17	3	3	1791.33333	5374
27-AUG-17	3	2	1102.33333	3307
28-AUG-17	3	2	1003.66667	3011

4. Write a query to create a view that finds the highest order on a day.

```
SQL> CREATE VIEW Max_amt_cust_salesman AS
```

```
2 SELECT Order_date, MAX(Purchase_amt) AS Max_amt
```

```
3 FROM Orders
```

```
4 GROUP BY Order_date;
```

View created.

```
SQL> SELECT * FROM Max_amt_cust_salesman;
```

ORDER_DAT	MAX_AMT
-----	-----
25-AUG-17	11299
26-AUG-17	3999
28-AUG-17	1975
27-AUG-17	2216

5. Write a query to create a view that finds the salesmen who issued orders on either August 17th, 2012 or October 10th, 2012.

```
SQL> CREATE VIEW Salesman_sales AS
  2 SELECT SalesmanID, Order_date
  3 FROM Orders
  4 WHERE Order_date = '25-Aug-17' OR Order_date = '28-Aug-17';
View created.
```

```
SQL> SELECT * FROM Salesman_sales;
```

SALESMANID	ORDER_DAT
-----	-----
5912	28-AUG-17
3428	28-AUG-17
3428	28-AUG-17
5912	25-AUG-17