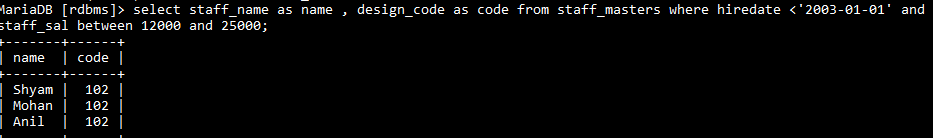
**Data Query Language**

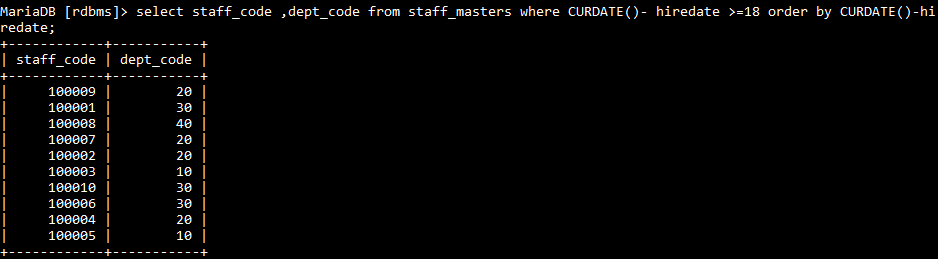
1. **List the Name and Designation code of the staff who have joined before Jan 2003 and whose salary range is between 12000 and 25000. Display the columns with user defined Column headers. Hint: Use As clause along with other operators**

select staff\_name as name , design\_code as code from staff\_masters where hiredate <'2003-01-01' and staff\_sal between 12000 and 25000;



1. **List the staff code, name, and department number of the staff who have experience of 18 or more years and sort them based on their experience**

select staff\_code ,dept\_code from staff\_masters where CURDATE()- hiredate >=18 order by CURDATE()-hiredate;

****

1. **Display the staff details who do not have manager. Hint: Use is null**

select \* from staff\_masters where mgr\_code is null;



1. **Display the Book details that were published during the period of 2001 to 2004. Also display book details with Book name having the character ‘&’ anywhere.**

select \* from book\_masters where book\_pub\_year between 2001 and 2004 or book\_name like '%&%'; 

1. List the names of the staff having ‘\_’ character in their name.

select staff\_name from staff\_masters where '\_' in(staff\_name);

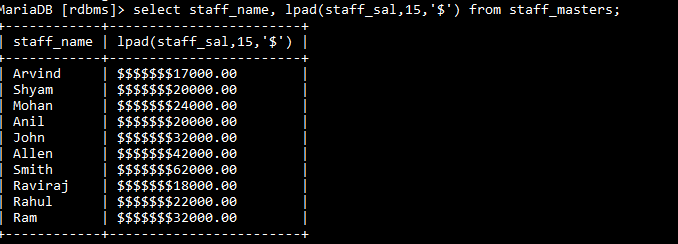


**Single Row And Group Functions**

**2.1**

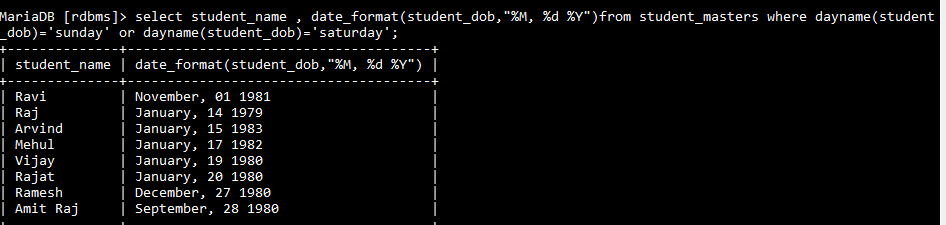
**1. Create a query which will display Staff Name, Salary of each staff. Format the salary to be 15 characters long and left padded with ‘$’.**

1. select staff\_name , lpad(staff\_sal,15,'$') from staff\_masters;

****

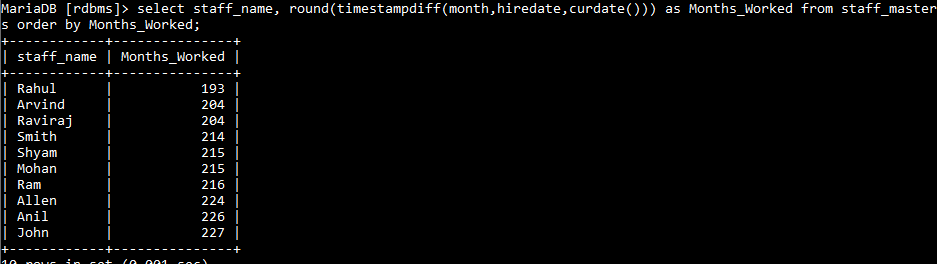
**2. Display name and date of birth of students where date of birth must be displayed in the format similar to “January, 12 1981” for those who were born on Saturday or Sunday.**

select student\_name , date\_format(student\_dob,"%M, %d %Y")from student\_masters where dayname(student\_dob)='sunday' or dayname(student\_dob)='saturday';

****

**3. Display each Staff name and number of months they worked for the organization. Label the column as ‘Months Worked’. Order your result by number of months employed. Also Round the number of months to closest whole number.**

select staff\_name, round(timestampdiff(month,hiredate,curdate())) as Months\_Worked from staff\_masters order by Months\_Worked;



4**. List the details of the staff who have joined in first half of December month (irrespective of the year).**

select \* from staff\_masters where month(hiredate)=12 and date(hiredate) between 1 and 15;

