SQL server

1. create a database ,we are using MSSMS:

-Right click on database and new database,so I created one and name it (emploeemanagement),

-database have 2 files,1.(db.name.mdf):contain database

,tables,etc.2.(db.name.log):contain metadata and important info about the tables and records of the data.

2.starts with table:

We create table in our db , in first column name it :Departmentid

Type: int ,2nd column :Derpartmentname , Type:nvarchar(100)

\*\*nchar:accept only English,nvarchar:accept all languages.\*\*

-then give table name and save ,so always give table :tbDepartmentid

\*\*use tb in beginig\*\*

-now add row in our table , right click and edit top 200 row

So we gave the number and names in columns,but we can make the numbers be written automatically in sql by pk :

In column properties:

Identity speceficition:yes \*to start auto

\*\*if you cant save changes-go to –tools-options-designer-remove the mark from(prevent saving changes require..etc)\*\*

-now create new table for job id and job name,it will be:

Job\_id ,Type:int ,,Job\_name Type:nvarchar(100),

\*\*in job\_name,default value will write developer\*\*,

\*\*default value ,if you leave it null will be written developer\*\*

\*\*don’t forget to set pk for job\_id\*\*

\*\*for edit tables just right click on table and design\*\*

-create table for emploees,emploees\_id Type:int,

Emploees\_name ,Type:nvarchar(100),Hiring\_date,Type:datatime

Salary, Type:decimal(18,4),\*\*use decimal for money\*\*

Departmentid,Type:int,\*\*must be same as departmentid in the other table because when I do relation we don’t get problem so type will be same\*\*

-same for job\_id and Nationality\_id.

\*\*checking which allow null and not and set\*\*

3.make relation between tables:

-Right click on database digram and add them all.

-drag fk to pk ,then right click on the line to see properties from tables and columns.\*\*drag each column to its pk table.\*\*save.\*

4.create table from sql ,go to new query:

create table tbcities

(

City\_id int primary key identity(1,1),

City\_name nvarchar(100) not null

)

\*\*this how to create table from sql,but usually use the default one\*\*

So add a column in employee table for City\_id and do the relation in the diagram . as previous process.

5.get the data by sql :

select City\_name , City\_id \*\*culomn’s name\*\*

from tbcities \*\*table’s name\*\*

select \* //\* means all //

from tbjobs

select \* from tbDepartmentid

where Departmentid=3 \*\*getting specific data from table\*\*\*we can use < > to get all bigger or smaller values\*\*\*where used always for specific data\*\*

select \* from tbjobs

where job\_name ='manager'

\*\*if you get any problem in Arabic language go to database and right click

Go to option>>collation then select Arabic format.\*\*

\*\*you can use like for all column starts with specific word ex:

select \*

from tbcities

where City\_name like'ksa%'\*\*it will show which just started with ksa\*\*

select \*

from tbcities

where City\_name like'%ksa'\*\*it will show which just end with ksa\*\*

select \*

from tbcities

where City\_name like'%ksa%'\*\*it will show all columns have ksa \*\*

select \*

from tbcities

where City\_id=2

or City\_name like'india%'\*\*I can use it with (and) also\*\*

\*\*this will show id =2 and all columns starts with india\*\*

select \* from tbcities

where City\_id != 1 \*\*it will show all except pk =1\*\*

select \* from tbcities

where City\_name like '%ksa%' \*\*it shows all ksa\*\*

select \* from tbcities

where City\_name not like '%ksa%' \*\*it shows all coulmns except those have ksa\*\*

select \* from tbcities

where City\_id in (1,3,5) \*\*it shows selected values\*\*

select \* from tbcities

where City\_id not in (1,3,5) \*\*it shows all except selected values\*\*

select \* from tbcities

where City\_id between 1 and 5 \*\*just show from range 1 to 5\*\*

select \* from tbcities

where City\_name like 'ksa%'

or City\_name like 'india%' \*\*search for to values\*\*

5.sort data :

