**Assignment**

1. Create a table with the following structure (Table Name – Employee)

Empid integer

Empname varchar(100)

Salary decimal(10,3)

Country varchar(100)

Hiredate Date

Mgrid integer

**Query:** **create table Employee(Empid integer,Empname varchar(100), Salary decimal(10,3),Country varchar(100),Hiredate Date,Mgrid integer);**

2. Insert Few Records into the Table

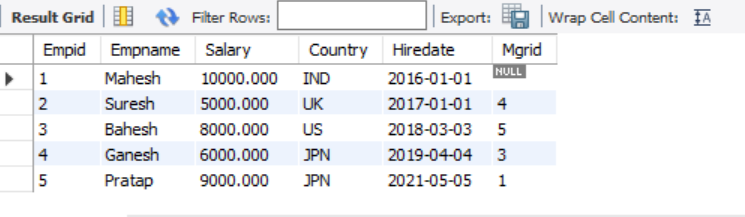
|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 1 | Mahesh | 10000 | IND | 01-01-2016 | NULL |
| 2 | Suresh | 5000 | UK | 01-01-2017 | 4 |
| 3 | Bahesh | 8000 | US | 03-03-2018 | 5 |
| 4 | Ganesh | 6000 | JPN | 04-04-2019 | 3 |
| 5 | Pratap | 9000 | JPN | 05-05-2021 | 1 |

**Query:** **insert into Employee(empid,empname,salary,country,hiredate,mgrid)value**

**(1,'Mahesh',10000,'IND','2016-01-01',NULL),(2,'Suresh',5000,'UK','2017-01-01',4)**

**(3,'Bahesh',8000,'US','2018-03-03',5),(4,'Ganesh',6000,'JPN','2019-0404',3),**

**(5,'Pratap',9000,'JPN','2021-05-05',1);**



3.Update Suresh’s Salary with the Value 5500.

**Query:** **update Employee set salary=5500 where empname='Suresh';**

Graphical user interface, application

Description automatically generated

4. Update hiredate as 01-01-YYYY for Empname with pattern Second Character as ‘a’.

**Query:** **update employee set Hiredate=concat(YEAR(hiredate),'-01-01') where Empname like '\_a%';**

**Graphical user interface, application

Description automatically generated**

5. Delete Record with Country = UK.

**Query:** **delete from employee where Country="UK";**

**Graphical user interface, text, application

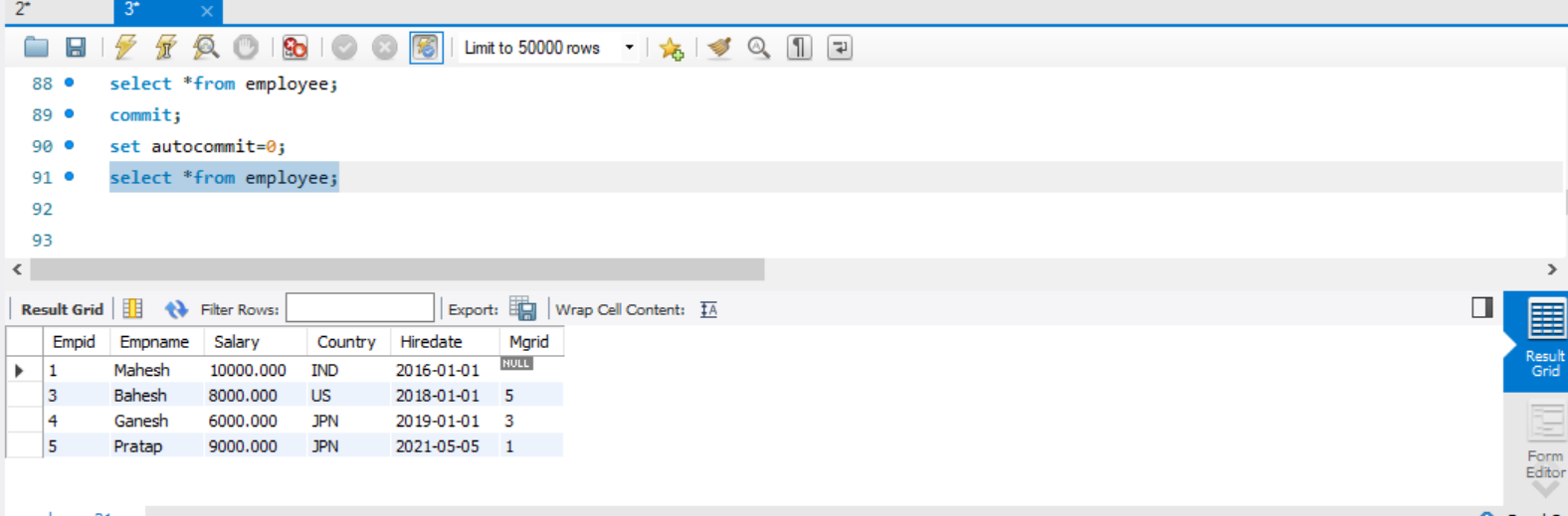
Description automatically generated**

6. Verify how Commit and Rollback behaves with Data Presentation between Multiple SQL sessions.

**Commit:** The COMMIT statement lets a user save any changes or alterations on the current transaction. These changes then remain permanent. Auto-commit mode means that when a statement is completed, the method commit is called on that statement automatically To avoid these we can give set autocommit=0.

**Rollback**:The ROLLBACK statement lets a user undo all the alterations and changes that occurred on the current transaction after the last COMMIT.

