Create table in sql

CREATE TABLE user\_details (

userid int NOT NULL IDENTITY(1,1),

username varchar(255) DEFAULT NULL,

first\_name varchar(50) DEFAULT NULL,

last\_name varchar(50) DEFAULT NULL,

gender varchar(10) DEFAULT NULL,

passwords varchar(50) DEFAULT NULL,

isstatus tinyint DEFAULT NULL,

PRIMARY KEY(userid)

)

Insert into table

INSERT INTO user\_details(userid, username, first\_name, last\_name, gender, passwords, isstatus) VALUES

(1, 'rogers63', 'david', 'john', 'Female', 'e6a33eee180b07e563d74fee8c2c66b8', 1)

Or

INSERT INTO user\_details VALUES

(1, 'rogers63', 'david', 'john', 'Female', 'e6a33eee180b07e563d74fee8c2c66b8', 1)

Constraints

Constraints is limitation in table like rule of insert data in table

Primary key

Primary key is mix up of both unique and not null

Unique

Index

**Default** default value if something not enter in column then auto put default value

**Check**

CREATE TABLE user\_details (

userid int NOT NULL IDENTITY(1,1),

username varchar(255) DEFAULT NULL,

first\_name varchar(50) DEFAULT NULL,

last\_name varchar(50) DEFAULT NULL,

gender varchar(10) DEFAULT NULL,

passwords varchar(50) DEFAULT NULL,

isstatus tinyint DEFAULT NULL check (isstatus >= 1),

PRIMARY KEY(userid)

)

Not null etc…

Delete data from table

Delete from <table name> where id = unique id

delete from user\_details where userid = 1

Truncate the table data

Truncate table <table name>

truncate table user\_details

Update the row data

Update <table name> set <column name> = 5000 where id = 2

update user\_details set first\_name = 'mangal' where userid = 1

**selec t command**

select username, first\_name from user\_details

where clause use

select username, first\_name from user\_details where username = ‘mangal singh’ or username = ‘rakesh yadva’

select \* from user\_details order by username asc

select \* from user\_details order by username desc

**Alter command**

DROP Constraints

Alter table table name drop constraint constraint name

alter table user\_details drop constraint DF\_\_user\_deta\_\_first\_\_4E88ABD4

again add constraint with alter

alter table order\_details add foreign key(userid) references user\_details(userid)

**change the database name**

alter database database name modify name = new database name

alter database users modify name = new\_users

or with store procedure

execute sp\_renamedb ‘database name’ ‘new database name’

execute sp\_renamedb 'users' 'new\_users’

change table name

execute sp\_rename ‘table name’ ‘new table name’

execute sp\_rename 'user\_details' 'new\_user\_details'

add new colum in table

alter table table name add column name data type

alter table user\_details add age int

delete the column

alter table user\_details drop column age

change the datatype

alter table table name alter column name data type

alter table user\_details alter column username nvarchar(255)

add not null constraint

alter table user\_details alter column username nvarchar(255) not null

add unique constraint

alter table user\_details add unique(userid)

drop unique constraint

alter table user\_details drop constraint UQ\_\_USER\_DE\_454353453

add primary key constraint

which colum you want to add primary key must have not null constraint too then apply primary key on it

alter table user\_details add primary key(userid)

alter table user\_details alter column userid int not null

drop primary key

alter table user\_details drop constraint PK\_\_USER\_DE\_454353453

add foreign key constraint

foreign key depend on primary key

alter table order\_details add foreign key(userid) references user\_details(userid)

drop foreign key constraint

alter table order\_details drop constraint FK\_ORDER\_DE\_34243242

add check constraint

alter table order\_details add check(user\_age >= 18)

drop check constraint

alter table order\_details drop constraint CK\_ORDER\_DE\_34243242

add default constraint

alter table order\_details add default 18 for user\_age

drop default constraint

alter table order\_details drop constraint DF\_ORDER\_DE\_34243242

ALIAS

Alias mean change the name of column for display some with proper name if I put name as some code words like std\_id