**5.Write a query that displays Staff Name, Salary, and Grade of all staff. Grade depends on the following table. Salary Grade Salary >=50000 A Salary >= 25000 < 50000 B Salary>=10000 < 25000 C OTHERS**

select staff\_name,staff\_sal,

-> case

-> when staff\_sal>=50000 then 'A'

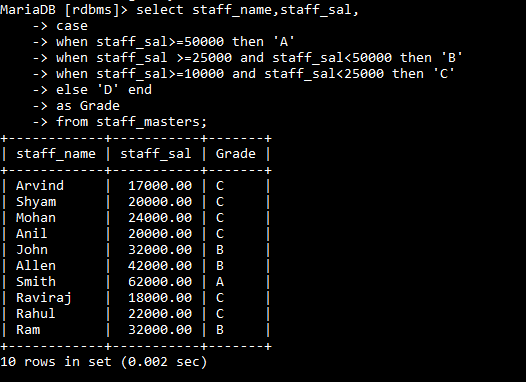
-> when staff\_sal >=25000 and staff\_sal<50000 then 'B'

-> when staff\_sal>=10000 and staff\_sal<25000 then 'C'

-> else 'D' end

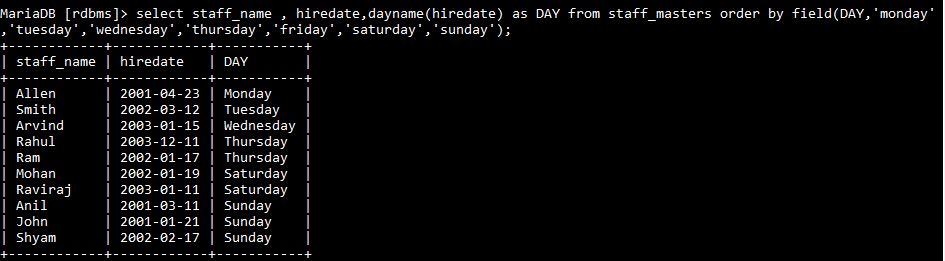
-> as Grade

-> from staff\_masters;



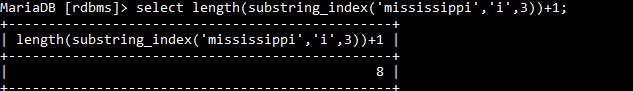
**6. Display the Staff Name, Hire date and day of the week on which staff was hired. Label the column as DAY. Order the result by the day of the week.**

select staff\_name , hiredate,dayname(hiredate) as DAY from staff\_masters order by field(DAY,'monday','tuesday','wednesday','thursday','friday','saturday','sunday');



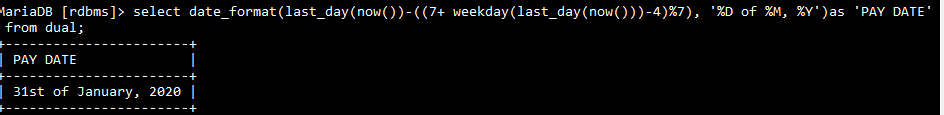
1. **Write a query to find the position of third occurrence of ‘i’ in the given word ‘Mississippi’.**

select length(substring\_index('mississippi','i',3))+1;



1. **Write a query to find the pay date for the month. Pay date is the last Friday of the month. Display the date in the format “Twenty Eighth of January, 2002”. Label the heading as PAY DATE. Hint: use to\_char, next\_day and last\_day functions**

select date\_format(last\_day(now())-((7+ weekday(last\_day(now()))-4)%7), '%D of %M, %Y')as 'PAY DATE' from dual;



1. **Display Student code, Name and Dept Name. Display “Electricals” if dept code = 20, “Electronics” if Dept code =30 and “Others” for all other Dept codes in the Dept Name column. Hint : Use Decode**

select student\_code, student\_name,

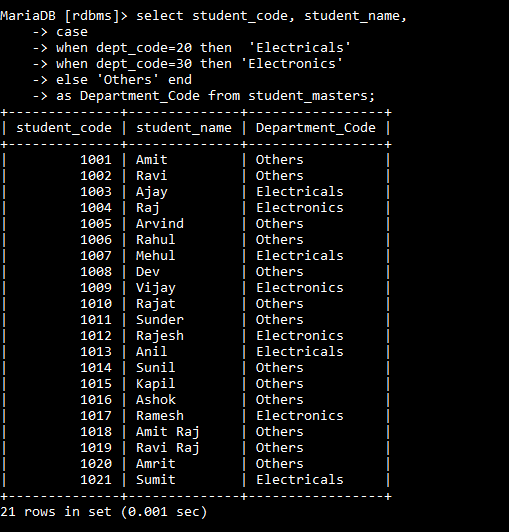
-> case

-> when dept\_code=20 then 'Electricals'

-> when dept\_code=30 then 'Electronics'

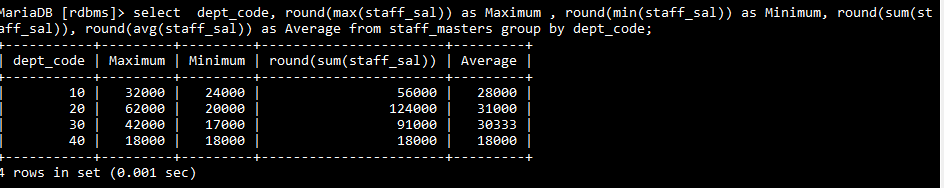
-> else 'Others' end

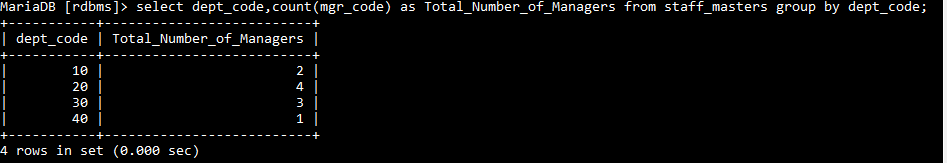
-> as Department\_Code from student\_masters;



