Lab7

create table Salesman

(

salesman\_id numeric(5,0) primary key,

name varchar(50),

city varchar(50),

commission numeric(3,2)

);

insert into Salesman values(5001,'James Hoog','New York',0.15);

insert into Salesman values(5002,'Nail Knite','Paris',0.13);

insert into Salesman values(5005,'Pit Alex','London',0.11);

insert into Salesman values(5006,'Mc Lyon','Paris',0.14);

insert into Salesman values(5003,'Lauson Hen',' ',0.12);

insert into Salesman values(5007,'Paul Adam','Rome',0.13);

)

select \* from Salesman

create table Customer

(

customer\_id numeric(5,0) primary key,

cust\_name varchar(80),

city varchar(80),

grade numeric(3,0),

salesman\_id numeric(5,0) references Salesman

);

insert into Customer values(3002,'Nick Rimando','New York',100,5001);

insert into Customer values(3005,'Graham Zusi','California',200,5002);

insert into Customer values(3001,'Brad Guzan','London',NULL,5005);

insert into Customer values(3004,'Fabian Johns','Paris',300,5006);

insert into Customer values(3007,'Brad Davis','New York',200,5001);

insert into Customer values(3009,'Geoff Camero','Berlin',100,5003);

insert into Customer values(3008,'Julian Green','London',300,5002);

insert into Customer values(3003,'Jozy Altidor','Moncow',200,5007);

select \* from Customer

create table Orders

(

ord\_no numeric(5,0) primary key,

purch\_amt numeric(10,3),

ord\_date date,

customer\_id numeric(5,0) references Customer,

salesman\_id numeric(5,0) references Salesman

);

insert into Orders values(70001,150.5,'2012-10-05',3005,5002);

insert into Orders values(70009,270.65,'2012-09-10',3001,5005);

insert into Orders values(70002,65.26,'2012-10-05',3002,5001);

insert into Orders values(70004,110.5,'2012-08-17',3009,5003);

insert into Orders values(70007,948.5,'2012-09-10',3005,5002);

insert into Orders values(70005,2400.6,'2012-07-27',3007,5001);

insert into Orders values(70008,5760,'2012-09-10',3002,5001);

insert into Orders values(70010,1983.43,'2012-10-10',3004,5006);

insert into Orders values(70003,2480.4,'2012-10-10',3009,5003);

insert into Orders values(70012,250.45,'2012-06-27',3008,5002);

insert into Orders values(70011,75.29,'2012-08-17',3003,5007);

insert into Orders values(70013,3045.6,'2012-04-25',3002,5001);

select \* from orders

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

select o.ord\_no,o.ord\_date,s.name,s.salesman\_id from orders o, salesman s where s.salesman\_id = o.salesman\_id and s.name = 'Paul Adam';

1. Write a query to display all the orders for the salesman who belongs to the city New York.

select \* from orders, salesman where orders.salesman\_id = salesman.salesman\_id and salesman.city='New York';

1. Write a query to find all the orders issued against the salesman who works for customer whose id is 3007.

select \* from orders, salesman where orders.salesman\_id =(select salesman\_id from customer where customer\_id='3007');

1. Write a query to display all the orders which values are greater than the average order value for 10th October 2012.

select \* from orders where purch\_amt > ( select avg(purch\_amt) from orders where ord\_date='2012-10-10');

1. Write a query to find all orders attributed to salesman in New York.

select \* from orders, salesman where orders.salesman\_id = salesman.salesman\_id and salesman.city='New York';

1. Write a query to display the commission of all the salesmen servicing customers in Paris.

select \* from salesman where salesman.salesman\_id=(select salesman\_id from customer where city='Paris');

1. Write a query to display all customers with orders on October 5, 2012.

select customer\_id,cust\_name from customer where customer\_id in (select customer\_id from orders where ord\_date='2012-10-5');

1. Write a query to display all the customers with orders issued on date 17th August, 2012

select o.ord\_no,customer.cust\_name from customer,orders o where o.customer\_id= customer.customer\_id and o.ord\_date='2012-08-17';

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

select salesman\_id,name from salesman where salesman\_id in(select salesman\_id from customer group by salesman\_id having count(salesman\_id)>1)

1. Write a queries to find all orders with order amounts which is above-average amounts for their customers.

select \* from orders where purch\_amt > (select avg(purch\_amt) from orders, customer where orders.customer\_id= customer.customer\_id);

1. 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 at least 1000.00 above the maximum order amount for that date.

select ord\_date,sum(purch\_amt) from orders group by ord\_date having sum(purch\_amt)>max(purch\_amt)+1000;

1. Write a query to extract the data from the customer table if and only if one or more of the customers in the customer table are located in London.

select \* from customer where exists (select city from customer where city='London');

1. Write a query to find the salesmen who have multiple customers

select \* from salesman where salesman\_id in (select salesman\_id from customer group by salesman\_id having count(salesman\_id)>1)

1. Write a query to find all the salesmen who worked for only one customer.

select \* from salesman where salesman\_id in (select salesman\_id from customer group by salesman\_id having count(salesman\_id)=1)

1. Write a query to find salesman with customers located in their cities.

select s.name,c.cust\_name from salesman as s, customer as c where s.city=c.city and s.salesman\_id=c.salesman\_id;

1. Write a query to display all the orders that had amounts that were greater than at least one of the orders from October 9th 2012.

select \* from orders where purch\_amt> any(select purch\_amt from orders where ord\_date='2012-10-09');

1. Write a query to find all orders with amount smaller than any amount for a customer in London

select \* from orders where purch\_amt< any(select purch\_amt from orders where customer\_id in(select customer\_id from customer where city='London'));