Select std\_id as Student\_id from Student\_table

This called alias for show someone as proper name

**Joint**

**Sql join used to combined two or more table data in a result set**

**Inner join**

create table emp\_tbl(emp\_id int unique not null, emp\_name nvarchar(max) not null, email nvarchar(max), designation nvarchar(max) not null)

select \* from emp\_tbl

insert into emp\_tbl values(1,'mangal singh','mangalsingh@gmail.com', 'manager')

insert into emp\_tbl values(2,'bipin singh','bipin@gmail.com', 'it incharg')

insert into emp\_tbl values(3,'dm singh','dm@gmail.com', 'assitant manager')

insert into emp\_tbl values(4,'w singh','w@gmail.com', 'computer operator')

create table dep\_tbl(dep\_id int unique not null, dep\_name nvarchar(max) not null, dep\_salary nvarchar(max) not null, emp\_id int unique not null)

select \* from dep\_tbl

insert into dep\_tbl values(1,'IT','50000', 3)

insert into dep\_tbl values(2,'Admin','80000', 4)

insert into dep\_tbl values(3,'Account','35000', 1)

insert into dep\_tbl values(4,'HR','20000', 2)

start joining

select \* from emp\_tbl as a inner join

dep\_tbl as b on

a.emp\_id = b.emp\_id

select a.emp\_id, emp\_name, email,designation,dep\_name, dep\_salary from emp\_tbl as a inner join

dep\_tbl as b on

a.emp\_id = b.emp\_id

LEFT Join

create table emp\_tbl(emp\_id int unique not null, emp\_name nvarchar(max) not null, email nvarchar(max), designation nvarchar(max) not null)

select \* from emp\_tbl

insert into emp\_tbl values(1,'mangal singh','mangalsingh@gmail.com', 'manager')

insert into emp\_tbl values(2,'bipin singh','bipin@gmail.com', 'it incharg')

insert into emp\_tbl values(3,'dm singh','dm@gmail.com', 'assitant manager')

insert into emp\_tbl values(4,'w singh','w@gmail.com', 'computer operator')

insert into emp\_tbl values(5,'b singh','b@gmail.com', 'assistant director')

insert into emp\_tbl values(6,'c singh','c@gmail.com', 'director')

create table dep\_tbl(dep\_id int unique not null, dep\_name nvarchar(max) not null, dep\_salary nvarchar(max) not null, emp\_id int unique not null)

select \* from dep\_tbl

insert into dep\_tbl values(1,'IT','50000', 3)

insert into dep\_tbl values(2,'Admin','80000', 4)

insert into dep\_tbl values(3,'Account','35000', 1)

insert into dep\_tbl values(4,'HR','20000', 2)

select \* from emp\_tbl as a left join

dep\_tbl as b on

a.emp\_id = b.emp\_id

select a.emp\_id, emp\_name, email,designation,dep\_name, dep\_salary from emp\_tbl as a left join

dep\_tbl as b on

a.emp\_id = b.emp\_id

RIGHT JOIN

create table emp\_tbl(emp\_id int unique not null, emp\_name nvarchar(max) not null, email nvarchar(max), designation nvarchar(max) not null)

select \* from emp\_tbl

insert into emp\_tbl values(1,'mangal singh','mangalsingh@gmail.com', 'manager')

insert into emp\_tbl values(2,'bipin singh','bipin@gmail.com', 'it incharg')

insert into emp\_tbl values(3,'dm singh','dm@gmail.com', 'assitant manager')

insert into emp\_tbl values(4,'w singh','w@gmail.com', 'computer operator')

insert into emp\_tbl values(5,'b singh','b@gmail.com', 'assistant director')

insert into emp\_tbl values(6,'c singh','c@gmail.com', 'director')

create table dep\_tbl(dep\_id int unique not null, dep\_name nvarchar(max) not null, dep\_salary nvarchar(max) not null, emp\_id int unique not null)

select \* from dep\_tbl

insert into dep\_tbl values(1,'IT','50000', 3)

insert into dep\_tbl values(2,'Admin','80000', 4)

insert into dep\_tbl values(3,'Account','35000', 1)

insert into dep\_tbl values(4,'HR','20000', 2)

insert into dep\_tbl values(5,'sport','25000', 7)

insert into dep\_tbl values(6,'academic','10000', 8)

select \* from emp\_tbl as a right join

dep\_tbl as b on

a.emp\_id = b.emp\_id

select a.emp\_id, emp\_name, email,designation,dep\_name, dep\_salary from emp\_tbl as a right join

dep\_tbl as b on

a.emp\_id = b.emp\_id

FULL OUTER JOIN

create table emp\_tbl(emp\_id int unique not null, emp\_name nvarchar(max) not null, email nvarchar(max), designation nvarchar(max) not null)

select \* from emp\_tbl

insert into emp\_tbl values(1,'mangal singh','mangalsingh@gmail.com', 'manager')

insert into emp\_tbl values(2,'bipin singh','bipin@gmail.com', 'it incharg')

insert into emp\_tbl values(3,'dm singh','dm@gmail.com', 'assitant manager')

insert into emp\_tbl values(4,'w singh','w@gmail.com', 'computer operator')

insert into emp\_tbl values(5,'b singh','b@gmail.com', 'assistant director')