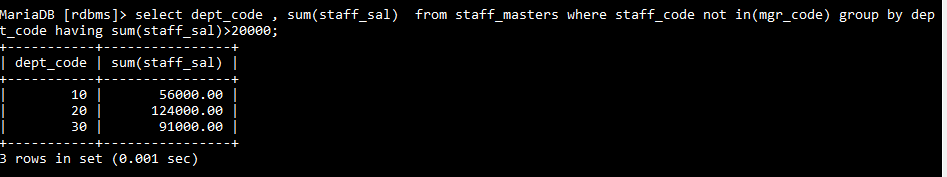
**2.2 Group Functions**

**1. Display the Highest, Lowest, Total & Average salary of all staff. Label the columns Maximum, Minimum, Total and Average respectively for each Department code. Also round the result to the nearest whole number.**

1. select dept\_code, round(max(staff\_sal)) as Maximum , round(min(staff\_sal)) as Minimum, round(sum(staff\_sal)), round(avg(staff\_sal)) as Average from staff\_masters group by dept\_code;
2. **Display Department code and number of managers working in that department. Label the column as ‘Total Number of Managers’ for each department.**

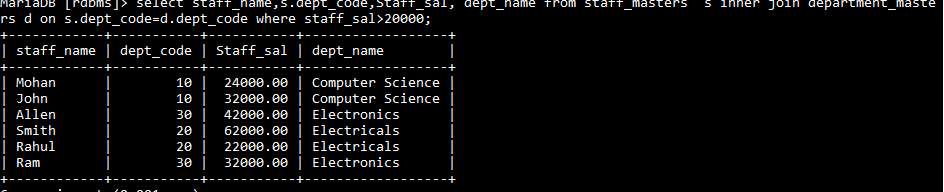
select dept\_code,count(mgr\_code) as Total\_Number\_of\_Managers from staff\_masters group by dept\_code;

1. **Get the Department number, and sum of Salary of all non-managers where the sum is greater than 20000**.

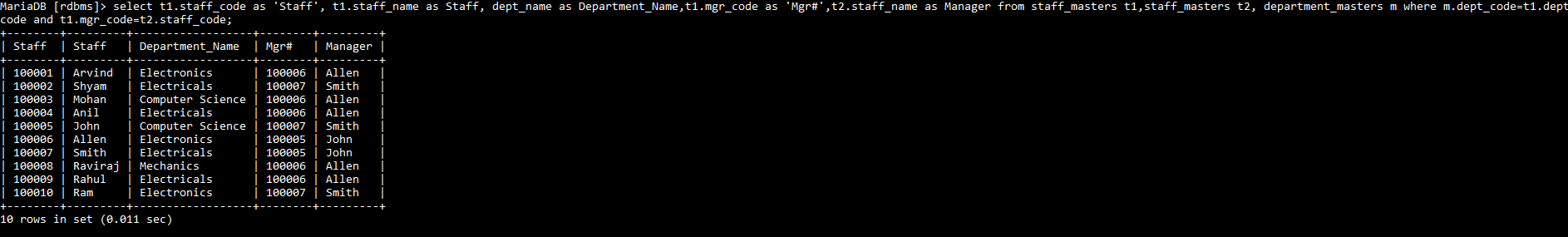
select dept\_code , sum(staff\_sal) from staff\_masters where staff\_code not in(mgr\_code) group by dept\_code having sum(staff\_sal)>20000;

**3.JOINS AND SUBQUERIES**

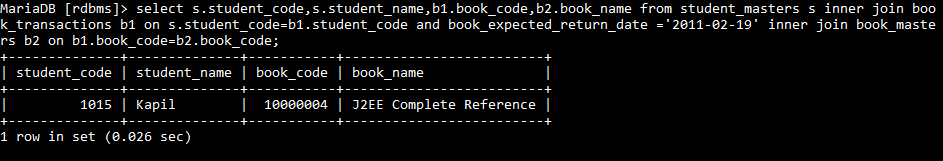
1. **Write a query which displays Staff Name, Department Code, Department Name, and Salary for all staff who earns more than 20000.**

select staff\_name,s.dept\_code,Staff\_sal, dept\_name from staff\_masters s inner join department\_masters d on s.dept\_code=d.dept\_code where staff\_sal>20000;

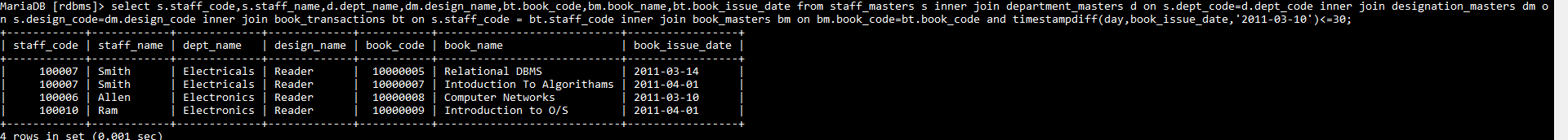
1. **Display Staff Code, Staff Name, Department Name, and his manager’s number and name. Label the columns Staff#, Staff, Mgr#, Manager.**

select t1.staff\_code as 'Staff', t1.staff\_name as Staff, dept\_name as Department\_Name,t1.mgr\_code as 'Mgr#',t2.staff\_name as Manager from staff\_masters t1,staff\_masters t2, department\_masters m where m.dept\_code=t1.dept\_code and t1.mgr\_code=t2.staff\_code;