select \* from tbcities \*\*it show alphabetical order\*\*

order by City\_name

select \* from tbcities \*\*it show revers order\*\*

order by City\_name desc

select \* from tbcities

where City\_id between 1 and 3

order by City\_id desc

6.Insert data:

insert into tbcities

values ('aden') \*\*inserting data\*\*

select \* from tbemploee

where City\_id=3 \*\*taking from selected city\*\*

select \* from tbemploee

where salary > 5000 \*\*who their salary more than 5000\*\*

select top 2 \* from tbemploee \*\*fisrt 2 emploees\*\*

select top 2 \* from tbemploee

order by Emploee\_id desc \*\*last 2 emploees\*\*

select top 1 \* from tbemploee \*\*heist salary\*\*

order by salary desc

7.update and edit the data;(most important in update ‘’’where’’’)

update tbemploee \*\*edit and update data\*\*

set Salary=2000

where Emploee\_id=3

update tbemploee \*\*updated all emploees salary in department 4\*\*

set Salary=5000

where Departmentid=4

update tbemploee \*\*added 500 for all emploees their salary 9000\*\*

set Salary=9500

where Salary=9000

update tbemploee \*\*updated 2 values for one employee\*\*

set Salary=9500 , Work\_side='google'

where Departmentid=1

\*\*you can edit more than 2 values just type comma and write value\*\*

8.Delete data :((very important ‘’’where’’’)))

delete from tbemploee \*\*delete employee\*\*

where Emploee\_id=6

delete from tbDepartmentid

where Departmentid=4

\*\*it won’t delete because some emoploees related to this department\*\*

\*\*if you delete department nobody related to it accept\*\*

9.search for data (more than table),join:

select Empolee\_name,Departmentid from tbemploee \*\*it shows name and Depatment\_id only\*\*

select Empolee\_name,Departmentname

from tbemploee,tbDepartments

where tbemploee.Departmentid=tbDepartments.Departmentid

\*\*this is call join ,we take data from 2 tables\*\*

\*\*if you don’t use join it will show repeated data\*\*

select Empolee\_name,Departmentname,Job\_name

from tbemploee,tbDepartments,tbjobs

where tbemploee.Departmentid=tbDepartments.Departmentid and

tbemploee.Job\_id=tbjobs.Job\_id

\*\*it shows 3 columns for name,department name,job name\*\*

\*\*we use this because we take data from another table\*\*

--here is another way((join)): (inner):\*\*it gets joint data(common)

select Empolee\_name ,Departmentname,Job\_name,Salary

from tbemploee join tbDepartments

on tbemploee.Departmentid=tbDepartments.Departmentid

join tbjobs

on tbemploee.Job\_id=tbjobs.Job\_id

where Salary > 5000 \*\*condition for salary\*\*

\*\*clear way to do join\*\*

select \* from tbemploee join tbDepartments

on tbemploee.Departmentid=tbDepartments.Departmentid

\*\*it shows repeated department\_id for the employee and department and there is some departments not showing because not approved the condition ,no employees in those departments\*\*

\*\*we can use right –left join that show all data even not joint, means of right that one the right hand of join see the ex bellow:\*\*

select \* from tbemploee right join tbDepartments \*\*explained above\*\*

on tbemploee.Departmentid=tbDepartments.Departmentid

\*\*it shows all joint data and not joint data for in right table\*\*

\*\*we can use the left join else\*\*

10.back up and restore for database:

Right click on database>tasks>back up>remove and add file path>then save .bak

\*\*extension must be bak ex,empolee.bak\*\*

-restore database: right click on database>restore>device>add file >choose destination for back up must be new name database for the restoration > ok

11.join while bridge:

\*\*relation right click>relation>add >select the columns\*\*

We created two tables,courses,category and did the relations between category\_id in tbcourses and category\_id in tbcategory

And we created other tables and did relations for all ,then we go to database digram and add the digram and save((we already did the relation just we do this for save it))