insert into emp\_tbl values(6,'c singh','c@gmail.com', 'director')

create table dep\_tbl(dep\_id int unique not null, dep\_name nvarchar(max) not null, dep\_salary nvarchar(max) not null, emp\_id int unique not null)

select \* from dep\_tbl

insert into dep\_tbl values(1,'IT','50000', 3)

insert into dep\_tbl values(2,'Admin','80000', 4)

insert into dep\_tbl values(3,'Account','35000', 1)

insert into dep\_tbl values(4,'HR','20000', 2)

insert into dep\_tbl values(5,'sport','25000', 7)

insert into dep\_tbl values(6,'academic','10000', 8)

select \* from emp\_tbl as a full outer join

dep\_tbl as b on

a.emp\_id = b.emp\_id

select a.emp\_id, emp\_name, email,designation,dep\_name, dep\_salary from emp\_tbl as a full outer join

dep\_tbl as b on

a.emp\_id = b.emp\_id

Identity and auto increment

Identity(1,1) it auto incremented

Increment by one

If identity(seed,number with increment)

Like

Identity(100,1)

Then start from 100 and auto increment by 1

If I put something like this on identity(1,5)

Then start from one and increment by 5

create table emp\_tbl(emp\_id int unique not null, emp\_name nvarchar(max) not null, email nvarchar(max), designation nvarchar(max) not null)

union and union all

in the union show the result set with both data in single result set

without common entries in table

union all show the result set with both data in single result set

with common entries as well

create table football\_tbl(id int unique not null, name nvarchar(max) not null, email nvarchar(max))

insert into football\_tbl values(1,'mangal singh','mangalsingh@gmail.com')

insert into football\_tbl values(2,'bipin singh','bipin@gmail.com')

insert into football\_tbl values(3,'santosh singh','santosh@gmail.com')

insert into football\_tbl values(4,'shiv singh','shiv@gmail.com')

insert into football\_tbl values(5,'pavan singh','pavan@gmail.com')

insert into football\_tbl values(6,'sachin singh','sachin@gmail.com')

create table hocky\_tbl(id int unique not null, name nvarchar(max) not null, email nvarchar(max))

insert into hocky\_tbl values(1,'mangal singh','mangalsingh@gmail.com')

insert into hocky\_tbl values(2,'bipin singh','bipin@gmail.com')

insert into hocky\_tbl values(3,'santosh singh','santosh@gmail.com')

insert into hocky\_tbl values(4,'shiv singh','shiv@gmail.com')

insert into hocky\_tbl values(5,'pavan singh','pavan@gmail.com')

insert into hocky\_tbl values(6,'sachin singh','sachin@gmail.com')

select \* from football\_tbl

union

select \* from hocky\_tbl

1 mangal singh mangalsingh@gmail.com

2 bipin singh bipin@gmail.com

3 dm singh dm@gmail.com

3 santosh singh santosh@gmail.com

4 shiv singh shiv@gmail.com

4 w singh w@gmail.com

5 b singh b@gmail.com

5 pavan singh pavan@gmail.com

6 c singh c@gmail.com

6 sachin singh sachin@gmail.com

select \* from football\_tbl

union all

select \* from hocky\_tbl

1 mangal singh mangalsingh@gmail.com

2 bipin singh bipin@gmail.com

3 santosh singh santosh@gmail.com

4 shiv singh shiv@gmail.com

5 pavan singh pavan@gmail.com

6 sachin singh sachin@gmail.com

1 mangal singh mangalsingh@gmail.com

2 bipin singh bipin@gmail.com

3 dm singh dm@gmail.com

4 w singh w@gmail.com

5 b singh b@gmail.com

6 c singh c@gmail.com

Intersect

Intersect have only common data shows in result set

select \* from football\_tbl

intersect

select \* from hocky\_tbl

1 mangal singh mangalsingh@gmail.com

2 bipin singh bipin@gmail.com

Except

Except show the left side data in result set without common entries

select \* from football\_tbl

except

select \* from hocky\_tbl

3 santosh singh santosh@gmail.com

4 shiv singh shiv@gmail.com

5 pavan singh pavan@gmail.com

6 sachin singh sachin@gmail.com

Aggregate function

Sum

Max

Min

Avg

Count

Sum syntex

select sum(salary) as totalamount from emp\_detailsSum syntax

max syntax

select max(salary) as maxamount from emp\_details

min syntax

select min(salary) as minamount from emp\_details

avg syntax

select avg(salary) as avgamount from emp\_details

count syntax

select count(salary) as totalamount from emp\_details

GROUP BY

create table employees(id int not null primary key identity, name varchar(max), Gender varchar(50), salary int, city varchar(max), dep\_id int unique not null)

