CAPSTONE-1

Basic question till joins:

Q1: Create Table Employee with 6 columns that is employeeid,employeename,gender,managerid,salary,deptno.

Syntax :-

create table Employee

(

Employee\_ID int not null primary key,

Employee\_Name varchar(50) not null check(len(Employee\_Name)<=50),

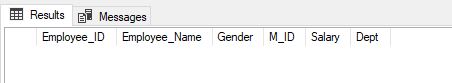
Gender char(2) not null,

M\_ID int foreign key references employee(Employee\_ID) default 'NULL',

Salary int check(Salary>0),

Deptno int not null,

);



Create table department with 3 columns deptno, deptname,city.

create table Department

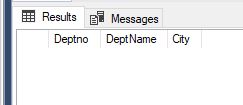
(

Deptno int not null primary key,

DeptName varchar(20) ,

City varchar(20) check(City IN('NOIDA','DELHI' ,'LUCKNOW'))

);



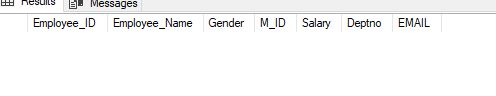
Here choose the appropriate datatype

Q2: add email column in employee table, and change column name city to location in department table.

Email :-

alter table Employee

add EMAIL varchar(30) check(EMAIL like '%\_@\_\_%.\_\_%') default 'NA';



Location :-

alter table Employee

drop constraint FK\_01;

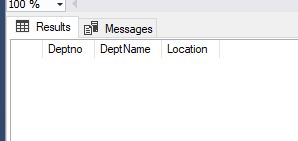
alter table Department

drop constraint PK\_\_Departme\_\_0149C6869243A674;

alter table Department

drop constraint [CK\_\_Department\_\_City\_\_30F848ED];

exec sp\_rename ‘Department.city’,’Location’,’Column => To rename column



Q3: Add following constraints in employee and department table:-

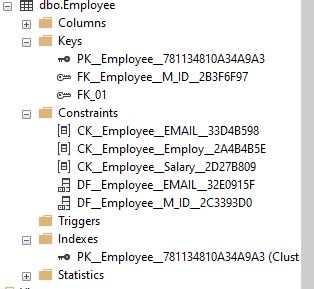
* add primary key to employeeid
* add primary key to department table in department table.
* Add foreign key to deptno in employee table referencing deptno in department table.
* Add foreign key to manager id in employee table referencing employeeid
* Add default constraint to email column where default value is ‘na’.
* Add check constraint to city column that, city can be from Noida,delhi and lucknow only.

City varchar(20) check(City IN('NOIDA','DELHI' ,'LUCKNOW'))

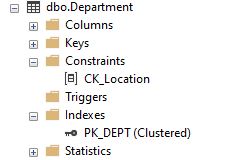
* Add check constraint to salary column, that non negative salary value are not possible.

Salary int check(Salary>0),

Employee Table Primary Key and check constraint:-

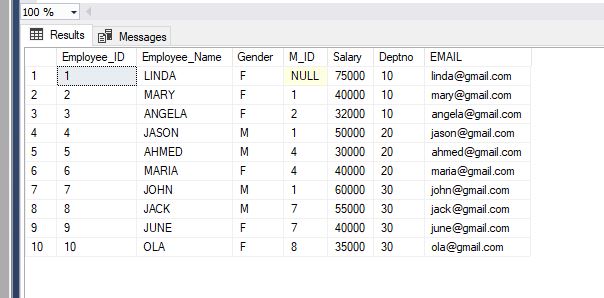


Department Table Primary key and Check constraint:-

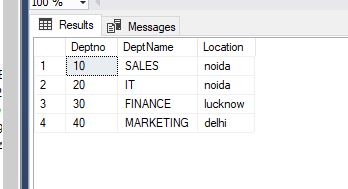


Q4: write insert command to insert the given data into both the tables.