**I select all data from my table and I commit it.**



**After commit I update some column on my table**.

Graphical user interface, text, application

Description automatically generated

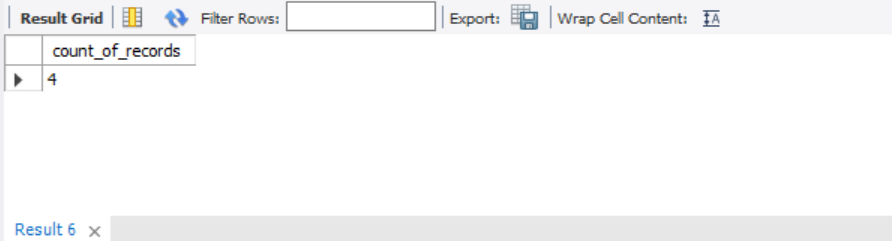
**Now I give rollback and select all data from my table**

Graphical user interface, text, application, email

Description automatically generated

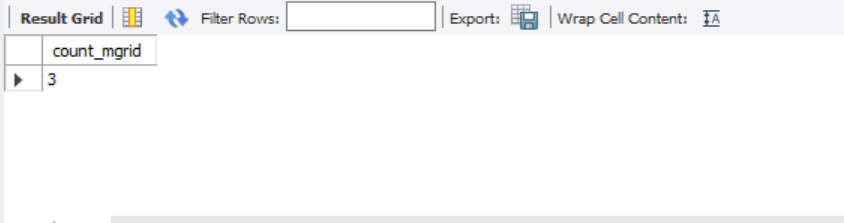
7.Find Count of Records on the Above Table after 4.

**Query:** **select count(\*)as count\_of\_records from employee where empid>4;**

****

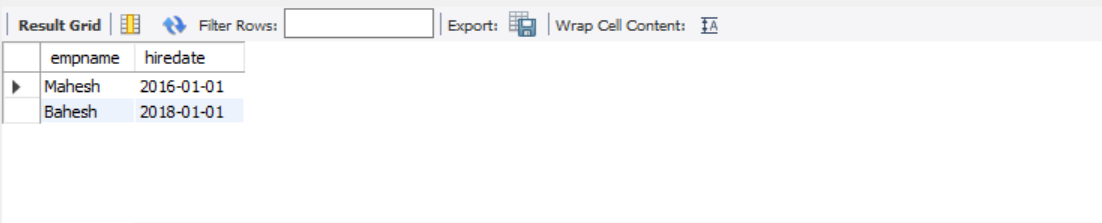
8. Find Count of Records on the Above Table after 4. Based on Mgrid.

**Query:** **select count(mgrid) as count\_mgrid from employee;**

****

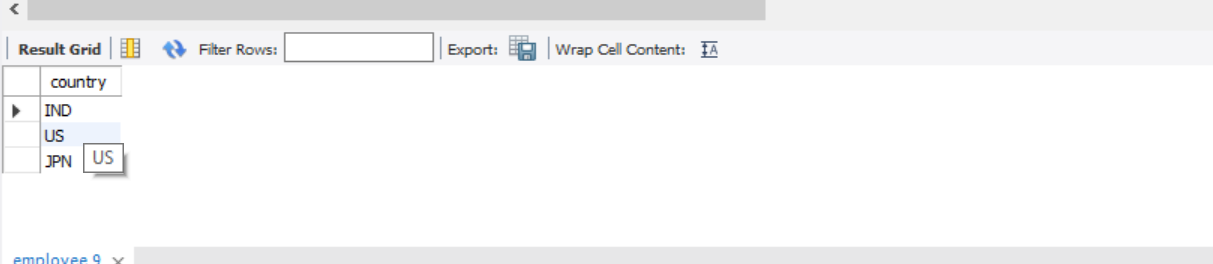
**9.** Retrieve Empname, hiredate for those Employees joined before Jan 2019.

**Query: select empname,hiredate from employee where year(Hiredate)<2019;**



10. Retrieve unique Country values on Employee Table

**Query: select distinct country from employee;**

****

11. Create another table Employee\_New with same structure as Employee but without data.

**Query: create table employee\_new as select \*from employee where 1=0;**

**select \*from employee\_new;**

**12.** Insert Data from Employee into Employee\_New.

**Query: insert into employee\_new(empid,empname,salary,country,hiredate,mgrid)**

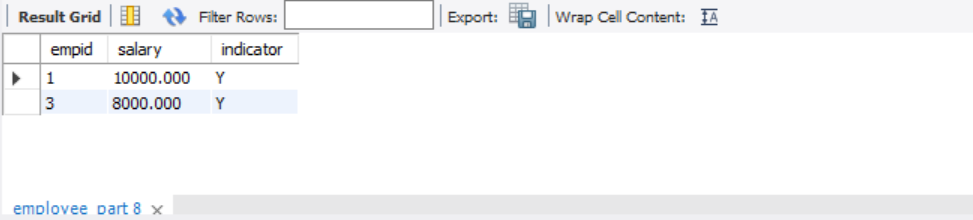
**select empid,empname,salary,country,hiredate,mgrid from employee;**

**select \*from employee\_new;**

13. Create another table Employee\_Part with same structure as Employee only with Empid, Salary columns for those employees based out of JPN.

