**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 customer.

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.

**Create table salesmen(snum number(6) primary key not null,sname varchar2(15),city varchar2(12),commission number(4));**

**insert into salesmen values (1001,'piyush','london','12');**

**insert into salesmen values (1002,'sejal','surat','13');**

**insert into salesmen values (1004,'miti','london','11');**

**insert into salesmen values (1007,'rajesh','baroda','15');**

**insert into salesmen values (1003,'anand','new delhi','10');**

**create table customers (cnum number(6) primary key not null,cname varchar2(12),city varchar2(10),rating number(6),snum number(6) ,foreign key(snum) references salesmen(snum) );**

**insert into customers values (2002,'gita','rome',200,1003);**

**insert into customers values (2003,'lalit','surat',200,1002);**

**insert into customers values (2004,'govind','bombay',300,1002);**

**insert into customers values (2006,'chirag','london',100,1001);**

**insert into customers values (2008,'chinmay','surat',300,1007);**

**insert into customers values (2007,'pratik','rome',100,1004);**

**create table orders (onum number(6) primary key not null,amount number(8,2)**

**,odate date,cnum number(6),snum number(6), foreign key(snum) references salesmen(snum),foreign key(cnum) references customers(cnum));**

**insert into orders values (3001,18.69,'10-mar-97',2008,1007);**

**insert into orders values (3003,767.19,'10-mar-97',2001,1001);**

**insert into orders values (3005,5160.45,'10-mar-97',2003,1002);**

**insert into orders values (3006,1098.16,'10-mar-97',2008,1007);**

**insert into orders values (3009,1713.23,'10-apr-97',2002,1003);**

**insert into orders values (3007,75.75,'10-apr-97',2004,1002);**

**insert into orders values (3008,4723.00,'10-may-97',2006,1001);**

**insert into orders values (3010,1309.95,'10-jun-97',2004,1002);**

**insert into orders values (3011,9891.88,'10-jun-97',2006,1001);**

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,s.sname from salesmen s,customers c where c.snum=s.snum;**

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

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

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

**Select o.onum,o.amount,c.cname,s.sname from orders o,customers c,salesmen s where o.cnum=c.cnum and o.snum=s.snum;**

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

**Select o.onum,o.amount from orders o,customers c,salesmen s where s.snum=c.snum and o.cnum=c.cnum and c.city <> s.city;**

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

**Select c.cnum, c.cname from customers c, salesmen s where c.snum=s.snum and s.commission>12;**

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

**Select o.amount,o.onum,(s.commission\*o.amount/100) commission\_amount from**

**customers c, orders o,salesmen s where o.cnum=c.cnum AND c.rating>100 and o.snum**

**=s.snum;**

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

**Select c1.cname,c2.cname from customers c1,customers c2 where c1.rating=c2.**

**rating and c1.cname < c2.cname;**

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

**select c.cname from customers c where c.city IN (select c.city from custome**

**rs c where c.snum=(select s.snum from salesmen s where s.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 from customers c1, customers c2 where c1.snum=c2.snum and c1.cname<c2.cname;**

