**ASSIGNMENT 3**

Q1: **Create the following Databases.**

**TABLE NAME:Salesmen**

**SNUM SNAME CITY COMMISSION**

-------------------------------------------------------

1001 Piyush London 12 %

1002 Sejal Surat 13 %

1004 Miti London 11 %

1007 Rajesh Baroda 15 %

1003 Anand New Delhi 10 %

SNUM : A unique number assigned to each salesman.

SNAME : The name of salesman.

CITY : The location of salesmen.

COMMISSION: The Salemen's commission on orders.

TABLE NAME: Customers

CNUM CNAME CITY RATING SNUM

-------------------------------------------------------

2001 Harsh London 100 1001

2002 Gita Rome 200 1003

2003 Lalit Surat 200 1002

2004 Govind Bombay 300 1002

2006 Chirag London 100 1001

2008 Chinmay Surat 300 1007

2007 Pratik Rome 100 1004

CNUM : A unique number assigned to each customer.

CNAME : The name of the custom er.

CITY : The location of the customer.

RATING : A level of preference indicator given to this customer.

SNUM : The number of salesman assigned to this customer.

TABLE NAME:Orders

ONUM AMOUNT ODATE CNUM SNUM

-------------------------------------------------------

3001 18.69 10/03/97 2008 1007

3003 767.19 10/03/97 2001 1001

3005 5160.45 10/03/97 2003 1002

3006 1098.16 10/03/97 2008 1007

3009 1713.23 10/04/97 2002 1003

3007 75.75 10/04/97 2004 1002

3008 4723.00 10/05/97 2006 1001

3010 1309.95 10/06/97 2004 1002

3011 9891.88 10/06/97 2006 1001

ONUM : A unique number assigned to each order.

AMOUNT : The amount of an order.

ODATE : The date of an order. CNUM : The number of customer making the order.

SNUM : The number of salesman credited with the sale.

Solve the following queries using above databases and group by clause.

Q5: Solve the following queries using above databases

1. Show the name of all customers with their salesman's name.

**select c.cname ,(select sname from salesman s where c.snum=s.snum)"Salesman" from customers c;**

2. List all customers and salesmen who shared a same city.

**select c.cname,s.sname,c.city from customers c,salesman s where c.city=s.city;**

3. List all orders with the names of their customer and salesman.

**select o.\*,(select sname from salesman s where o.snum=s.snum)"Salesman",(select cname from customers c where o.cnum=c.cnum) "Customers" from orders o;**

4. List all orders by the customers not located in the same city as their salesman.

**(with city)**

**select \* from (select o.\*,(select city from salesman s where o.snum=s.snum) as S\_city,(select city from customers c where o.cnum=c.cnum) as C\_city from orders o) where NOT(S\_city=C\_city);**

**select onum,amount,odate,cnum,snum from (select o.\*,(select city from salesman s where o.snum=s.snum) as S\_city,(select city from customers c where o.cnum=c.cnum) as C\_city from orders o) where NOT(S\_city=C\_city);**

5. List all customers serviced by salespeople with commission above 12%.

**Select cname from customers where snum IN(Select snum from salesman where commission>12);**

6. Calculate the amount of the salesman commission on each order by a customer with rating above 100.

**select amount,comm,comm\*amount/100 "Comm Amount",rate from (select amount,(select commission from salesman s where o.snum=s.snum) as comm,(Select rating from customers c where o.cnum=c.cnum) as rate from orders o) where rate>100;**

7. Find all pairs of customers having the same rating without duplication.

**select c1.cname as c1.name,c2.cname as c2name from customers c1,customers c2 where c1.rating=c2.rating and c1.cnum!=c2.cnum and c1.cnum<c2.cnum order by c1.cnum;**

8. List all customers located in cities where salesman Sejal has customers.

**Select cname from customers where city IN(Select city from customers where snum=(select snum from salesman where sname='Sejal'));**