(1,'LINDA','F',10,null,75000),(2,'MARY','F',10,1,40000),(3,'ANJELA','F',10,2,32000),(4,'JASON','M',20,1,50000),(5,'AHMED','M',20,4,30000),(6,'MARIA','F',20,4,40000),(7,'JOHN','M',30,1,60000),(8,'JACK','M',30,7,55000),(9,'JUNE','F',30,7,40000),(10,'OLA','M',30,8,35000)



(10,'SALES','noida'),(20,'IT','noida'),(30,'FINANCE','lucknow'),(40,'MARKETING','delhi')

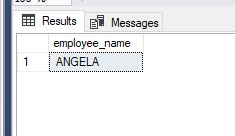


Q5: Write query for following result:

* to get the employees where first and last character of their name are same.

select employee\_name from employee

where RIGHT(employee\_name,1) =LEFT(Employee\_Name,1);



* Find employees where the length of name is greater then the length of employee where eid=5.

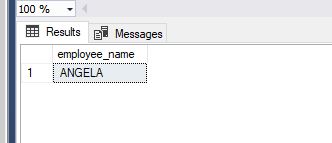
declare @Sname varchar(20),@Lname int

set @Sname=(select Employee\_name from Employee where Employee\_ID=5)

set @Lname=LEN(@Sname)

select employee\_name from employee

where LEN(employee\_name)>@Lname;

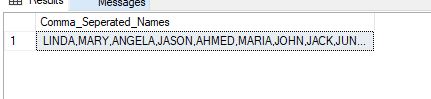


* In the output get all the neame in one row sepearting by comma.

For ex:- abc,bcd,def,fgh…

SELECT STRING\_AGG (CONVERT(NVARCHAR(max),Employee\_Name), ',') AS Comma Seperated Name

FROM Employee;

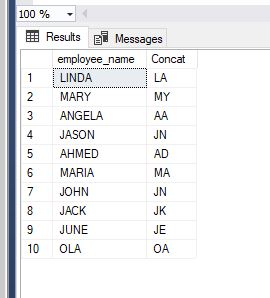


* Find the nickname of all employee by putting first and last character of there name together.

For ex: linda= LA

select employee\_name , CONCAT(left(employee\_name,1) ,right(employee\_name,1))'Concat'

from employee;



Q6: write queries to get the following reults:

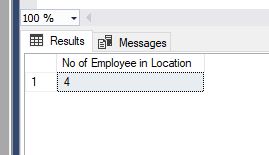
A: find the count of employee who works in lucknow location.

select COUNT(employee\_id)'No of Employee in Location' from Employee a

left join Department b

on a.Deptno = b.deptno

where [location] = 'lucknow';



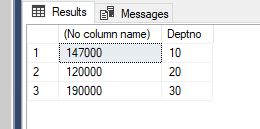
B: find the total salary of employees for each deptno.

select sum(salary),a.Deptno from Employee a

left join Department b

on a.Deptno = b.deptno

group by a.Deptno ;



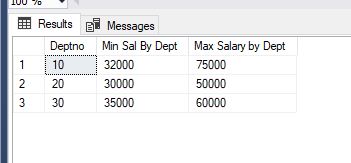
C: Find the least and maximum salaried employee for each department.

select a.Deptno, min(salary)'Min Sal By Dept',MAX(salary)'Max Salary by Dept' from Employee a

left join Department b

on a.Deptno = b.deptno

group by a.Deptno ;



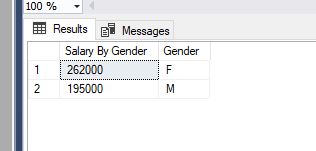
D: what is the total salary for male and female employee.

select sum(salary)'Salary By Gender',a.Gender from Employee a

left join Department b

on a.Deptno = b.deptno

group by a.Gender;



E: how many employee get the salary greater then the average salary of marketing department.

declare @Average\_Of\_M\_Dept int

set @Average\_Of\_M\_Dept = (select a.salary from Employee a left join Department b on a.Deptno=b.Deptno where b.DeptName='Marketing') ----DOUBT

select c.employee\_name , c.salary from Employee c

left join Department d

on c.Deptno=d.Deptno

where c.salary>=@Average\_Of\_M\_Dept;

F: which department have more number of employees then the employees in finance department.

declare @COFD int

set @COFD = (select COUNT(a.Employee\_ID) from Employee a left join Department b on a.Deptno=b.Deptno where b.DeptName='FINANCE')

select d.DeptName , COUNT(c.Employee\_ID)'No of Employees in the Dept' from Employee c

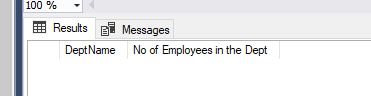
left join Department d

on c.Deptno=d.Deptno

group by Deptname

having COUNT(DeptName)>@COFD

;



Q7: what are the date function you know?