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

**Select s1.sname,s2.sname from salesmen s1,salesmen s2 where s1.city=s2.city and s1.sname<s2.sname;**

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

**Select o.onum, c.cname from orders o,customers c where o.cnum=c.cnum;**

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

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

13. Extract all orders of Miti.

**Select o.onum, o.amount from orders o where o.snum = (select s.snum from salesmen s where s.sname=’miti’);**

14. Extract all orders of Baroda's salesmen.

**Select o.onum, o.amount from orders o where o.snum IN (select s.snum from salesmen s where s.city=’baroda’);**

**Or**

**Select o.onum, o.amount from orders o,salesmen s where o.snum=s.snum and s.city=’baroda’;**

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

**Select o.onum, o.amount from orders o where o.snum= (Select c.snum from customers c where c.cname=’harsh’);**

**or**

**Select o.onum, o.amount from orders o,customers c where o.snum=c.snum and c**

**.cname='harsh';**

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

**Select onum, amount from orders where amount > (Select avg(amount) from orders where odate=’4-oct-97’);**

17. Find the average commission of salesmen in London

**Select avg(commission) from salesmen where city=’london’;**

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

**Select o.onum, o.amount from orders o, salesmen s where o.snum=s.snum and s.city=’london’;**

**or**

**Select o.onum, o.amount from orders o where o.snum IN (Select s.snum from salesmen s where s.city=’london’);**

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

**Select s.commission from salesmen s where s.snum IN (Select c.snum from customers c where c.city=’london’);**

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

**Select c.cnum, c.cname from customers c where c.cnum -1000 > (select snum from salesmen where sname=’sejal’);**

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

**Select count(\*) from customers where rating>(Select avg(rating) from customers where city=’surat’);**

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

**Select o.onum,o.amount from orders o where o.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 distinct(cnum) fro**

**m 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) from 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, sname from salesmen where snum IN (Select snum from customers group by snum having count(\*) > 1);**

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

**select onum, amount from orders a , (select avg(amount) avg\_amount, cnum from orders group by cnum) b where a.cnum=b.cnum and a.amount >= b.avg\_amount;**

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) from orders group by odate having sum(amount) > (Select max(amount)+2000 from orders 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 s.snum,s.sname from customers c,salesmen s where s.city=c.city and s.snum < >c.snum;**

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

S**elect distinct snum from customers a where exists (select snum from custo**

**mers b where a.snum=b.snum and a.cname<> b.cname);**

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

**Select snum,sname,city from salesmen where snum IN( select snum from customers a where exists (select snum from customers b where a.snum=b.snum and a.cname<> b.cname));**

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

**select snum,sname from salesmen where snum IN (select snum from customers g**

**roup by snum having count(\*) = 1);**

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

**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 cnum,cname,city from customers where (select count(\*) from customers where city=’surat’) >1;**

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

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

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

**Select snum, sname from salesmen s where EXISTS (select snum from customers c where c.rating>200 and s.snum=c.snum);**

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

**Select onum,amount 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 rating,cnum,cname from customers where rating > ALL (select rating from customers where city=’rome’);**

**OR**

**Select c1.rating,c1.cnum,c1.cname from customers c1 where c1.rating > ALL (**

**select c2.rating from customers c2 where c2.city='rome');**

**OR**

**Select c1.rating,c1.cname,c1.cnum from customers c1 where NOT EXISTS (select c2.rating from customers c2 where c2.city=’rome’ AND c1.rating <= c2.rating);**

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

**Select cname from customers where rating NOT IN (Select rating from customers where city=’surat’);**

**OR**

**Select cname from customers where rating != ALL (Select rating from custome**

**rs where city='surat');**

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

**Select cnum,rating from customers where rating >= ANY(Select rating from customers where snum =(Select snum from salesmen where sname=’sejal’));**

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

**Select onum,amount 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 s.sname from salesmen s where s.city=’london’ UNION select c.cname from customers c where c.city=’london’;**

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

**select o1.odate, o1.snum lrgst\_amt\_snum, o1.amount lrgst\_amount, o2.snum smalst\_ordr\_snum,o2.amount smalst\_amt from orders o1,orders o2 where (o1.amount,o2.amount) IN(select max(amount),min(amount) 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 |

**create table employees(emp\_id number(4) primary key not null,first\_name var**

**char(12),last\_name varchar(12),salary number(7),dept\_id number(3),foreign key(de**

**pt\_id) references departments(dept\_id) );**

**insert into employees values(111,'steven','king',24000,20);**

**insert into employees values(112,'john','hopkings',12000,30);**

**insert into employees values(113,'alexander','roy',10000,20);**

**insert into employees values(114,'carlie','nayer',23000,20);**

**insert into employees values(115,'julies','ceaser',8000,40);**

**insert into employees values(116,'james','mathew',9000,30);**

**insert into employees values(117,'andrew','matt',5500,30);**

**insert into employees values(118,'sunil','pal',25000,20);**

**insert into employees values(119,'roshan','kumar',15000,40);**

**insert into employees values(120,'rahul','kapoor',16000,40);**

Sample table: Departments

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

**Create table departments(dept\_id number(3) primary key not null,dept\_name varchar2(15));**

**Insert into departments values(20,’accounting’);**

**Insert into departments values(30,’it’);**

**Insert into departments values(40,’marketing’);**

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,salary from employees where salary >(select salary from employees 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 from employees where dept\_id=(select dept\_id from departments where dept\_name=’it’);**