insert into employees values('mangal singh', 'male', 10000, 'Mumbai',1)

insert into employees values('santosh singh', 'male', 12000, 'bangalor',2)

insert into employees values('pooja singh', 'female', 40000, 'Mumbai',3)

insert into employees values('bipin singh', 'male', 110000, 'pune',4)

insert into employees values('soni singh', 'female', 140000, 'delhi',5)

insert into employees values('dheeraj singh', 'female', 16000, 'pune',6)

insert into employees values('shiv singh', 'male', 17000, 'Mumbai',7)

select name,city, sum(salary), count(salary) from employees group by city,name

bipin singh pune 110000 1

dheeraj singh pune 16000 1

mangal singh Mumbai 10000 1

pooja singh Mumbai 40000 1

santosh singh bangalor 12000 1

shiv singh Mumbai 17000 1

soni singh delhi 140000 1

select gender,city, sum(salary), count(salary) from employees group by city,gender

female delhi 140000 1

female Mumbai 40000 1

female pune 16000 1

male bangalor 12000 1

male Mumbai 27000 2

male pune 110000 1

havin g clause and where clouse

where clause cannot use with group by command

that’s the reason we use having clause

select gender,city, sum(salary), count(city) from employees group by city,gender having city in ('mumbai')

female Mumbai 40000 1

male Mumbai 27000 2

select city, sum(salary), count(city) from employees group by city having city in ('mumbai')

Mumbai 67000 3

Where clause cannot use after group by command

We can use before group by command

Like this

select city, sum(salary), count(city) from employees where city in ('mumbai') group by city

save result before we got

Mumbai 67000 3

Where clause used to for filtering rows and having clause use to filtering groups

select city, sum(salary), count(city) from employees

group by city

having sum(salary) > 60000

delhi 140000 1

Mumbai 67000 3

pune 126000 2

where clouse cannot work with aggregate function before group by command

select city, sum(salary) from employees

where city in ('mumbai','pune')

group by city

having sum(salary) > 5000

Mumbai 67000

pune 126000

select id,city, sum(salary) from employees

where city in ('mumbai','pune') and id = 1 or id = 6

group by city,id

having sum(salary) > 5000

1 Mumbai 10000

6 pune 16000

Not in condtion

select id,city, sum(salary) from employees

where city not in ('mumbai')

group by city,id

having sum(salary) > 5000

2 bangalor 12000

4 pune 110000

5 delhi 140000

6 pune 16000

With in condition

select id,city, sum(salary) from employees

where city in ('mumbai')

group by city,id

having sum(salary) > 5000

1 Mumbai 10000

3 Mumbai 40000

7 Mumbai 17000

Not in condtion with having clause

select id,city, sum(salary) from employees

group by city,id

having city not in ('mumbai')

2 bangalor 12000

4 pune 110000

5 delhi 140000

6 pune 16000

With in condition with having clause

select id,city, sum(salary) from employees

group by city,id

having city in ('mumbai')

1 Mumbai 10000

3 Mumbai 40000

7 Mumbai 17000

select name,city,salary, Dep\_name from employees as a full outer join

department as b on

a.dep\_id = b.dep\_id

group by name,city,salary,Dep\_name

having city in ('mumbai','pune')

**VIEWS**

View in sql is just a save query

View also considerd as a virtual table

create view emp\_dep

as

select name,city,salary, Dep\_name from employees as a full outer join

department as b on

a.dep\_id = b.dep\_id

group by name,city,salary,Dep\_name

having city in ('mumbai','pune')

select \* from emp\_dep

bipin singh pune 110000 Counselling

dheeraj singh pune 16000 Admin

mangal singh Mumbai 10000 account

pooja singh Mumbai 40000 Admin

shiv singh Mumbai 17000 account

sp\_helptext use for show the view code

like

sp\_helptext emp\_dep

create view emp\_dep

as

select name,city,salary, Dep\_name from employees as a full outer join

department as b on

a.dep\_id = b.dep\_id

group by name,city,salary,Dep\_name

having city in ('mumbai','pune')

alter a view

alter view emp\_dep

as

select name,city,sum(salary) as salary from employees as a full outer join

department as b on

a.dep\_id = b.dep\_id

where city in ('mumbai')

group by city,name

having sum(salary) > 5000

select \* from emp\_dep

mangal singh Mumbai 10000

pooja singh Mumbai 40000

shiv singh Mumbai 17000

drop the view

drop view emp\_dep

insert into view

insert into emp\_dep(name,Gender,salary,city,dep\_id) values('lal singh', 'male',330000, 'delhi',4)