1. **Create a query that will display Student Code, Student Name, Book Code, and Book Name for all students whose expected book return date is today.**

select s.student\_code,s.student\_name,b1.book\_code,b2.book\_name from student\_masters s inner join book\_transactions b1 on s.student\_code=b1.student\_code and book\_expected\_return\_date ='2011-02-19' inner join book\_masters b2 on b1.book\_code=b2.book\_code; 

1. **Create a query that will display Staff Code, Staff Name, Department Name, Designation name, Book Code, Book Name, and Issue Date for only those staff who have taken any book in last 30 days. . If required, make changes to the table to create such a scenario.**

Selects.staff\_code,s.staff\_name,d.dept\_name,dm.design\_name,bt.book\_code,bm.book\_name,bt.book\_issue\_date from staff\_masters s inner join department\_masters d on s.dept\_code=d.dept\_code inner join designation\_masters dm on s.design\_code=dm.design\_code inner join book\_transactions bt on s.staff\_code = bt.staff\_code inner join book\_masters bm on bm.book\_code=bt.book\_code and timestampdiff(day,book\_issue\_date,'2011-03-10')<=30;

1. **Generate a report which contains the following information.**

**Staff Code, Staff Name, Designation Name, Department, Book Code, Book Name,**

**Author, Fine For the staff who has not returned the book. Fine will be calculated as Rs. 5 per day.**

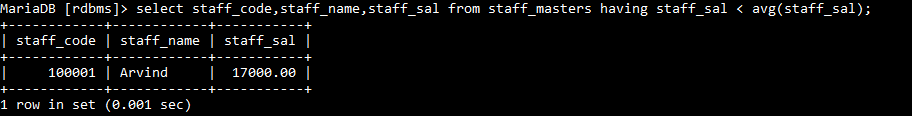
**Fine = 5 \* (No. of days = Current Date – Expected return date). Include records in the table to suit this problem statement**

select s.staff\_code , s.staff\_name,d.dept\_name,de.design\_name,b.book\_code,b.book\_name,t.book\_issue\_date,

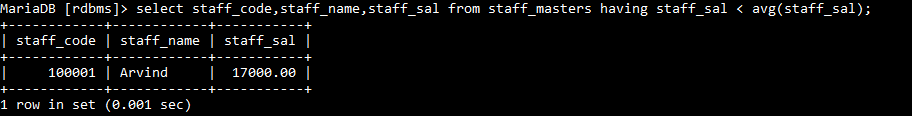
-> (datediff(curdate(),t.book\_expected\_return\_date)\*5) as fine

-> from staff\_masters s , department\_masters d , designation\_masters de, book\_masters b, book\_transactions t

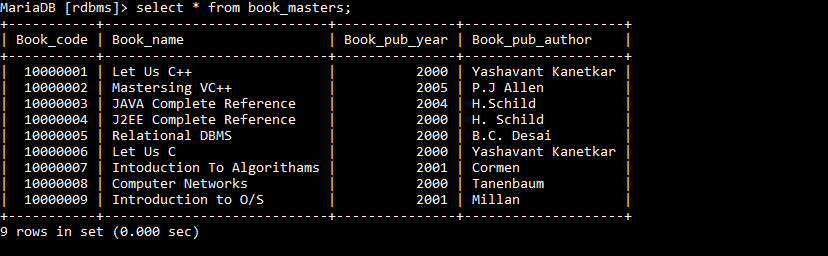
-> where s.staff\_code=t.staff\_code and d.dept\_code=s.dept\_code and de.design\_code=s.design\_code

-> and t.book\_code=b.book\_code and t.book\_actual\_return\_date is null;

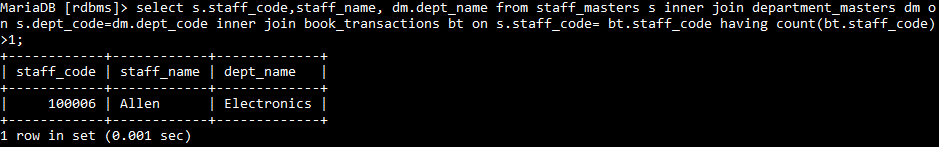
6**. List Staff Code, Staff Name, and Salary for those who are getting less than the average salary of organization.**

select staff\_code,staff\_name,staff\_sal from staff\_masters having staff\_sal < avg(staff\_sal);

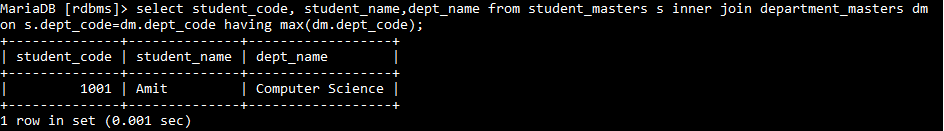
**7.Display Author Name, Book Name for those authors who wrote more than one book**

select book\_name,book\_pub\_author from book\_masters having count(book\_pub\_author)>1;

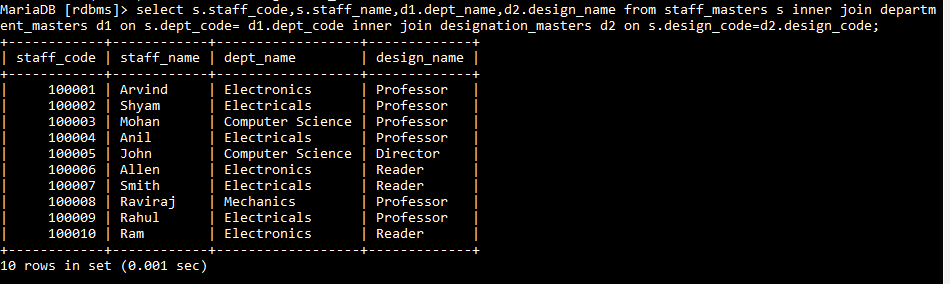
**8. Display Staff Code, Staff Name, and Department Name for those who have taken more than one book.**

select s.staff\_code,staff\_name, dm.dept\_name from staff\_masters s inner join department\_masters dm on s.dept\_code=dm.dept\_code inner join book\_transactions bt on s.staff\_code= bt.staff\_code having count(bt.staff\_code)>1;