1. Write a query to display all orders with amount smaller than any amount for a customer in London.

select \* from orders where purch\_amt< any(select purch\_amt from orders where customer\_id in(select customer\_id from customer where city='London'));

1. Write a query to display only those customers whose grade are, in fact, higher than every customer in New York.

select \* from customer where grade> all(select grade from customer where city='New York');

1. Write a query to find all those customers who holds a different grade than any customer of the city Dallas.

select \* from customer where grade not in (select grade from customer where city='Dallas');

Aggregate Functions:

1. Write a SQL statement to find the total purchase amount of all orders

select sum(purch\_amt) from orders;

1. Write a SQL statement to find the average purchase amount of all orders

select avg(purch\_amt) from orders;

1. Write a SQL statement to find the number of salesmen currently listing for all of their customers.

select count(distinct salesman\_id) from customer;

1. Write a SQL statement know how many customer have listed their names.

select count(customer\_id) from customer;

1. Write a SQL statement find the number of customers who gets at least a gradation for his/her performance.

select count(customer\_id) from customer where grade>0;

1. Write a SQL statement to get the maximum purchase amount of all the orders.

select max(purch\_amt) from orders;

1. Write a SQL statement to get the minimum purchase amount of all the orders

select min(purch\_amt) from orders;

1. Write a SQL statement which selects the highest grade for each of the cities of the customers.

select city,max(grade) from customer group by city;

1. Write a SQL statement to find the highest purchase amount ordered by the each customer with their ID and highest purchase amount.

select customer\_id,max(purch\_amt) from orders group by customer\_id;

1. Write a SQL statement to find the highest purchase amount ordered by the each customer on a particular date with their ID, order date and highest purchase amount.

select customer\_id,ord\_date,max(purch\_amt) from orders group by customer\_id,ord\_date;

1. Write a SQL statement to find the highest purchase amount on a date '2012-08-17' for each salesman with their ID.

select salesman\_id,ord\_date,max(purch\_amt) from orders group by salesman\_id,ord\_date order by ord\_date;

1. Write a SQL statement to find the highest purchase amount with their ID and order date, for only those customers who have highest purchase amount in a day is more than 2000

select customer\_id,ord\_date,max(purch\_amt) from orders group by customer\_id,ord\_date having max(purch\_amt)>2000;

1. Write a SQL statement to find the highest purchase amount with their ID and order date, for those customers who have a higher purchase amount in a day is within the range 2000 and 6000.

select customer\_id,ord\_date,max(purch\_amt) from orders group by customer\_id,ord\_date having max(purch\_amt)>2000 and max(purch\_amt)<6000;

1. Write a SQL statement to find the highest purchase amount with their ID, for only those customers whose ID is within the range 3002 and 3007.

select customer\_id,max(purch\_amt) from orders group by customer\_id having customer\_id>=3002 and customer\_id<=3007;

1. Write a SQL statement to display customer details (ID and purchase amount) whose IDs are within the range 3002 and 3007 and highest purchase amount is more than 1000.

select customer\_id,max(purch\_amt) from orders group by customer\_id having customer\_id>=3002 and customer\_id<=3007 and max(purch\_amt)>1000;

1. Write a SQL statement to find the highest purchase amount with their ID, for only those salesmen whose ID is within the range 5003 and 5008.

select salesman\_id,max(purch\_amt) from orders group by salesman\_id having salesman\_id>=5003 and salesman\_id<=5008;

1. Write a SQL statement that counts all orders for a date August 17th, 2012

select count(ord\_no) from orders where ord\_date='2012-08-17';

1. Write a SQL statement that counts the number of different non NULL city values for salesmen.

select count(salesman\_id) from salesman where city!=' ';

1. Write a query that counts the number of salesmen with their order date and ID registering orders for each day.

select ord\_date,count(salesman\_id) from orders group by ord\_date;

Quering Multiple Tables

1. Write a query to find those customers with their name and those salesmen with their name and city who lives in the same city

select s.name,c.cust\_name from salesman as s,customer as c where s.city=c.city;

1. Write a SQL statement to find the names of all customers along with the salesmen who works for them.

select c.cust\_name,s.name from salesman as s,customer as c where s.salesman\_id=c.salesman\_id;

1. Write a SQL statement to display all those orders by the customers not located in the same cities where their salesmen lives.

select \* from orders where customer\_id not in (select c.customer\_id from salesman as s,customer as c where s.city=c.city);

1. Write a SQL statement that find out each order number followed by the name of the customers who made the order.

select o.ord\_no,c.cust\_name from orders as o,customer as c where c.customer\_id=o.customer\_id;

1. Write a SQL statement that short out the customer and their grade who made an order. Each of the customer must have a grade and served by at least a salesman, who belongs to a city.

select distinct c.customer\_id,c.grade from salesman as s,customer as c where s.city=c.city and c.grade>0;

1. Write a query that produces all customers with their name, city, salesman and commission, who served by a salesman and the salesman works at a rate of commission within 12% to 14%.

select c.cust\_name,s.name,s.commission from customer as c, salesman as s where c.salesman\_id=s.salesman\_id and (s.commission>=0.12 and s.commission<=0.14);

1. Write a SQL statement that produces all orders with order number, customer name, commission rate and earned commission amount for those customers who carry their grade more than 200 and served by an existing salesman.

select o.ord\_no,c.cust\_name,o.purch\_amt,s.commission,o.purch\_amt\*s.commission as comm\_amt from orders as o,customer as c, salesman as s where o.customer\_id=c.customer\_id and o.salesman\_id=s.salesman\_id and c.grade>200;