**OR**

**Select first\_name,last\_name from employees e, departments d where e.dept\_id=d.dept\_id and d.dept\_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, salary from employees where salary>(select avg(salary) from employees);**

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,salary from employees where salary >(select avg(salary) from employees) and dept\_id=(select dept\_id from departments where dept\_name='it');**

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

**select \* from employees where salary IN ( select min(salary) from employees group by dept\_id);**

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

**select \* from employees where salary > ALL ( select avg(salary) from employees group by dept\_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

**create table customer(id number(4) primary key not null,name varchar2(12),a**

**ge number(3),address varchar2(18),salary number(6,2));**

**insert into customer values(001,'ramesh',32,'ahmedabad',2000.00);**

**insert into customer values(002,'khalid',25,'delhi',1500.00);**

**insert into customer values(003,'kaushik',23,'kota',2000.00);**

**insert into customer values(004,'chaitali',25,'mumbai',6500.00);**

**insert into customer values(005,'hardik',27,'bhopal',8500.00);**

**insert into customer values(006,'komal',22,'mp',4500.00);**

**insert into customer values(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 customer;**

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 customer 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

**insert into view1 values(010,'fazil',27);**

**insert into view1 values(014,'shoaib',31);**

**insert into view1 values(016,'john',NULL);**

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

SQL> select \* from view1;

ID NAME AGE

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

1 ramesh 32

2 khalid 25

3 kaushik 23

4 chaitali 25

5 hardik 27

6 komal 22

7 muffy 24

10 fazil 27

14 shoaib 31

16 john

10 rows selected.

SQL> select \* from customer;

ID NAME AGE ADDRESS SALARY

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

1 ramesh 32 ahmedabad 2000

2 khalid 25 delhi 1500

3 kaushik 23 kota 2000

4 chaitali 25 mumbai 6500

5 hardik 27 bhopal 8500

6 komal 22 mp 4500

7 muffy 24 indore 10000

10 fazil 27

14 shoaib 31

16 john

10 rows selected.

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';**

SQL> select \* from view1;

ID NAME AGE

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

1 ramesh 32

2 khalid 25

3 kaushik 23

4 chaitali 25

5 hardik 27

6 ABC 22

7 muffy 24

10 fazil 27

14 shoaib 31

16 john

10 rows selected.

SQL> select \* from customer;

ID NAME AGE ADDRESS SALARY

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

1 ramesh 32 ahmedabad 2000

2 khalid 25 delhi 1500

3 kaushik 23 kota 2000

4 chaitali 25 mumbai 6500

5 hardik 27 bhopal 8500

6 ABC 22 mp 4500

7 muffy 24 indore 10000

10 fazil 27

14 shoaib 31

16 john

10 rows selected.

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;**

SQL> select \* from view1;

ID NAME AGE

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

1 ramesh 32

2 khalid 25

3 kaushik 23

5 hardik 27

6 ABC 22

7 muffy 26

10 fazil 27

14 shoaib 31

16 john

9 rows selected.

SQL> select \* from view2;

ID NAME AGE ADDRESS SALARY

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

5 hardik 27 bhopal 8500

7 muffy 26 indore 15000

SQL> select \* from customer;

ID NAME AGE ADDRESS SALARY

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

1 ramesh 32 ahmedabad 2000

2 khalid 25 delhi 1500

3 kaushik 23 kota 2000

5 hardik 27 bhopal 8500

6 ABC 22 mp 4500

7 muffy 26 indore 15000

10 fazil 27

14 shoaib 31

16 john

9 rows selected.

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 customer where age IS NOT NULL WITH CHECK OPTION;**

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

**SQL> insert into view3 values('rohan',null);**

**insert into view3 values('rohan',null)**

**\***

**ERROR at line 1:**

**ORA-01400: cannot insert NULL into ("SYSTEM"."CUSTOMER"."ID")**

**SQL> insert into view3 values('joe',42);**

**insert into view3 values('joe',42)**

**\***

**ERROR at line 1:**

**ORA-01400: cannot insert NULL into ("SYSTEM"."CUSTOMER"."ID")**