\*\*here it shows students names and their courses,we used 3 tables\*\*

select Student\_name,Course\_name from

tbstudents join tbstudentscourses

on tbstudents.Student\_id=tbstudentscourses.Student\_id

join tbcourses

on tbcourses.Course\_id=tbstudentscourses.Course\_id

>>so we want to show up all students even whom not taken courses follow the steps:

select Student\_name,Course\_name from

tbstudents left join tbstudentscourses

on tbstudents.Student\_id=tbstudentscourses.Student\_id

left join tbcourses

on tbcourses.Course\_id=tbstudentscourses.Course\_id

\*\*we used 2 left because the 2nd join take the joint data only so we added anther left to 2nd join \*\*

>>then I want to show only students who don’t sign in any course :

select Student\_name,Course\_name from

tbstudents left join tbstudentscourses

on tbstudents.Student\_id=tbstudentscourses.Student\_id

left join tbcourses

on tbcourses.Course\_id=tbstudentscourses.Course\_id

where Course\_name is null

\*just add (where Course is null)\*\*

>>I want to show courses nobody sign in it:

select Student\_name,Course\_name from

tbstudents join tbstudentscourses

on tbstudents.Student\_id=tbstudentscourses.Student\_id

right join tbcourses

on tbcourses.Course\_id=tbstudentscourses.Course\_id

where Student\_name is null

\*\*same as previous but for courses\*\*

select \*

from (select Course\_name from tbcourses) as miniCourses

\*it just shows small table that I name miniCourses\*

select \*

from (select \* from tbcourses

where Total\_hours > 30) as mini

\*\*always modified with where\*\*

select Student\_name ,Course\_name,Total\_hours+2 as number

from tbstudents ,tbcourses

\*we can use some tables values plus something and give it new column as number\*

select Student\_name ,'course name'+Course\_name as new\_cours

from tbstudents ,tbcourses

\*we added some value to the output ,must give column name\*\*

select Student\_name + ' '+ Course\_name as new\_cours

from tbstudents ,tbcourses

\*we make 2 column in 1 column and give it name, don’t forget spaces\*

12.use (sum-max-min-avg-count-group by):

select sum(Total\_hours)as all\_Hours from tbcourses

\*added total of hours from the column\*

select max(Total\_hours)as maximum from tbcourses

\*shows heist number of hours\*

select min(Total\_hours)as smallest from tbcourses

\*shows smallest no. of hours\*

select avg(Total\_hours)as average from tbcourses

\*it shows the average of hours\*

select count(Total\_hours)as counts from tbcourses

\*it shows no. of courses in the coulmns\*

-I can make them all in one time:

select sum(Total\_hours)as all\_Hours,

max(Total\_hours)as maximum,

min(Total\_hours)as smallest ,

avg(Total\_hours)as average ,

count(Total\_hours)as counts

from tbcourses

select sum(Total\_hours)as all\_Hours,

max(Total\_hours)as maximum,

min(Total\_hours)as smallest ,

avg(Total\_hours)as average ,

count(Total\_hours)as counts,

(select count(\*) from tbstudents)as studentsCount

from tbcourses

\*I can add select to select \*

select statistic.all\_Hours +500 as new\_sum from (

select sum(Total\_hours)as all\_Hours,

max(Total\_hours)as maximum,

min(Total\_hours)as smallest ,

avg(Total\_hours)as average ,

count(Total\_hours)as counts,

(select count(\*) from tbstudents)as studentsCount

from tbcourses) as statistic

\*we can modified as much we want\*

-aggregate:

select count(tbcourses.Course\_id)as student\_Co\_count,student\_name

from tbcourses join tbstudentscourses

on tbcourses.Course\_id=tbstudentscourses.Course\_id

join tbstudents

on tbstudents.Student\_id=tbstudentscourses.Student\_id

group by Student\_name\*here shows students name and no. of courses\*

\*always when you use aggregate the group by must be the after prackets (student\_name)\*