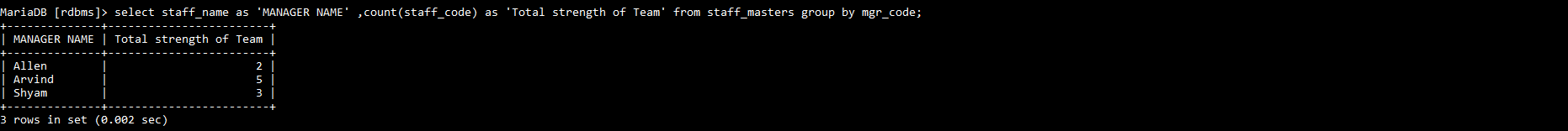
**9.Display the Student Code, Student Name, and Department Name for that department in which there are maximum number of student studying.**

select student\_code, student\_name,dept\_name from student\_masters s inner join department\_masters dm on s.dept\_code=dm.dept\_code having max(dm.dept\_code);

1. **Display Staff Code, Staff Name, Department Name, and Designation name for those who have joined in last 3 months.**

select s.staff\_code,s.staff\_name,d1.dept\_name,d2.design\_name from staff\_masters s inner join department\_masters d1 on s.dept\_code= d1.dept\_code inner join designation\_masters d2 on s.design\_code=d2.design\_code;

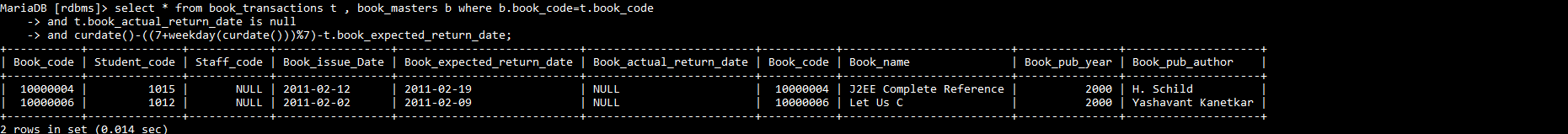
1. **Display the Manager Name and the total strength of his/her team.**

select staff\_name as 'MANAGER NAME' ,count(staff\_code) as 'Total strength of Team' from staff\_masters group by mgr\_code;

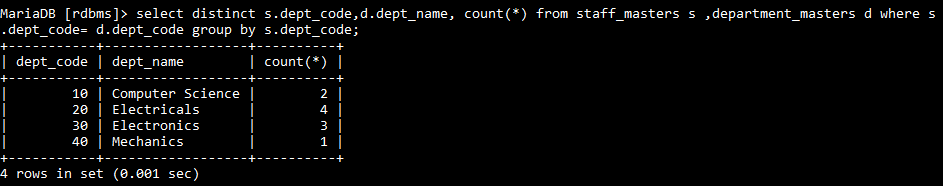
1. **Display the details of books that have not been returned and expected return date was last Monday. Book name should be displayed in proper case.. Hint: You can change /add records so that the expected return date suits this problem statement**

select \* from book\_transactions t , book\_masters b where b.book\_code=t.book\_code

-> and t.book\_actual\_return\_date is null

-> and curdate()-((7+weekday(curdate()))%7)-t.book\_expected\_return\_date;

13. **Write a query to display number of people in each Department. Output should display Department Code, Department Name and Number of People.**

select distinct s.dept\_code,d.dept\_name, count(\*) from staff\_masters s ,department\_masters d where s.dept\_code= d.dept\_code group by s.dept\_code;

4.1 **DATABASE OBJECTS**

**1. Create the Customer table with the following columns.**

**CustomerId int(5)**

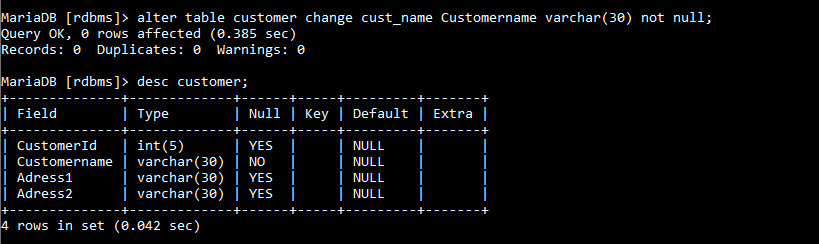
**Cust\_Name varchar(20)**

**Address1 Varchar(30)**

**Address2 Varchar(30)**

create table Customer(CustomerId int(5), Cust\_name varchar(20), Adress1 varchar(30),Adress2 varchar(30));

2**. Modify the Customer table Cust\_Name column of datatype with Varchar2(30), rename the column to CustomerName and it should not accept Nulls.**

alter table customer change cust\_name Customername varchar(30) not null;

3. **Add the following Columns to the Customer table.**

**Gender Varchar(1)**

**Age int(3)**

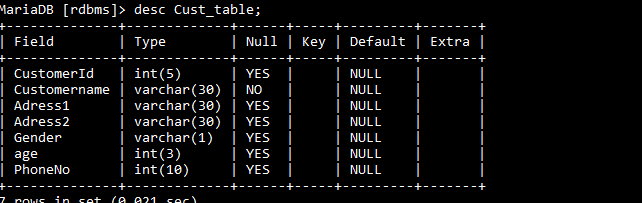
**PhoneNo int(10)**

**b) Rename the Customer table to Cust\_Table**

alter table customer add Gender varchar(1);

alter table customer add age int(3);