**Query: create table employee\_part as select empid,salary from employee where country<>"JPN";**

**select \*from employee\_part;**

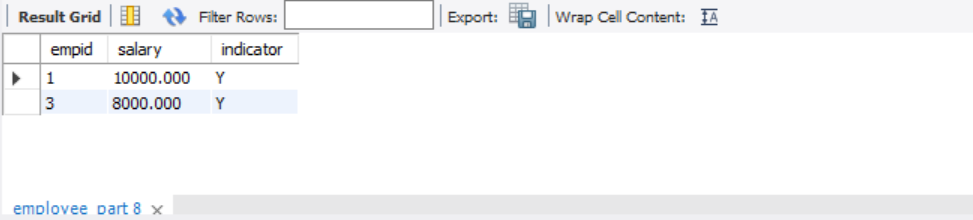
****

14. Add a column Indicator varchar(1) to Employee\_Part with Default value set to Y. Query the Table to see the Data.

**Query:** **alter table employee\_part add indicator varchar(1);**

**update employee\_part set indicator='Y';**

**select \*from employee\_part;**

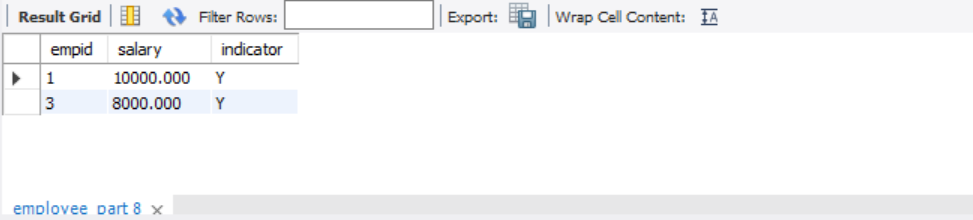
****

**15.** Modify datatype of Salary column to decimal(15,3) on Employee\_Part.

**Query:** **ALTER TABLE employee\_part**

**modify COLUMN salary decimal(15,3);**

**select \*from employee\_part;**

****

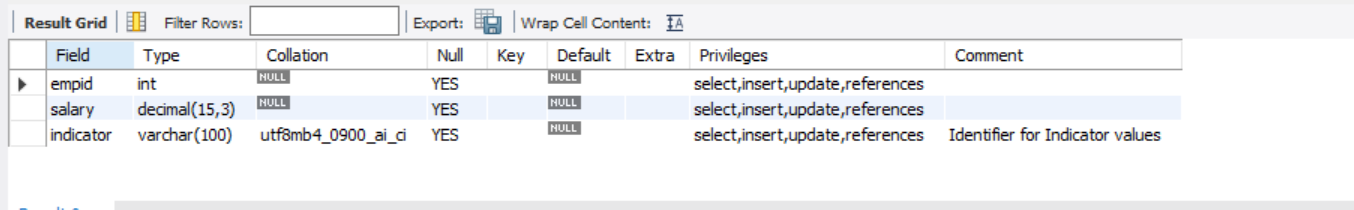
**16.** Add comments on the Indicator column as ‘Identifier for Indicator values’.

**Query: ALTER TABLE employee\_part**

**MODIFY COLUMN indicator**

**varchar(100) COMMENT "Identifier for Indicator values";**

**SHOW FULL COLUMNS FROM employee\_part;**

****

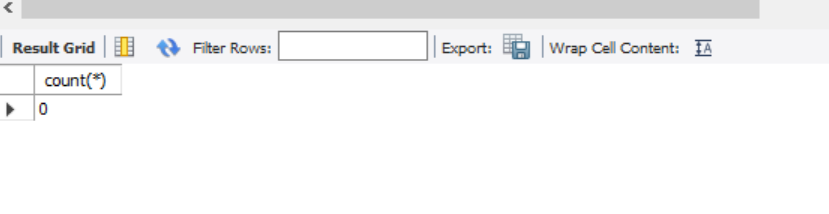
**17.** Add constraint to Indicator column to check for values = Y.

**Query: alter table employee\_part add constraint indicator check(indicator='Y');**

**18.** Truncate the table Employee\_Part and then check count of records on the Table.

**Query:** **truncate table employee\_part;**

**select count(\*) from employee\_part;**

****

**19.** Add a column Indicator varchar(1) to Employee\_Part with Default value set to Y.

**Query: update employee\_part set indicator='Y';**

**20.** Create the Table as follows., and try the below steps.,

Department – Table Name

Deptid integer Primary key

Deptname varchar(100)

Insert Few records into the Department Table

(100, HR. 200, Sales. 300, Marketing. 400 Research. 500 Admin)

**Query: create table department(deptid int primary key ,deptname varchar(20) );**

**insert into department(deptid,deptname)values(100,'HR'),**

**(200,'Sales'),(300,'Marketing'),**

**(400,'Research'),(500,'Admin');**

**select \*from department;**

Employee\_Constraint – Table Name (for data use the above example data on 2.)

Empid integer

country varchar(100)

name varchar(30)

salary decimal(11,2)

deptid integer

Mgrid integer

**Query:** **create table employee\_constraint(Empid integer primary key,**

**country varchar(100),hiredate date,**

**name varchar(30),**

**salary decimal(11,2) not null,**

**deptid integer,**

**Mgrid integer,check(salary>2000),constraint**

**foreign key(deptid) references department(deptid)**

**on delete cascade,constraint unq\_cons unique(name));**

**insert into employee\_constraint(empid,country,name,salary,mgrid,hiredate)**

**(select empid,country,empname,salary,mgrid,hiredate from employee\_new);**

**update employee\_constraint set deptid=100 where name="mahesh" ;**

**update employee\_constraint set deptid=500 where name="bahesh" ;**

**update employee\_constraint set deptid=400 where name="ganesh" ;**

**update employee\_constraint set deptid=300 where name="pratap" ;**

i. Add check constraint on Deptid after table creation to check for Deptid <= 500.

**Query: alter table employee\_constraint add constraint check\_deptid check(deptid<=500);**

**ii.** Disable the above check constraint

**Query: alter table employee\_constraint disable constraint check\_deptid;**

iii. Enable the above check constraint.

**Query: alter table employee\_constraint enable constraint check\_deptid;**

iv. Drop the above check constraint.

**Query: alter table employee\_constraint drop constraint check\_deptid;**

v. Drop the Unique constraint on name.

**Query:alter table employee\_constraint drop constraint unq\_cons**

vi. Recreate the Unique constraint on name

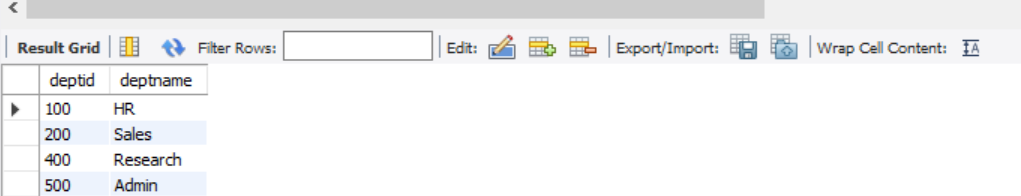
**Query: alter table employee\_constraint add constraint unq\_cons unique(name);**

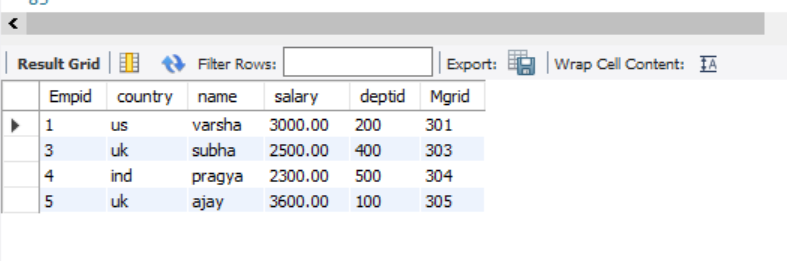
vii. Verify how ON DELETE CASCADE/ON DELETE SET NULL works with Foreign Key

**Query:** **delete from department where deptname="marketing";**

**select \*from department;**

**select \*from employee\_constraint;**

****

****

viii. Drop Primary key constraint on Department Table via CASCADE keyword to check. Employee\_Constraint Table Foreign key constraint also dropped.