select \* from emp\_dep

1 mangal singh male 10000 Mumbai 1 account

2 santosh singh male 12000 bangalor 2 HR

3 pooja singh female 40000 Mumbai 3 Admin

4 bipin singh male 110000 pune 4 Counselling

5 soni singh female 140000 delhi 2 HR

6 dheeraj singh female 16000 pune 3 Admin

7 shiv singh male 17000 Mumbai 1 account

8 shivani singh female 30000 pratapgarh 2 HR

9 rabina singh female 330000 pratapgarh 3 Admin

10 lal singh male 330000 delhi 4 Counselling

insert into emp\_dep(name,Gender,salary,city,dep\_id,Dep\_name) values(' the beast', 'male',3330000, 'delhi',5)

insert into emp\_dep (Dep\_name) values('director')

select \* from emp\_dep

1 mangal singh male 10000 Mumbai 1 account

2 santosh singh male 12000 bangalor 2 HR

3 pooja singh female 40000 Mumbai 3 Admin

4 bipin singh male 110000 pune 4 Counselling

5 soni singh female 140000 delhi 2 HR

6 dheeraj singh female 16000 pune 3 Admin

7 shiv singh male 17000 Mumbai 1 account

8 shivani singh female 30000 pratapgarh 2 HR

9 rabina singh female 330000 pratapgarh 3 Admin

10 lal singh male 330000 delhi 4 Counselling

11 the beast male 3330000 delhi 5 director

Update the view

update emp\_dep set Dep\_name = 'CO' where id = 11

select \* from emp\_dep

1 mangal singh male 10000 Mumbai 1 account

2 santosh singh male 12000 bangalor 2 HR

3 pooja singh female 40000 Mumbai 3 Admin

4 bipin singh male 110000 pune 4 Counselling

5 soni singh female 140000 delhi 2 HR

6 dheeraj singh female 16000 pune 3 Admin

7 shiv singh male 17000 Mumbai 1 account

8 shivani singh female 30000 pratapgarh 2 HR

9 rabina singh female 330000 pratapgarh 3 Admin

10 lal singh male 330000 delhi 4 Counselling

11 TheBeast male 3330000 delhi 5 CO

delete from emp\_dep where id = 11

but we got error View or function 'emp\_dep' is not updatable because the modification affects multiple base tables.

So you can delete from originals table then its automatically delete from view

Like Operator

The like operator is used in where clause to search for specified

Pattern in a column

There are 3 wildcard used in conjuction with like operator

% the percent sign represent zero, one or multiple character

\_ the Underscore represent a single character

[] for multiple character

Where customer name like ‘a%’ find any values that start with “a

select \* from emp\_dep where name like 'm%'

Where customer name like ‘%a’ find any values that end with “a”

select \* from emp\_dep where name like '%h'

Where customer name like ‘%ng%’ find any values that have ‘ng’ in any position

select \* from emp\_dep where name like '%l %'

Where customer name like ‘\_a’ finds any value that have ‘a’ in the second position

select \* from emp\_dep where name like 'm\_\_\_\_\_\_\_\_\_\_\_'

Where customer name like ‘\_\_a’ finds any value that have ‘a’ in the third position

Where customer name like ‘a\_\_’ finds any value that have ‘a’ in the first position

select \* from emp\_dep where name like 'm\_\_\_\_\_\_\_\_\_\_h'

Where customer name like ‘[a,b]%’ find any values that start with “a,b”

Where customer name like ‘[a-d]%’ find the values that start between a to d

SUB QUERY

select \* from emp\_dep

where id in

(select id from emp\_dep where city = 'pratapgarh')

8 shivani singh female 30000 pratapgarh 2 HR

9 rabina singh female 330000 pratapgarh 3 Admin

select name,salary,city,Dep\_name from emp\_dep

where id in

(select id from emp\_dep where city = 'pratapgarh')

shivani singh 30000 pratapgarh HR

rabina singh 330000 pratapgarh Admin

update emp\_dep set salary = salary+2000 where

id in (

select id from emp\_dep where city = 'pratapgarh')

1 mangal singh male 10000 Mumbai 1 account

2 santosh singh male 12000 bangalor 2 HR

3 pooja singh female 40000 Mumbai 3 Admin

4 bipin singh male 110000 pune 4 Counselling

5 soni singh female 140000 delhi 2 HR

6 dheeraj singh female 16000 pune 3 Admin

7 shiv singh male 17000 Mumbai 1 account

8 shivani singh female 36000 pratapgarh 2 HR

9 rabina singh female 336000 pratapgarh 3 Admin

10 lal singh male 330000 delhi 4 Counselling

delete from emp\_dep where

id in (

select id from emp\_dep where city = 'mumbai'