9. Find all pairs of customers served by a single salesman with the salesman's name and no.

**Select c1.cname,c2.cname,c1.snum,(select sname from salesman s where s.snum=c1.snum )”Sname” from customers c1,customers c2 where c1.snum=c2.snum and c1.cname!=c2.cname and c1.cnum<c2.cnum order by c1.snum,c1.cname,c2.cname;**

10. List all salesmen who are living in the same city without duplicate rows.

**select sname from salesman where city IN(select city from salesman group by city having count(\*)>1);**

11. List all pairs of orders by a given customer with customer name.

**select o.\*,(select cname from customers c where o.cnum=c.cnum) from orders o where cnum=&cnum;**

**//Select count(\*) from orders group by cnum**

**//Select distinct o1.onum,o2.onum,(select cname from customers c where c.cnum=o1.cnum) as custname from orders o1,orders o2 where o1.cnum=o2.cnum and o1.onum<=o2.onum order by custname,o1.onum,o2.onum ;**

12. Produce the name and city of all the customers with the same rating as Harsh.

**Select cname,city from customers where rating=(select rating from customers where cname='Harsh') and cname!='Harsh';**

13. Extract all orders of Miti.

**Select o.\* from orders o where snum=(Select snum from Salesman where sname='Miti');**

14. Extract all orders of Baroda's salesmen.

**Select o.\* from orders o where snum IN (Select snum from Salesman where city='Baroda');**

15. Find all orders of the salesman who services 'Harsh'.

**select o.\* From orders o where snum=(select snum from customers where cname='Harsh')**

16. List all orders that are greater than the average of October 4,1997.

**select \* from orders where amount>(select sum(amount)/Count(\*) from orders where odate='04-OCT-97');**

17. Find the average commission of salesmen in London

**Select sum(commission)/count(\*)"Avg Comm" from salesman where city='London';**

**Select avg(commission)"Avg Comm" from salesman where city='London';**

18. Find all orders attributed to salesmen in 'London' using both the subquery and join methods.

**Select (select sname from salesman s where o.snum=s.snum)”Salesman”,o.\* from orders o where snum IN(select snum from salesman where city=’London’);**

19. List the commiss ion of all salesmen serving customers in 'London'.

**Select snum,sname,commission from salesman where snum IN(select snum from orders where cnum IN(select cnum from customers where city=’London’));**

20. Find all customers whose cnum is 1000 above than the snum of Sejal.

**Select cnum,cname from customers where cnum>1000+(Select snum from salesman where sname=’Sejal’);**

21. Count the no. of customers with the rating above than the average of 'Surat'.

**Select count(\*)"No\_Cust" from customers where rating>(select sum(rating)/count(\*) from customers where city=’Surat’);**

22. List all orders of the customer 'Chirag'.

**Select \* from orders where cnum=(select cnum from customers where cname=’Chirag’);**

23. Produce the name and rating of all customers who have above average orders.

**Select cname,rating from customers where cnum IN(select cnum from orders where amount>(select avg(amount) from orders));**

24. Select the total amount in orders for each salesman for whom this total is greater than the amount of the largest order in the table.

**Select sum(amount) fromb orders group by snum having sum(amount)>(Select max(amount) from orders) ;**

25. List the name and number of all salesmen who has more than one customer.

**select snum,(select sname from salesman s where c.snum=s.snum) from customers c group by snum having count(\*)>1;**

**//select snum,(select sname from salesman s where c.snum=s.snum) from customers c group by snum having count(\*)>1;**

**Select sname,snum from salesman where snum IN(select snum from orders group by cnum having count(cnum)>1);**

26. Find all orders with amount atleast equal to the average amounts for their customers.

**select cnum,amount from orders o1 where amount>=(select avg(amount) from orders o2 where o1.cnum=o2.cnum group by cnum);**

27. Calculate the total amount ordered on each day eliminating those days where the total amount was not atleast Rs. 2000 above the maximum amount of that day.