**Query: ALTER TABLE department DROP PRIMARY KEY cascade**;

ix. Create Table Level Named check constraint to check for Empid > 0 and Salary > 2000.

**Query: alter table employee\_constraint add constraint check\_cns check(empid>0 and salary>2000);**

x.Drop column Salary in such a way the Dependent constraints are also dropped.

**Query: alter table employee\_constraint**

**drop constraint check\_cns;**

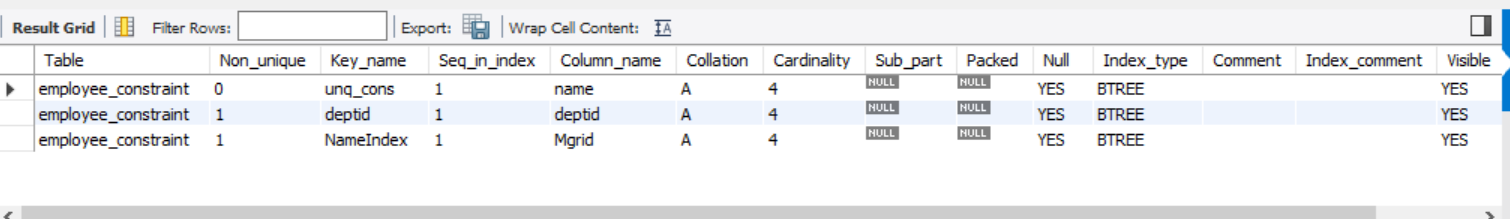
**alter table employee\_constraint**

**drop column salary;**

21. Create Non Unique Index on Mgrid column of Employee\_Constraint table.

**Query: CREATE INDEX NameIndex ON employee\_constraint (mgrid);**

**show index from employee\_constraint;**

****

**22.** Drop the above Index.

**Query: drop index NameIndex on employee\_constraint;**

**23.** Create Synonym for the Employee\_Constraint Table as Emp\_Cons.

**Query: CREATE SYNONYM emp\_constable**

**FOR a.employee\_constraint;**

24. Create Sequence starting with 10, ending with 20, incremented by 5 with cycle.

**Query:CREATE SEQUENCE sequence\_1**

**start with 10**

**increment by 5**

**minvalue 0**

**maxvalue 20**

**cycle;**

25. Create new Table Emp\_Sequence with Empid, Salary column without any constraint. Insert few records into Emp\_Sequence table with sequence generated by the above sequence generator and some random salary values.

**Sequence won’t work with mysql**

26. Create a view with Name Emp\_View based on Employee\_Constraint Table with the criteria below (all New columns for the view).,

* + - Salary + (Salary \*0.5) as New\_Salary
    - Hiredate + 5 days as Timestamp Datatype
    - Calculate no of days between current date and hiredate in Days
    - Concatenate Mr. with the Empname
    - Find out the character position of h (second occurrence from start, second occurrence from the end)
    - Take 2-4 characters on Empname and Display as Emp\_Sub\_Name
    - Justify/Right Pad the Empname with overall 15 characters with Spaces
    - Get length of Empname
    - Round Salary to decimal(10,0)
    - Print Region column as if country belongs to IND, JPN, SRI then ‘ASIA’ else ‘Others’
  + All these operations to be stored as separate columns on the View for those Employees who are not from SRI and Salary > 500.