)

Join

select \* from employees

where dep\_id in

(

select dep\_id from department where Dep\_name = 'admin'

)

3 pooja singh female 40000 Mumbai 3

6 dheeraj singh female 16000 pune 3

9 rabina singh female 336000 hpratapgarh 3

Order by can use

select \* from employees this call outer query

where dep\_id in

(

select id from emp\_dep where Dep\_name = 'admin' this is called subquery

) order by id desc

We can filter out the particular colum with mention on outer query where \* from

Types of subqueries

sub query devided into two amin categories

scalar subqueries

this subqueries return one row to the outer sql statement

operators = > < >= <= !=

select \* from employees

where id =

(

select id from emp\_dep where city = 'bangalor'

) order by id desc

multivalued subqueries

subqueries that return more than one row to the outer sql statement

operator in , any and all

select \* from employees

where id in

(

select id from emp\_dep where salary = 10000 or salary = 12000

) order by id desc

2 santosh singh male 12000 bangalor 2

1 mangal singh male 10000 Mumbai 1

select \* from employees

where id < any

(

select id from emp\_dep where salary = 10000 or salary = 12000

) order by id desc

1 mangal singh male 10000 Mumbai 1

select \* from employees

where id > all

(

select id from emp\_dep where salary = 10000 or salary = 12000

) order by id desc

10 lal singh male 330000 delhi 4

9 rabina singh female 336000 pratapgarh 3

8 shivani singh female 36000 pratapgarh 2

7 shiv singh male 17000 Mumbai 1

6 dheeraj singh female 16000 pune 3

5 soni singh female 140000 delhi 2

4 bipin singh male 110000 pune 4

3 pooja singh female 40000 Mumbai 3

SELF Contained subqueryi

Self contained subquery is that sub query which are not dependent on outer subquery

Mean self dependent query

select \* from employees

where dep\_id in

(select dep\_id from department where Dep\_name = 'HR')

2 santosh singh male 12000 bangalor 2

5 soni singh female 140000 delhi 2

8 shivani singh female 36000 pratapgarh 2

Co-Related Subquery

These query refrences one or more column from outer query and therefor dependet on outer query

In simple word

these sub query are called co-related subquery which are depend on outer query

co- related subquery cannot be run separately from the outer query

select \* from employees as e

where e.dep\_id in

(

select dep\_id from department where e.gender = 'male'

)

1 mangal singh male 10000 Mumbai 1

2 santosh singh male 12000 bangalor 2

4 bipin singh male 110000 pune 4

7 shiv singh male 17000 Mumbai 1

10 lal singh male 330000 delhi 4

Between operator

select \* from employees where salary between 20000 and 40000

3 pooja singh female 40000 Mumbai 3

8 shivani singh female 36000 pratapgarh 2

TOP

select top 5 \* from employees

1 mangal singh male 10000 Mumbai 1

2 santosh singh male 12000 bangalor 2

3 pooja singh female 40000 Mumbai 3

4 bipin singh male 110000 pune 4

5 soni singh female 140000 delhi 2

Percent

select top 40 percent \* from employees

1 mangal singh male 10000 Mumbai 1

2 santosh singh male 12000 bangalor 2

3 pooja singh female 40000 Mumbai 3

4 bipin singh male 110000 pune 4

Distinct

Distice operator return unique value not return any dublicate value

select distinct name from employees

IN Operator

select \* from employees where city in ('mumbai', 'pratapgarh')

1 mangal singh male 10000 Mumbai 1

3 pooja singh female 40000 Mumbai 3

7 shiv singh male 17000 Mumbai 1

8 shivani singh female 36000 pratapgarh 2

9 rabina singh female 336000 pratapgarh 3

NOT IN

select \* from employees where city not in ('mumbai', 'pratapgarh')

2 santosh singh male 12000 bangalor 2

4 bipin singh male 110000 pune 4

5 soni singh female 140000 delhi 2

6 dheeraj singh female 16000 pune 3

10 lal singh male 330000 delhi 4

SELECT INTO

Select into command use for take a backup of table

select \* into emp from employees

select \* from emp

1 mangal singh male 10000 Mumbai 1

2 santosh singh male 12000 bangalor 2

3 pooja singh female 40000 Mumbai 3

4 bipin singh male 110000 pune 4

5 soni singh female 140000 delhi 2

6 dheeraj singh female 16000 pune 3

7 shiv singh male 17000 Mumbai 1

8 shivani singh female 36000 pratapgarh 2

9 rabina singh female 336000 pratapgarh 3

10 lal singh male 330000 delhi 4

select name, salary, city into emp from employees

select \* from emp

mangal singh 10000 Mumbai

santosh singh 12000 bangalor

pooja singh 40000 Mumbai

bipin singh 110000 pune

soni singh 140000 delhi

dheeraj singh 16000 pune

shiv singh 17000 Mumbai

shivani singh 36000 pratapgarh

rabina singh 336000 pratapgarh

lal singh 330000 delhi

select name, salary, city into emp from employees where city in ('mumbai','pune')