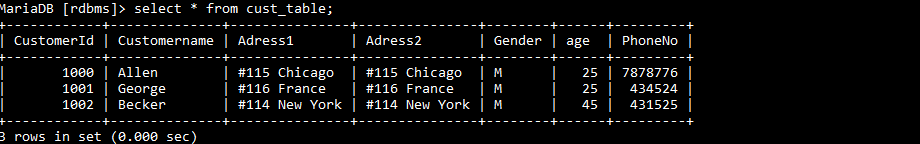
alter table customer add PhoneNo int(10);

3b. alter table customer rename to Cust\_Table;

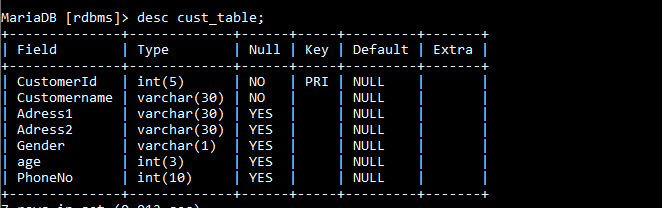
1. **Insert rows with the following data in to the Customer table.**

insert into cust\_table values(1000,'Allen','#115 Chicago','#115 Chicago','M',25,7878776);

insert into cust\_table values(1001,'George','#116 France','#116 France','M',25,434524);

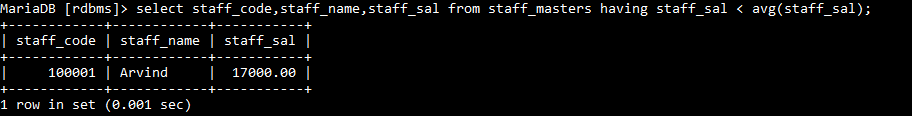
insert into cust\_table values(1002,'Becker','#114 New York','#114 New York','M',45,431525);

1. **Add the Primary key constraint for Customerld with the name Custld\_Prim.**

alter table cust\_table add constraint CustId\_Prim primary key(CustomerId);

1. **Insert the row given below in the Customer table and see the message generated by the Oracle server.**

**1002, John, #114 Chicago, #114 Chicago, M, 45, 439525**

insert into Cust\_table values(1002,'John','#114 Chicago','#114 chicago','M',45,439525);

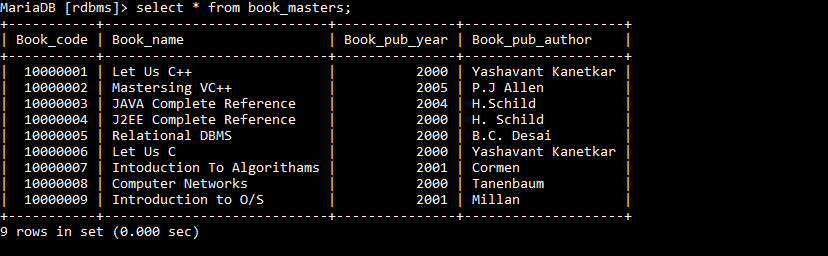
1. **Disable the constraint on CustomerId, and insert the following data:**

**1002, Becker, #114 New York, #114 New york , M, 45, 431525**

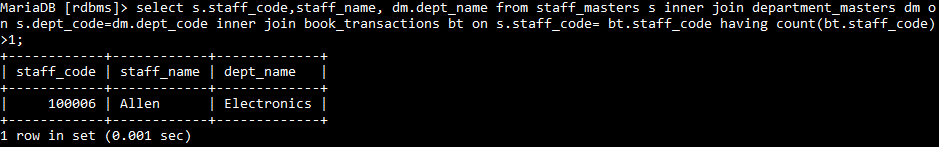
**1003, Nanapatekar, #115 India, #115 India , M, 45, 431525**

alter table cust\_table drop primary key;

insert into cust\_table values(1002,'Becker','#114 New York','#114 New York','M',45,431525);

insert into cust\_table values(1003,'Nanapatekar','#115 India','#115 India','M',45,431525);

1. **Enable the constraint on CustomerId of the Customer table, and see the message generated by the Oracle server.**

alter table cust\_table add primary key(Customerid);

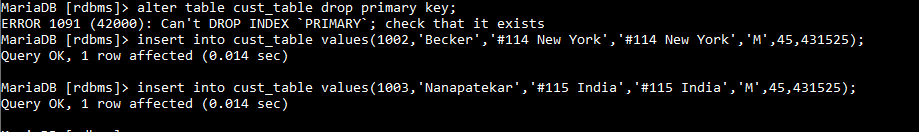
1. **Drop the constraint Custld\_Prim on CustomerId and insert the following Data. Alter Customer table, drop constraint Custid\_Prim.**

**1002, Becker, #114 New York, #114 New york , M, 45, 431525, 15000.50**

**1003, Nanapatekar, #115 India, #115 India , M, 45, 431525, 20000.50**

alter table cust\_table drop primary key;

insert into cust\_table values(1002,'Becker','#114 New York','#114 New York','M',45,431525);

insert into cust\_table values(1003,'Nanapatekar','#115 India','#115 India','M',45,431525);****

1. **Delete all the existing rows from Customer table, and let the structure remain itself using TRUNCATE statement.**

truncate table cust\_table;

1. **In the Customer table, add a column E\_mail.**

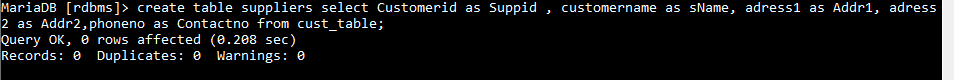
alter table cust\_table add column e\_mail varchar(30);

1. **Drop the E\_mail column from Customer table.**

alter table cust\_table drop column e\_mail;