**Query: create view emp\_view as select new\_salary,timestamp\_datatype,**

**date\_diff,empname,char\_pos,emp\_sub\_name,**

**new\_name,l\_name,round(cast(new\_salary as decimal(10,0))) as r\_salary,region from (**

**select empid,country,salary\*0.5 as new\_salary,**

**adddate(hiredate,5) as timestamp\_datatype,**

**datediff(curdate(),hiredate) as date\_diff,**

**concat('MR ',name) as empname,**

**deptid,mgrid,substr(name,2,4)**

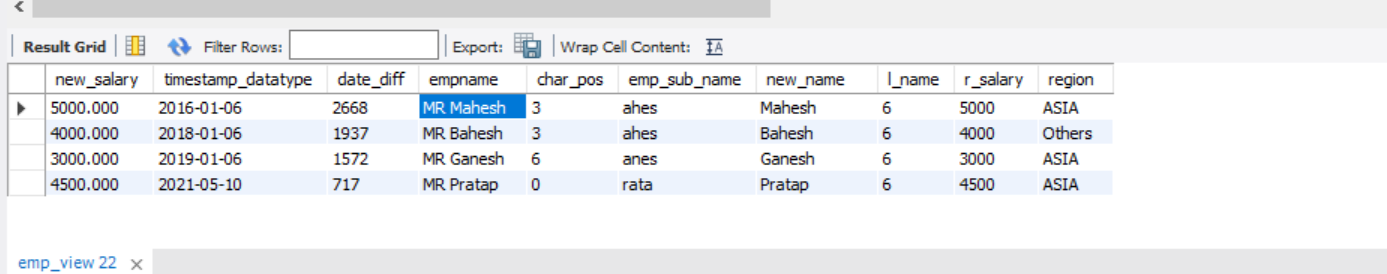
**as emp\_sub\_name,instr(name,'h')**

**as char\_pos,rpad(name,15," ") as new\_name,length(name) as l\_name,**

**case when country in ("IND","JPN","SRI")**

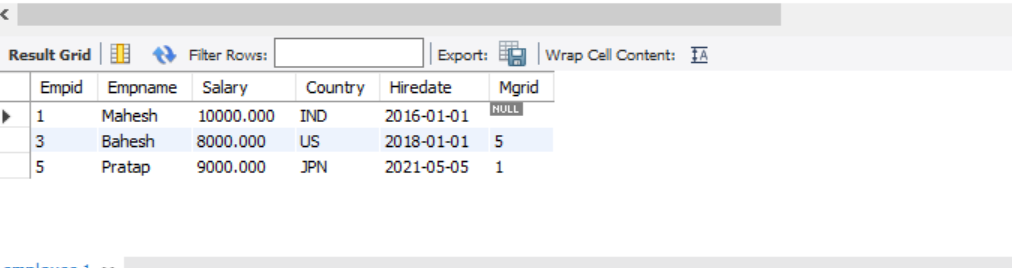
**then "ASIA" else "Others" end as region from employee\_constraint)a where country!="sri";**

**select \*from view;**



**27.** Print records from 2. Whose salary is > min salary of the entire table.

**Query: select \* from employee where salary>(select min(Salary) from employee);**

****

**28.** Print records from 2. Whose salary is < max salary of the table by Country.

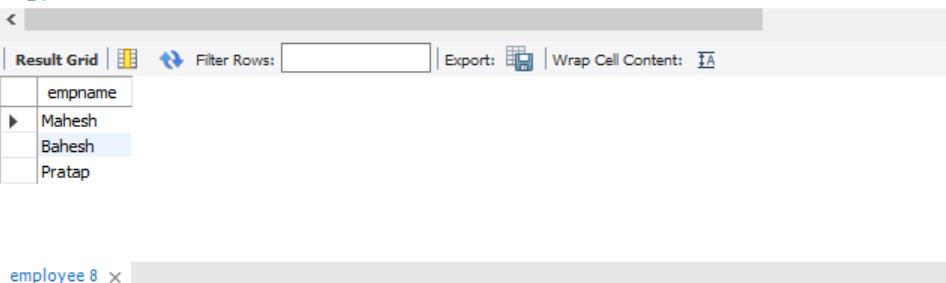
**Query:** **select \* from employee where salary<(select max(Salary) from employee);**

**Graphical user interface, text, application

Description automatically generated**

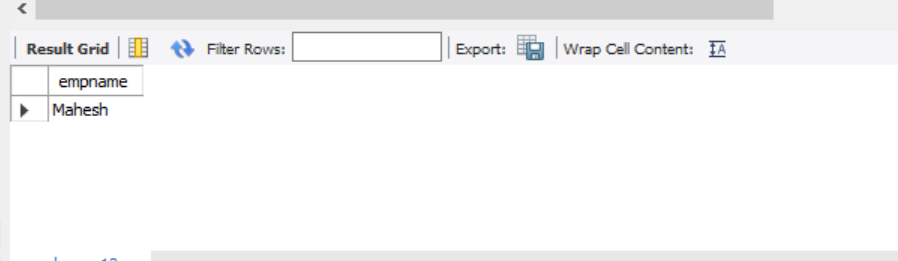
**29.** Print records from 2. Where Employees are also Managers.

**Query:** **select empname from employee where empid in (select mgrid from employee);**

****

30. Print records from 2. Where Employees don’t have any Manager.

**Query: select empname from employee where mgrid is null;**

****

**31.** Refer the Above Table, Create the same and try the below.,

Join (Self, equi,left/right/full -> outer)

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

Empid,empname,deptid,Mgrid deptid, deptname

1,pratap,100,4 100,HR

2,nithin,101,4 101,Sales

3,mahesh,null,1 null,null

4,suresh,104,null 105,MKT

**Query: create table emp(Empid int ,empname varchar(20),deptid int ,Mgrid int);**

**insert into emp(Empid,empname,deptid,Mgrid)values(1,'pratap',100,4),(2,'nithin',101,4),(3,'mahesh',null,1),(4,'suresh',104,null);**

**create table dept(deptid int,dept\_name varchar(20));**

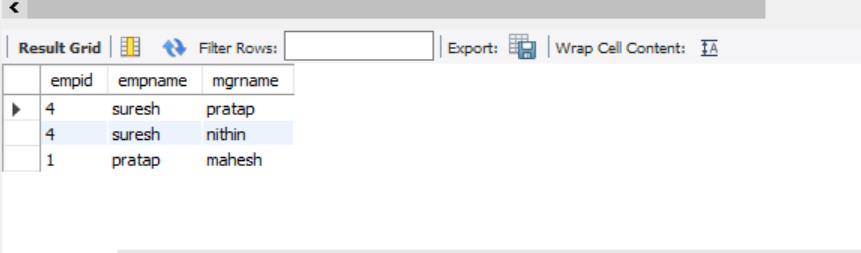
**insert into dept(deptid,dept\_name)values(100,'HR'),(101,'Sales'),(null,null),(105,'MKT');**

i. Self Join Emp with Emp on Empid and Mgrid and get O/p as below.,

Empid,Empname,Mgrname

**Query: select a.empid,a.empname,b.empname as mgrname from**

**emp a join emp b on a.empid=b.mgrid;**

****

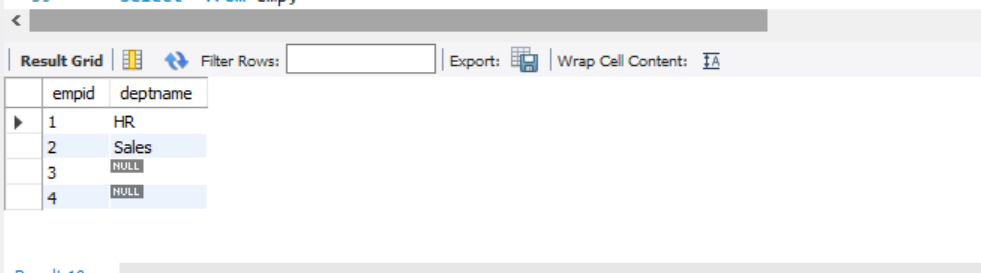
**ii.** Get all records from Emp and matching Deptname from Dept table

Empid, Deptname.