mangal singh 10000 Mumbai

pooja singh 40000 Mumbai

bipin singh 110000 pune

dheeraj singh 16000 pune

shiv singh 17000 Mumbai

join in select into command

select \* into emp\_join from

employees as a

full outer join

department as b

on a.dep\_id = b.D\_id

select \* from emp\_join

1 mangal singh male 10000 Mumbai 1 1 account

2 santosh singh male 12000 bangalor 2 2 HR

3 pooja singh female 40000 Mumbai 3 3 Admin

4 bipin singh male 110000 pune 4 4 Counselling

5 soni singh female 140000 delhi 2 2 HR

6 dheeraj singh female 16000 pune 3 3 Admin

7 shiv singh male 17000 Mumbai 1 1 account

8 shivani singh female 36000 pratapgarh 2 2 HR

9 rabina singh female 336000 pratapgarh 3 3 Admin

10 lal singh male 330000 delhi 4 4 Counselling

select a.\*, b.Dep\_name into emp\_join from

employees as a

full outer join

department as b

on a.dep\_id = b.D\_id

1 mangal singh male 10000 Mumbai 1 account

2 santosh singh male 12000 bangalor 2 HR

3 pooja singh female 40000 Mumbai 3 Admin

4 bipin singh male 110000 pune 4 Counselling

5 soni singh female 140000 delhi 2 HR

6 dheeraj singh female 16000 pune 3 Admin

7 shiv singh male 17000 Mumbai 1 account

8 shivani singh female 36000 pratapgarh 2 HR

9 rabina singh female 336000 pratapgarh 3 Admin

10 lal singh male 330000 delhi 4 Counselling

Copy data from one data base to other database

select a.\*, b.Dep\_name into SalesDB.dbo.emp\_join from

employees as a

full outer join

department as b

on a.dep\_id = b.D\_id

copy table structur by this command

select \* into edm from employees where 1 <> 1

whole structure copied but constraint is not going to copy

copy one table data to other table

but both tables column must be same and equal

then you can perform this query

insert into student\_bio\_thane select name,age,class,std from student\_bio\_mumbai

1 manoj 12 7th A

2 dola 11 7th B

3 vishnu 13 7th C

4 shiva 12 7th A

5 gautam 11 7th B

6 amit 12 7th A

7 sandeep 13 7th A

8 ruchi 12 7th C

9 mangal 12 7th A

10 lal 11 7th B

11 santosh 13 7th C

12 bipin 12 7th A

13 rabina 11 7th B

14 shivani 12 7th A

15 dheeraj 13 7th A

16 akash 12 7th C

Insert specific data to specific column

insert into student\_bio\_thane(name,age) select name,age from student\_bio\_mumbai

change the database name and table name

rename database name

sp\_renamedb 'users', 'user'

table rename

sp\_rename 'customer','customer\_tbl'

sp\_rename 'customer','customer\_tbl','column'

column rename

sp\_rename 'customer\_tbl.customer','customers'

sp\_rename 'customer\_tbl.customer','customers','column'

sp\_rename '[customer\_tbl].[customer]','customers','column'

Store Procedure

Store procedure is a set of structured query language statement with an assigned name,which

Are stored in a relational database management system as a group so it can be reused and shred by

Multiple program

Types of store procedure

---system store procedure

--user defined store procedure

create proc emp\_data

as

begin

select \* from employees

end

exec emp\_data

1 mangal singh male 10000 Mumbai 1

2 santosh singh male 12000 bangalor 2

3 pooja singh female 40000 Mumbai 3

4 bipin singh male 110000 pune 4

5 soni singh female 140000 delhi 2

6 dheeraj singh female 16000 pune 3

7 shiv singh male 17000 Mumbai 1

8 shivani singh female 36000 pratapgarh 2

9 rabina singh female 336000 pratapgarh 3

10 lal singh male 330000 delhi 4

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

create proc emp\_byid

@id int

as

begin

select \* from employees where id = @id

end

emp\_byid 4

4 bipin singh male 110000 pune 4

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

create proc emp\_byidname

@id int,

@name varchar(50)

as

begin

select \* from employees where id = @id and name = @name

end

emp\_byidname 3,'santosh singh'