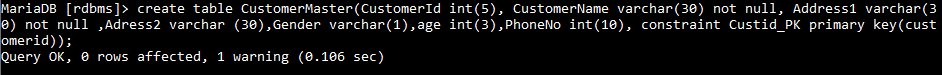
1. **Create the Suppliers table based on the structure of the Customer table. Include only the CustomerId, CustomerName, Address1, Address2, and phoneno columns.**

**Name the columns in the new table as SuppID, SName, Addr1, Addr2, and Contactno respectively.**

create table suppliers select Customerid as Suppid , customername as sName, adress1 as Addr1, adress2 as Addr2,phoneno as Contactno from cust\_table;

1. **Drop the above table and recreate the following table with the name CustomerMaster**

**drop table suppliers;**

create table CustomerMaster(CustomerId int(5), CustomerName varchar(30) not null, Address1 varchar(30) not null ,Adress2 varchar (30),Gender varchar(1),age int(3),PhoneNo int(10), constraint Custid\_PK primary key(customerid));

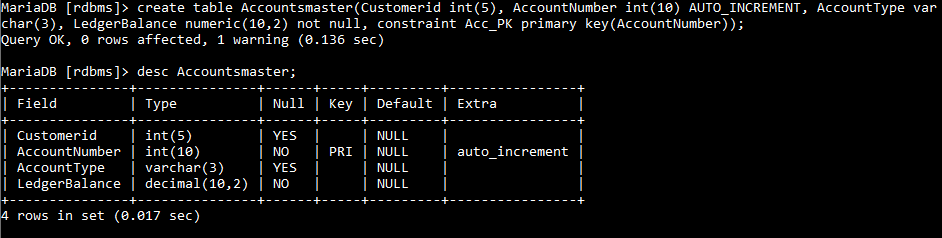
1. **Create the AccountsMaster table with the following Columns. Use auto generate to generate Account number**

**Customerld int(5)**

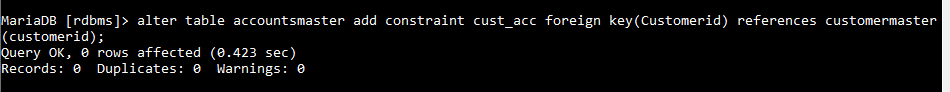
**AccountNumber int(10,2) Primary key(Name of constraint is Acc\_PK)**

**AccountType Char(3)**

**LedgerBalance int(10,2) Not Null**

create table Accountsmaster(Customerid int(5), AccountNumber int(10) AUTO\_INCREMENT, AccountType varchar(3), LedgerBalance numeric(10,2) not null, constraint Acc\_PK primary key(AccountNumber));

1. **Relate AccountsMaster table and CustomerMaster table through Customerld column with the constraint name Cust\_acc.**

alter table accountsmaster add constraint cust\_acc foreign key(Customerid) references customermaster(customerid);

1. **Insert the following rows to the CustomerMaster table:**

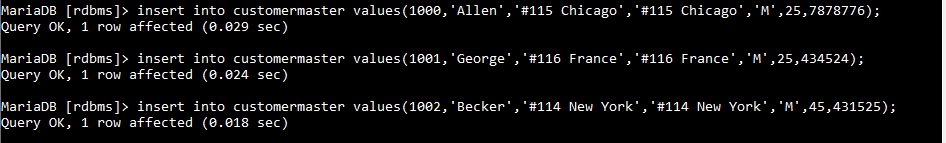
**1000, Allen, #115 Chicago, #115 Chicago, M, 25, 7878776**

**1001, George, #116 France, #116 France, M, 25, 434524**

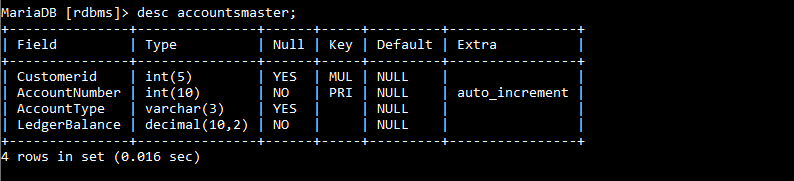
**1002, Becker, #114 New York, #114 New York, M, 45, 431525**

insert into customermaster values(1000,'Allen','#115 Chicago','#115 Chicago','M',25,7878776);

insert into customermaster values(1001,'George','#116 France','#116 France','M',25,434524);

insert into customermaster values(1002,'Becker','#114 New York','#114 New York','M',45,431525);

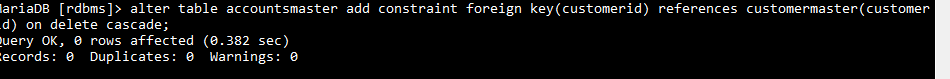
1. **Modify the AccountMaster table with the Check constraint to ensure AccountType should be either NRI or IND.**

alter table accountsmaster add check(Accounttype ='NRI' or accounttype='IND');

1. **Modify the AccountsMaster table keeping a Check constraint with the name Balance\_Check for the Minimum Balance which should be greater than 5000.**

alter table accountsmaster add constraint Balance\_Check check(LedgerBalance>5000);

1. **Modify the AccountsMaster table such that if Customer is deleted from Customer table then all his details should be deleted from AccountsMaster table.**

alter table accountsmaster add constraint foreign key(customerid) references customermaster(customerid) on delete cascade;

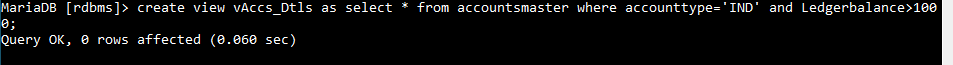
1. **Create Backup copy for the AccountsMaster table with the name ‘AccountDetails’**

create table AccountDetails as select \* from accountsmaster;