**Query: select e.empid as empid ,d.dept\_name as deptname**

**from emp e left join**

**dept d on d.deptid=e.deptid;**

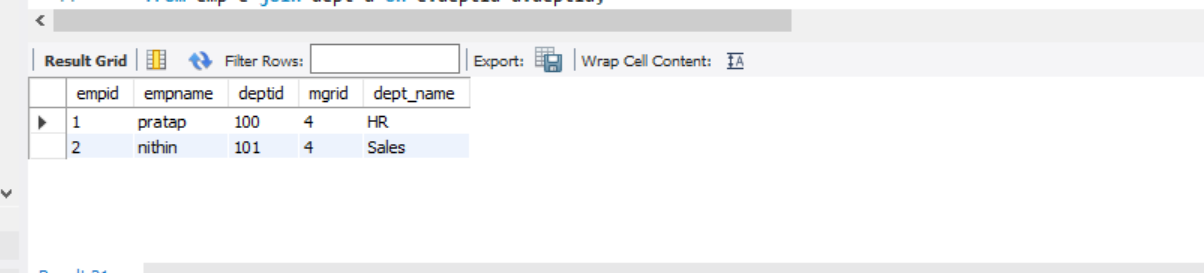
****

iii. Do Inner join and get records from Emp and matching Deptnames from Dept table on Deptid.

**Query: select e.empid,e.empname,e.deptid,e.mgrid,d.dept\_name**

**from emp e join dept d on e.deptid=d.deptid;**

**select \*from employee\_constraint;**

****

32. Insert 3-4 more randon records into Employee\_Constraint table.

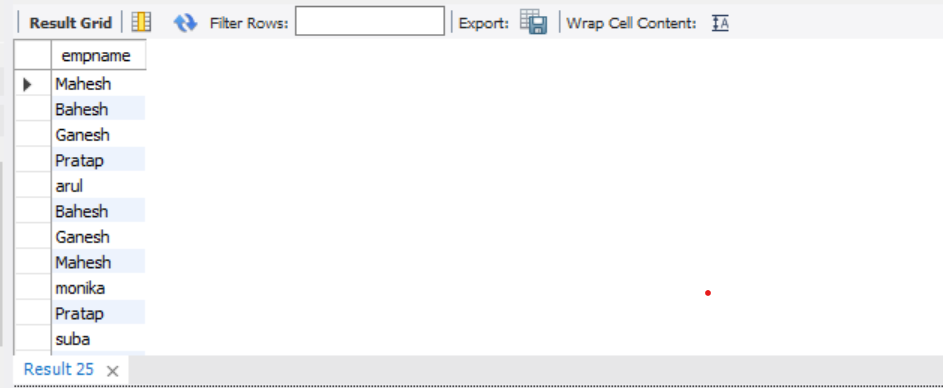
**Query: insert into employee\_constraint(empid,country,hiredate,name,deptid,mgrid)**

**values(6,'IND','2017-05-05','vaishu',500,9),(7,'uk','2015-04-05','monika',400,6),**

**(8,'aus','2019-05-01','suba',300,7),(9,'ing','2013-03-05','arul',200,5);**

**33.** Get all the Empnames from both Employee and Employee\_Constraint table.

**Query:select empname from employee union all select name from employee\_constraint;**



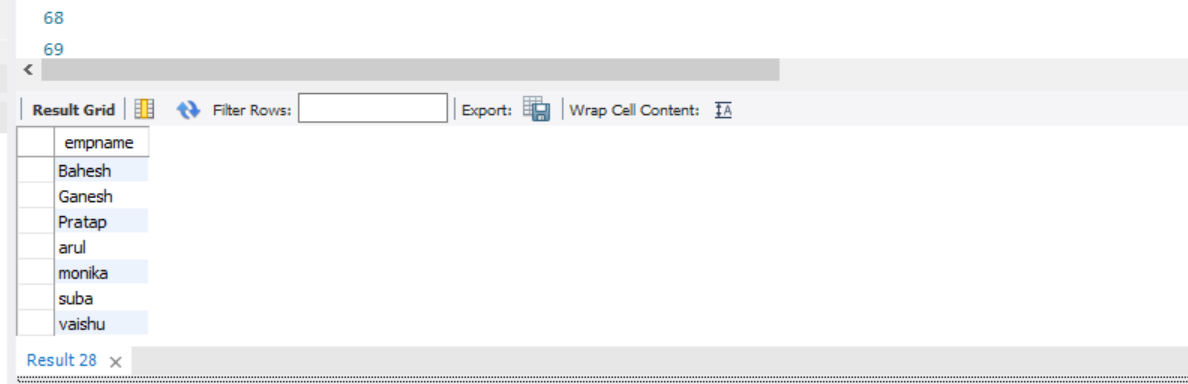
34. Get matching Empnames from both Employee and Employee\_Constraint table.

**Query: select e.empname from employee e join employee\_constraint ec on e.empname=ec.name;**

**In oracle we can also use distinct.**

**35.** Get all the Empnames from both Employee and Employee\_Constraint table with Dups removed.

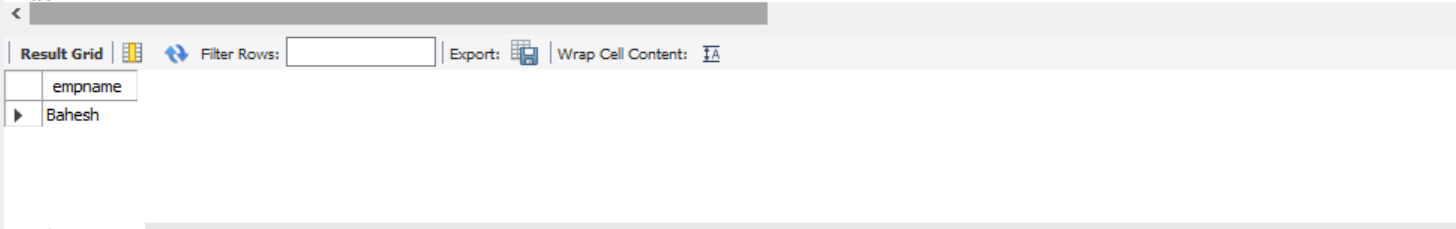
**Query: select empname from employee union select name from employee\_constraint;**