select sum(tbcourses.Total\_hours)as student\_total\_hr,student\_name

from tbcourses join tbstudentscourses

on tbcourses.Course\_id=tbstudentscourses.Course\_id

join tbstudents

on tbstudents.Student\_id=tbstudentscourses.Student\_id

group by Student\_name

\*it shows no. of hours been studied and student name\*

select sum(tbcourses.Total\_hours)as student\_total\_hr,student\_name,tbstudents.Student\_id

from tbcourses join tbstudentscourses

on tbcourses.Course\_id=tbstudentscourses.Course\_id

join tbstudents

on tbstudents.Student\_id=tbstudentscourses.Student\_id

group by Student\_name,tbstudents.Student\_id

\*here we added student\_id\*

\*we must add the select in group by\*

select sum(tbcourses.Total\_hours)as student\_total\_hr,student\_name,tbstudents.Student\_id

from tbcourses join tbstudentscourses

on tbcourses.Course\_id=tbstudentscourses.Course\_id

join tbstudents

on tbstudents.Student\_id=tbstudentscourses.Student\_id

group by Student\_name,tbstudents.Student\_id

having sum(tbcourses.Total\_hours) > 140

\*in aggregate we use for condition (having) instead of where\*

\*must take all the line like showing up\*

select count(s.Student\_id), c.course\_name

from tbstudents as s join tbstudentscourses as sc

on s.Student\_id=sc.Student\_id

join tbcourses as c

on c.Course\_id=sc.Course\_id

group by c.Course\_name

\*it shows no.of students having this course\*

\*we used shortcuts\*

select count(s.Student\_id), c.course\_name

from tbstudents as s join tbstudentscourses as sc

on s.Student\_id=sc.Student\_id

join tbcourses as c

on c.Course\_id=sc.Course\_id

where c.Course\_id=1

group by c.Course\_name

\*we used where because isn’t aggregate it just normal condition\*

13.use (union – union all);

select Student\_id from tbstudentscourses

union

select student\_id from tbstudents

\*it shows data non-recurring only\*

select Student\_id from tbstudentscourses

union all

select student\_id from tbstudents

\*it shows data even repeated one\*

\*\*for union and union all to work,the number ,Data type,and the order of the columns in the select statements should be same\*\*

14.Create view :

\*we use this view instead of repeating the code everytime and we save it in the database\*

create view vwstudentscourses

as

select Student\_name , Course\_name ,tbstudents.Student\_id

from tbstudents join tbstudentscourses

on tbstudents.Student\_id=tbstudentscourses.Student\_id

join tbcourses

on tbcourses.Course\_id=tbstudentscourses.Course\_id

\*and then just type like below\*:

select \* from vwstudentscourses

\*this shortcut will show all the process up\*

\*you can even use (where) with it\*

drop view vwstudentscourses

\*this how to delete it\*

\*you can see your view from the database by right click on views\*

\*you can create even view from database right click\*

15.use(len-replace-getdate()-datediff())

select Student\_name, len (Student\_name)as Letters , Course\_name ,tbstudents.Student\_id

from tbstudents join tbstudentscourses

on tbstudents.Student\_id=tbstudentscourses.Student\_id

join tbcourses

on tbcourses.Course\_id=tbstudentscourses.Course\_id

\*to count the letters just add len before the coulmn\*

select Student\_name ,replace(Course\_name , 'project','pp') ,tbstudents.Student\_id

from tbstudents join tbstudentscourses

on tbstudents.Student\_id=tbstudentscourses.Student\_id

join tbcourses

on tbcourses.Course\_id=tbstudentscourses.Course\_id

\*here replace part of string by replace\*

select day(GETDATE())

\*you can use day-month-year\*

select datediff(year,'06-27-1994',GETDATE())

\*calculate the date\*

££.how to copy data from database to database:

Go to database >right click>tasks>generate scripts>select new query window>advanced>Types of data to script:schema and data,then select the database you want to copy in and execute