**Select sum(amount),odate from orders group by odate having sum(amount)<2000+max(amount);**

**select odate,Total,Max from (Select odate,sum(amount)as Total,max(amount)as Max from orders group by odate) where Total<2000+max;**

**Select sum(amount) from orders o1 group by odate having sum(amount)<2000+(select max(amount) from orders o2 where o1.odate=o2.odate group by odate);**

28. Select the name and number of all salesmen who have customers in their cities who they do not service.

**select distinct sname,s.snum from customers c,salesman s where c.city=s.city and c.snum!=s.snum;**

29. Find the number of all the salesmen having multiple customers using EXIST.

**//**select count(\*) from salesman where snum IN (select snum from orders group by snum having count(\*)>1);

**Select count(\*) from salesman s where EXISTS(select 1 from customers c where s.snum=c.snum group by snum having count(\*)>1);**

30. Find the name,number and city of all the salesmen having multiple customers using EXIST.

**Select sname,snum,city from salesman s where EXISTS(select 1 from customers c where s.snum=c.snum group by snum having count(\*)>1);**

**//select sname,snum,city from salesman where snum IN (select snum from orders group by snum having count(\*)>1);**

31. Find the name and number of all the salesmen who serve only one customer.

**select sname,snum from salesman where snum IN (select snum from orders group by snum having count(\*)=1);**

32. Find all salesmen with more than one current order.

**select sname,snum from salesman where snum IN (select snum from orders group by snum having count(\*)>1);**

33. Display the customer information if and only if one or more of the customers in are located in 'Surat.

**select \* from customers where EXISTS(select count(\*) from customers group by city having city='Surat');**

34. Find all salesmen who have customers with more than one current order.

**select sname,snum from salesman where snum IN (select snum from orders group by snum having count(\*)>1);**

35. Find all salesmen who have customers with rating > 300 using EXIST and using join.

**Select snum,(select sname from salesman where c.snum=salesman.snum)”SNAME” from customers c where EXISTS(Select rating from customers c2 where c2.rating>300 and c.rating=c2.rating);**

36. Find all orders with amounts smaller than any amount for a customer in 'London'.

**Select \* from orders where amount< ANY(Select amount from orders where cnum IN(select cnum from customers where city=’London’));**

37. Find all the customers who have greater rating than every customer in 'Rome'.

**Select \* from customers where rating > ALL(select rating from customers where city=’Rome’);**

38. Select all customers whose rating doesn't match with any rating of customer of 'Surat'.

**Select \* from customers where rating NOT IN(select rating from customers where city=’Surat’);**

39. List all customers whose ratings are equal to or greater than ANY of 'Sejal'.

**select \* from customers where rating >= ANY(select rating from customers where cnum IN (select cnum from orders where snum=(select snum from salesman where sname='Sejal')));**

**select \* from customers where rating >= ANY(select rating from customers where cnum IN(select cnum from customers where snum=(select snum from salesman where sname='Sejal')));**

40. List all orders for amount greater than any for the customers in London.

**Select \* from orders where amount>ANY(select amount from orders where cnum IN(select cnum from customers where city=’London’));**

41. Find all salesmen and customers located in London.

**select sname "Londonite" from salesman where city='London' UNION select cname "Londonite" from customers where city='London';**

select sname,cname from salesman,customers where salesman.city=customers.city and salesman.city=’London’;

42. Find out which salesman produce largest and smallest orders on each date.

**select odate,max(amount)"Large",min(amount)"Small" from orders group by odate;**

Q2. Create the tables Employees and Departments having the following structures:

Sample table: Employees

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| EMPLOYEE\_ID | FIRST\_NAME | LAST\_NAME | SALARY | DEPARTMENT\_ID |
| 111 | Steven | King | 24000 | 20 |
| 112 | John | Hopkings | 12000 | 30 |
| 113 | Alexander | Roy | 10000 | 20 |
| 114 | Carlie | Nayer | 23000 | 20 |
| 115 | Julies | Ceaser | 8000 | 40 |
| 116 | James | Mathew | 9000 | 30 |
| 117 | Andrew | Matt | 5500 | 30 |
| 118 | Sunil | Pal | 25000 | 20 |
| 119 | Roshan | Kumar | 15000 | 40 |
| 120 | Rahul | Kapoor | 16000 | 40 |

Sample table: Departments

|  |  |
| --- | --- |
| DEPARTMENT\_ID | DEPARTMENT\_NAME |
| 20 | Accounting |
| 30 | IT |
| 40 | Marketting |

Perform the following operations using nested queries/sub-queries:

1. Find the names (first name as well as last name) and salaries of the employees who have higher salary than the employee whose last name is Hopkings.