****

**36.** Get Empnames from Employee not present on Employee\_Constraint

**Query: delete From employee\_constraint where name="bahesh" ;**

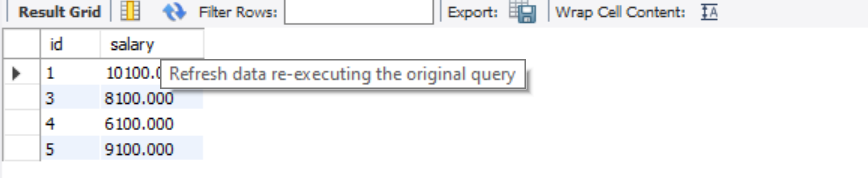
**select empname from employee where empname not in (select name from employee\_constraint);**

****

37. Create TEMP table to hold the results of SELECT ID, SALARY+100 from Employee ;

**Query: create temporary table temp select empid as id,salary+100 as salary from employee;**

**select\*from temp;**

****

**38.** Calculate ROW\_NUMBER, RANK and DENSE\_RANK for every row on Entire Table of Employee\_Constraint based on Salary Desc.

**Query: insert into employee\_constraint values(2,"uk","2017-03-04","divya",9000,300,4 );**

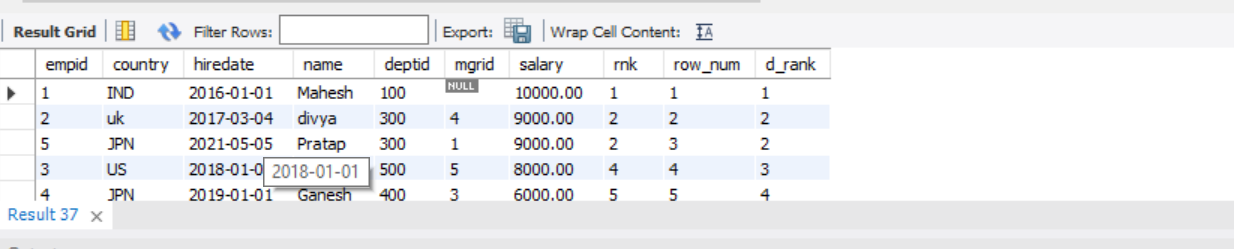
**select empid,country,hiredate,name,deptid,mgrid,salary,**

**rank()over(order by salary desc) as rnk,**

**row\_number()over(order by salary desc) as row\_num,**

**dense\_rank()over(order by salary desc)as d\_rank**

**from employee\_constraint;**



**39.** Calculate ROW\_NUMBER, RANK and DENSE\_RANK for every row on Table Employee\_Constraint based on Salary Desc GROUPed over Country.

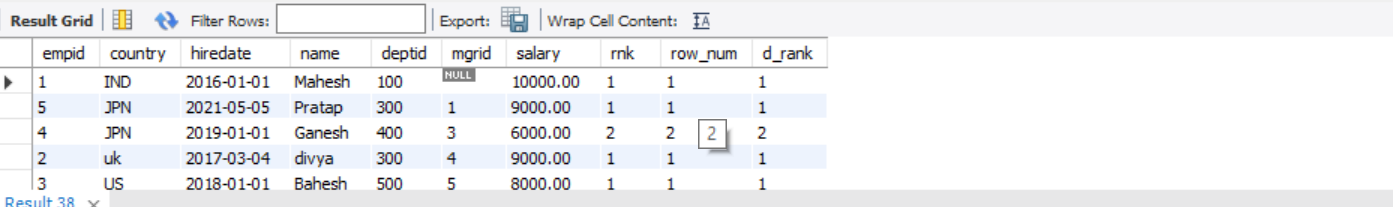
**Query: select empid,country,hiredate,name,deptid,mgrid,salary,**

**rank()over(partition by country order by salary desc) as rnk,**

**row\_number()over(partition by country order by salary desc) as row\_num,**

**dense\_rank()over(partition by country order by salary desc)as d\_rank**

**from employee\_constraint;**

****

**40.** Calculate ROW\_NUMBER for every row on Table Employee\_Constraint based on Salary Desc GROUPed over Country and keep only First Row on the Result set.

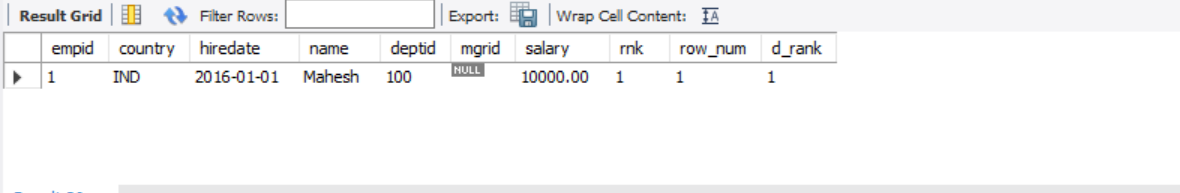
**Query: select empid,country,hiredate,name,deptid,mgrid,salary,**

**rank()over(partition by country order by salary desc) as rnk,**

**row\_number()over(partition by country order by salary desc) as row\_num,**

**dense\_rank()over(partition by country order by salary desc)as d\_rank**

**from employee\_constraint limit 1;**

****

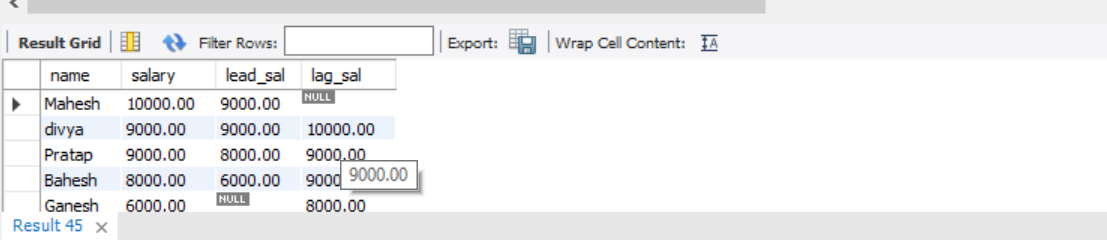
**41.** Calculate min(salary), avg(salary),max(salary) on Table Employee\_Constraint GROUPed over Country.