**Date Functions are used to fetch date values and manipulate dates with the help of multiple parameters. Some of the date functions are:-**

1. **GetDate() :-Used to get the current system date that is assigned to the system by the timezone.**

**Syntax :- GETDATE();**

1. **DATEADD():-This date function is used to get a specific date and add or remove certain no of days , months , years from this fetched date.**

**Syntax :- DATEADD(datepart,numbr,date)**

**Datepart :- Datepart :- Date part is the format in which the user wants the result in D/M/YY.**

**Number :-Gives the input for the no of Days/Months/Years that the user wants to add or differ from the fetched date**

**Date:- Date is either the fetched date by the user or is provided by the user manually.**

1. **DATEDIFF():-DATEDIFF returns the no days/months/years between a user provided start date & end date.**

**Syntax:- DATEDIFF(datepart,startdate,enddate)**

**Datepart :- datepart will provide the day ,month ,year in which the user wants the difference between the start date and end date.**

**StartDate:- Start date is the starting date provided by the user.**

**EndDate:-End date is the end date provided by the user.**

1. **Day():-Day function is used to return the day of the fetched date value or provided date value.**

**Syntax:- Day(date)**

**Date :- Date is the input provided by the user.**

1. **Month():-Month function is used to return the month of the fetched or provided date value.**

**Syntax:- Month(Date)**

**Date :- Date is the input provided by the user.**

1. **Year():-Year Function is used to return the year of the fetched or provided date value.**

**Syntax :- Year(Date)**

**Date :-Date is the input provided by the user.**

1. **Datepart():-Returns the desired day,month,year from the provided date input.**

**Syntax:-Datepart(Datepart,DateColumn)**

**Datepart :- Date part is the format in which the user wants the result in D/M/YY.**

**DateColumn:- Date column will the input from the user .**

1. **Convert():-Converts one type of date format to another date format.**

**Syntax :-Convert(DataType,Expression Style)**

**Data Type :- Data type is data type that user wants the date to be in ex Varchar(20)**

1. **Format():-Format date is used to format dates on the basis of day,month,year.**

**Syntax :-FORMAT(date,datepart)**

**Date :-Date is the input provided by the user.**

**Datepart :- Date part is the format in which the user wants the result in D/M/YY.**

1. **EOMONTH():-End of month results in giving the last day of the month selected.**

**Syntax :-EOMONTH(Date)**

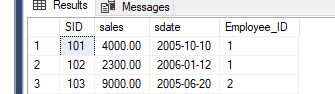
**Date:-Date for which the month end needs to be fetched.**

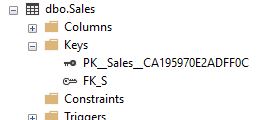
Q8: create table Sales(SID int primary key,sales money,sdate date,EID int)

After table is created, add Primary key on SID and foreign key on EID. This EID should reference Employee table primary key

Insert the records:

insert into Sales values(101,4000,'2005-10-10',1),(102,2300,'2006-01-12',1),(103,9000,'2005-06-20',2),(104,4500,'2007-02-10',2),(105,3200,'2007-02-27',3),(106,2100,'2008-05-25',3),(107,6200,'2008-04-14',4),(108,1900,'2009-10-24',4),(109,9100,'2009-03-20',5),(110,7600,'2010-12-12',5),(111,8300,'2010-09-15',6),(112,4800,'2010-06-23',6),(113,3900,'2010-08-10',7),(114,8100,'2010-11-14',7),(115,4900,'2011-10-19',8),(116,5000,'2011-04-20',8),(117,8400,'2011-05-24',8),(118,2700,'2011-10-12',9),(119,7100,'2012-08-20',9),(120,4100,'2012-03-15',9),(121,2100,'2012-03-15',10),(122,9300,'2012-09-15',10),(123,9200,'2013-07-15',11),(124,8300,'2013-03-15',11)