**Select first\_name||' '||last\_name "Names" from employees1 where salary>(select salary from employees1 where last\_name=’Hopkings’);**

1. Find the names (first and last name both) of all the employees who work in the IT department.

**Select first\_name||' '||last\_name "Names" from employees1 where department\_id=(select department\_id from departments1 where department\_name=’IT’);**

1. Find the names (first\_name, last\_name), salary of the employees whose salary is greater than the average salary.

**Select first\_name||' '||last\_name "Names",salary from employees1 where salary>(select avg(salary) from employees1);**

1. Find the names (first\_name, last\_name), salary of the employees who earn more than the average salary and who works in any of the IT departments.

**Select first\_name||' '||last\_name "Names",salary from employees1 where salary>(select avg(salary) from employees1 group by department\_id having department\_id=(Select department\_id from departments1 where department\_name=’IT’));**

1. Find the details of the employees who earn the same salary as the minimum salary for all departments.

**Select \* from employees1 where salary IN(select min(salary) from employees1 group by department\_id);**

1. Find the details of the employees whose salary is greater than average salary of all department.

**//Select \* from employees1 where salary>(select avg(sal) from employees1);**

**Select \* from employees1 e1 where salary>( select avg(salary) from employees1 e2 where e1.department\_id=e2.department\_id group by department\_id);**

Q 3. Create the “Customers” table having the following structure:

ID (as a primary key), NAME, AGE, ADDRESS and SALARY

Populate the table with the following records:

ID NAME AGE ADDRESS SALARY

001 Ramesh 32 Ahmedabad 2000.00

002 Khalid 25 Delhi 1500.00

003 kaushik 23 Kota 2000.00

004 Chaitali 25 Mumbai 6500.00

005 Hardik 27 Bhopal 8500.00

006 Komal 22 MP 4500.00

007 Muffy 24 Indore 10000.00

Perform the following operations on the above table:

1. Create a view called view1 on Customers table that should include ID, NAME and AGE from Customers.

**Create view view1 as select id,name,age from customers1;**

1. Create a view called view2 on Customers table that should include the records of Customers whose salaries are greater than 5000.

Create view view2 as select \* from customers1 where salary>5000;

1. Insert the following records of customers in the view1:

ID NAME AGE

010 Fazil 27

014 Shoaib 31

016 John NULL

Show the results of View1 as well as the base table Customers after insertion operation.

Insert into view1 values(010,’Fazil’,27);

Insert into view1 values(014,’Shoaib’,31);

Insert into view1 values(016,’John’,NULL);

Select \* from customers1;

Select \* from view1;

1. Change the name of customer Komal to ABC by using view1. Display the results of view1 as well as the base table Customers.

Update view1 set name=’ABC’ where name=’Komal’;

Select \* from customers1;

Select \* from view1;

1. Modify the salary and age of Muffy in view2 to 15000 and 26 respectively.

Update view2 set salary=15000,age=26 where name=’Muffy’;

1. Delete a record from view1 whose ID is 004. Display the results of view1, view2 and the base table Customers.

Delete from view1 where id=004;

Select \* from view1;

Select \* from view2;

Select \* from Customers1;

1. Destroy the structure of view2.

Drop view view2;

1. Create a view called view3 on Customers table with name and age. Add check option on age to be not null.

Create view view3 as select name,age from customers1 where age is not null with check option;

Alter view view3 add constraint age not null;

Alter view view3 add constraint agenotnull CHECK(age NOT NULL);

Alter view view3 add constraint CHECK(age NOT NULL);

Alter view view3 modify age number(2) NOT NULL;

1. Insert a record in view3 (‘Rohan’, NULL) and (‘Joe’, 42). Display the results of view3 and Customers table.

Insert into view3 values(‘Rohan’,NULL);