1. **Create a view ‘Acc\_view’ with columns Customerld, CustomerName, AccountNumber, AccountType, and LedgerBalance from AccountsMaster. In the view Acc\_view, the column names should be CustomerCode,.**

create view Acc\_view as (select c.customerid as CustomerCode, customername,accountnumber,accounttype,ledgerbalance from cust\_table c inner join accountdetails a on c.customerid=a.customerid);

1. **Create a view on AccountsMaster table with name vAccs\_Dtls. This view should list all customers whose AccountType is ‘IND’ and their balance amount should not be less than 10000. Using this view any DML operation should not violate the view conditions.**

create view vAccs\_Dtls as select \* from accountsmaster where accounttype='IND' and Ledgerbalance>1000;

1. **Create a view accsvw10 which will not allow DML statement against it.**

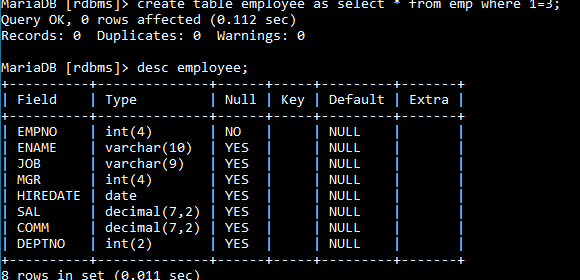
create view accsvw10 as select \* from accountsmaster with check option;



**5 . DATA MANIPULATION LANGUAGE**

1. **Create Employee table with same structure as EMP table.**

create table employee as select \* from emp where 1=3;

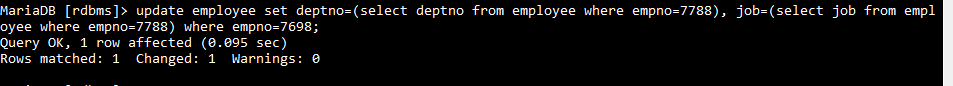
desc employee;

1. **Write a query to populate Employee table using EMP table’s empno, ename, sal, deptno columns.**

insert into employee (empno,ename,sal,deptno) select empno, ename,sal,deptno from emp;

select \* from employee;

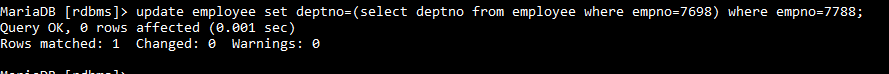
1. **Write a query to change the job and deptno of employee whose empno is 7698 to the job and deptno of employee having empno 7788.**

update employee set deptno=(select deptno from employee where empno=7788), job=(select job from employee where empno=7788) where empno=7698;

1. **Delete the details of department whose department name is ‘SALES’.**

delete from department\_masters where dept\_name='sales';

1. **Write a query to change the deptno of employee with empno 7788 to that of employee having empno 7698.**

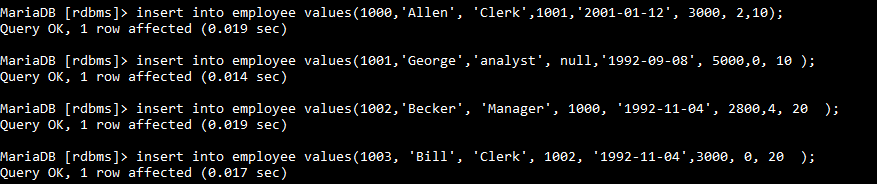
. update employee set deptno=(select deptno from employee where empno=7698) where empno=7788;

1. **Insert the following rows to the Employee table through parameter substitution.**

insert into employee values(1000,'Allen', 'Clerk',1001,'2001-01-12', 3000, 2,10);

insert into employee values(1001,'George','analyst', null,'1992-09-08', 5000,0, 10 );

insert intosa employee values(1002,'Becker', 'Manager', 1000, '1992-11-04', 2800,4, 20 );

insert into employee values(1003, 'Bill', 'Clerk', 1002, '1992-11-04',3000, 0, 20 );

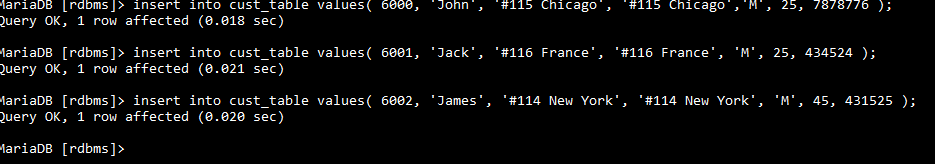
**6.Transaction Control Language Statements**

**1. Insert rows with the following data into the Customer table. 6000, John, #115 Chicago, #115 Chicago, M, 25, 7878776, 10000 • 6001, Jack, #116 France, #116 France, M, 25, 434524, 20000 • 6002, James, #114 New York, #114 New York, M, 45, 431525, 15000.50**

**Use parameter substitution.**

insert into cust\_table values( 6000, 'John', '#115 Chicago', '#115 Chicago','M', 25, 7878776

insert into cust\_table values( 6001, 'Jack', '#116 France', '#116 France', 'M', 25, 434524 )

insert into cust\_table values( 6002, 'James', '#114 New York', '#114 New York', 'M', 45, 431525 );

**2. Create a Savepoint named ‘SP1’ after third record in the Customer table** .

. savepoint sp1;

**3. Insert the below row in the Customer table.**

**6003, John, #114 Chicago, #114 Chicago, M, 45, 439525, 19000.60**

. insert into cust\_table values(6003, 'John', '#114 Chicago', '#114 Chicago', 'M', 45, 439525);



**4. Execute rollback statement in such a way that whatever manipulations done before Savepoint sp1 are permanently implemented, and the ones after Savepoint SP1 are not stored as a part of the Customer table.**

. rollback to sp1;



S