Q9:

1. Display dname,ename,salary for top 2 employees salary wise only for IT department. Use ranking function

-Using Ranking Function

select distinct top 2 a.Employee\_Name , b.deptname ,a.salary, DENSE\_RANK() over(order by Salary desc)'Ranking By salary'

from Employee a

left join Department b

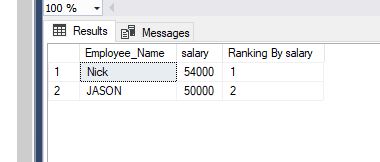
on a.Deptno=b.Deptno

right outer join Sales c

on a.Employee\_ID=c.Employee\_ID

where b.deptname = 'IT'

order by Salary desc;



1. Display top 2 salaried employees from every department. Show Dname, Ename and Salary

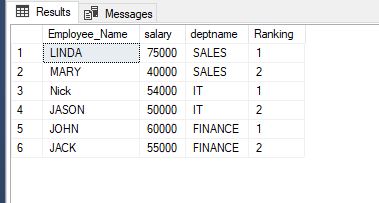
select \* from (select a.Employee\_Name ,a.salary,b.deptname, DENSE\_RANK() over(partition by b.deptno order by Salary desc) Ranking

from Employee a

left join Department b

on a.Deptno=b.Deptno) c

where c.Ranking<=2;



1. Display the department name and total sales done by those department.

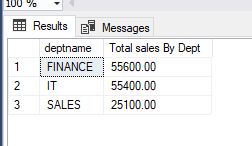
select distinct b.deptname ,sum(sales) over (partition by b.deptno order by b.deptno desc)from

employee a join department b

on a.deptno= b.deptno

join Sales c

on a.Employee\_ID=c.Employee\_ID



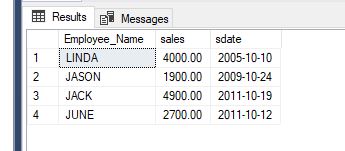
d.Display the name and sales for those employees who do sales in October

select Employee\_Name , s.sales ,s.sdate

from employee a inner join sales s

on a.Employee\_ID = s.Employee\_ID

where datepart(M,s.sdate)=10;



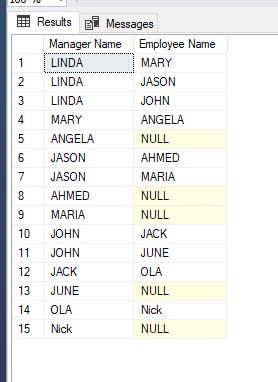
1. Display the name and manager name of all the employees including the name of Top level employee

SELECT m.Employee\_Name as 'Manager Name', e.Employee\_Name AS "Employee Name"

FROM employee E

right JOIN employee M

ON e.m\_id = m.Employee\_ID;



Q10:

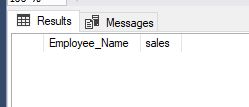
1. Find employees who have not done any sales.

select Employee\_Name , s.sales

from employee a inner join sales s

on a.Employee\_ID = s.Employee\_ID

where s.sales <=0;



1. Find employee who have done more number of sales then the employee who have done 2nd maximum total sale out of all employees.

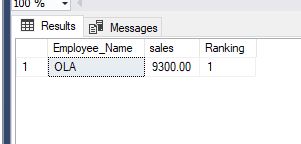
select \* from (select distinct a.Employee\_Name ,s.sales, DENSE\_RANK() over( order by s.sales desc) Ranking

from Employee a

left join sales s

on a.Employee\_ID=s.employee\_ID) c

where c.Ranking<2;



Q11: create view so that column included are employeename,deptname,salary,location. This view should be schemabinded and encrypted.

Create view [dbo].[sample01]

with SCHEMABINDING

as

select a.Employee\_Name,b.DeptName,a.Salary,b.Location

from dbo.Employee a join dbo.Department b

on a.Deptno=b.Deptno

Go