**Query: select country,min(salary) as min\_sal,max(salary) as max\_sal,avg(salary) as avg\_sal from employee\_constraint group by country;**

**42.** Calculate Lead/Lag on Salary on Table Employee\_Constraint based on Salary Desc.

**Query: select name,salary,lead(salary) over(order by salary desc) as lead\_sal,lag**

**(salary) over(order by salary desc) as lag\_sal from employee\_constraint;**



**43.** Create a dummy table with Empid, Salary and insert Dups for few records into the same. Identify Dups and Clear Dups using ROWID using Correlated Subqueries.

**Query: create table dummy(empid int ,salary int);**

**insert into dummy(empid,salary)values(1,200),(2,200),(1,200),(3,400),(3,400);**

**select \*from dummy;**

**In oracle we can use rowed to remove duplicate but in mysql there is no rowed concept.**

**delete from dummy a where rowid>**

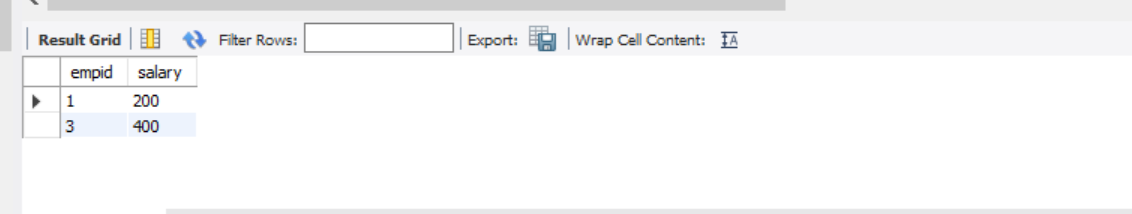
**(select min(rowid) from dummy b**

**where a.empid=b.empid);**

**I use backup table to remove duplicate in mysql .in our table we have empid and salary both have duplicate value so I used backup table.**

**create table dummy\_b as select distinct \*from dummy;**

**select \*from dummy\_b;**

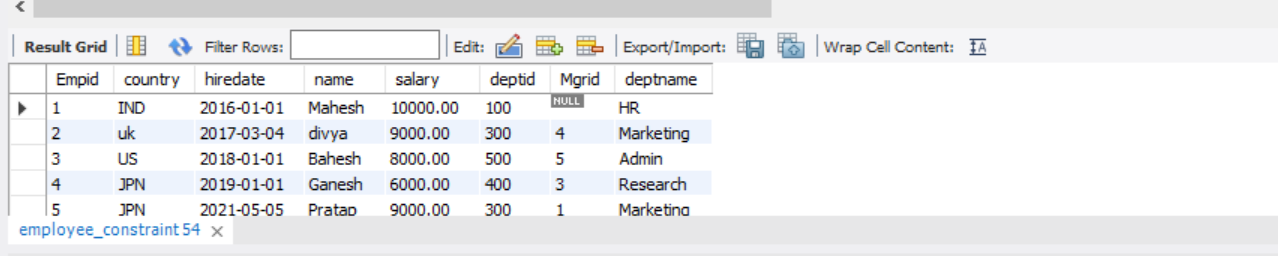
****

**44.** Create a New column Deptname on Employee\_Constraint table. Then Update Employee\_Constraint table Deptname value based on Deptid from its table by looking up against Departments table on the same column getting Deptname from Departments table, using Correlated Subquery.

**Query: alter table employee\_constraint add column deptname varchar(20);**

**select \*from employee\_constraintS;**

**update employee\_constraint e set deptname=(select d.deptname from department d where d.deptid=e.deptid );**

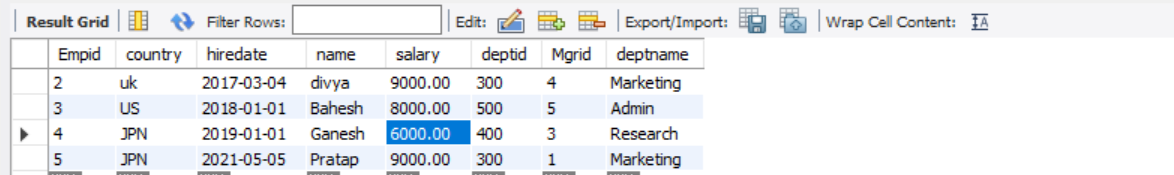
****

**45.** Delete records from Employee\_Constraint table based on its Deptid checked against Departments table on the same column and when Deptname = HR using Correlated Subqueries.

**Query: delete from employee\_constraint where deptid=(select deptid from department where**

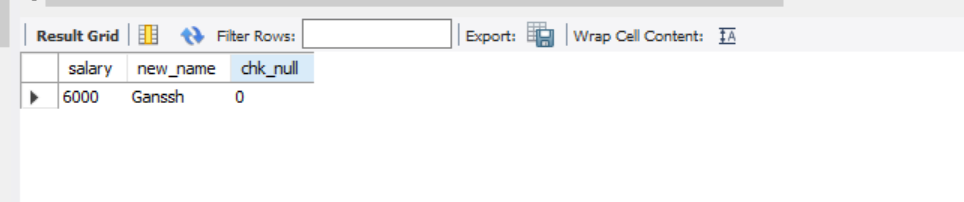
**deptname="hr");**

**select \*from employee\_constraint;**

****

**46.** Practice Single Row Functions, Character conversion Functions, NULL Check functions.

**Query: select round(salary) as salary,replace(name,"e","s") as new\_name,isnull(deptid) as chk\_null from employee\_constraint where name="ganesh";**

****

**47.** Understand difference between IN and EXISTS.