2 santosh singh male 12000 bangalor 2

3 pooja singh female 40000 Mumbai 3

Alter procedure

alter proc emp\_byidname

@id int,

@name varchar(50)

as

begin

select name,salary,city from employees where id = @id or name = @name

end

emp\_byidname 3,'santosh singh'

santosh singh 12000 bangalor

pooja singh 40000 Mumbai

show the store procedure code

sp\_helptext emp\_byidname

with encryption

alter proc emp\_byidname

@id int,

@name varchar(50)

with encryption

as

begin

select name,salary,city from employees where id = @id or name = @name

end

drop procedure

drop proc emp\_byidname

OUTPUT parameter

create proc emp\_det

@gender varchar(50),

@employeecount int output

as

begin

select @employeecount = count(id) from employees

where Gender = @gender

end

declare @totalcount int

execute emp\_det 'male', @totalcount output

select @totalcount

5

Functions

Scalar valued function

Return a single value

Create function without parameter

create function showmessage()

returns varchar (100)

as

begin

return 'welcome function'

end

select dbo.showmessage();

welcome function

create function with single parameter

create function takenumber(@num as int)

returns int

as

begin

return (@num \* @num)

end

select dbo.takenumber(3)

9

Create function with multiple parameters

create function addition(@num1 as int, @num2 as int)

returns int

as

begin

return(@num1 + @num2)

end

select dbo.addition(2,10)

12

ALTER FUNCTION

alter function takenumber(@num as int)

returns int

as

begin

return (@num \* @num \* @num)

end

select dbo.takenumber(3)

27

Drop a function

drop function dbo.takenumber

if else condition function

create function voterage(@age as int)

returns varchar(100)

as

begin

declare @str varchar(100)

if @age >= 18

begin

set @str = 'you are elegible'

end

else

begin

set @str = 'not eligible'

end

return @str

end

select dbo.voterage(18)

you are elegible

scalar function can call other function

create function mydatetime()

returns datetime

as

begin

return getdate()

end

select dbo.mydatetime()

inline table valued function

return a table

without parameter

create function users()

returns table

as

return(select \* from user\_details)

select \* from dbo.users()

with parameter

create function fn\_employees(@city as varchar(100))

returns table

as

return(select \* from employees where city in (@city))

select \* from fn\_employees('mumbai')

1 mangal singh male 10000 Mumbai 1

3 pooja singh female 40000 Mumbai 3

7 shiv singh male 17000 Mumbai 1

select \* from fn\_employees('mumbai') as a

inner join department as b

on

a.dep\_id = b.D\_id

1 mangal singh male 10000 Mumbai 1 1 account

3 pooja singh female 40000 Mumbai 3 3 Admin

7 shiv singh male 17000 Mumbai 1 1 account

alter function fn\_employees(@city as varchar(100), @salary int)

returns table

as

return(select \* from employees where city in (@city) and salary >= @salary)

select \* from fn\_employees('mumbai',12000)

select \* from fn\_employees('mumbai',12000) as a

inner join department as b

on

a.dep\_id = b.D\_id

3 pooja singh female 40000 Mumbai 3 3 Admin

7 shiv singh male 17000 Mumbai 1 1 account

Multi statement value function

Multi statement table value function is a table valued function that returns the result of multiple statement

create function fn\_getdata(@gender varchar(50))

returns @mytable table(std\_id int, std\_name varchar(max),std\_gender varchar(30))

as

begin

insert into @mytable

select id,name,Gender from employees where gender = @gender

return

end

select \* from fn\_getdata('male')

1 mangal singh male

2 santosh singh male

4 bipin singh male

7 shiv singh male

10 lal singh male

Three Tables join

create table teachers(T\_id int primary key identity, T\_Name varchar(100), salary int, Qualification varchar(50))

insert into teachers values('jhari prasad',10000, 'BCA')

insert into teachers values('hari prasad',10000, 'BCA')

insert into teachers values('lari prasad',10000, 'BCA')

insert into teachers values('tari prasad',10000, 'BCA')

insert into teachers values('mari prasad',10000, 'BCA')

select \* from teachers

select \* from students

select \* from branch

create table students(S\_id int primary key identity, S\_Name varchar(100), class varchar(50),fees int ,T\_id int foreign key(T\_id) references teachers(T\_id))

insert into students values('mangal singh','7th', 2000,4)

insert into students values('amit singh','7th', 2000,5)

insert into students values('rakesh singh','7th', 2000,3)

insert into students values('bharat singh','7th', 2000,1)

insert into students values('priya singh','7th', 2000,2)

create table branch (B\_id int primary key identity, City varchar(50),S\_id int foreign key(S\_id) references students(S\_id))

insert into branch values('mumbai',1)

insert into branch values('pune',3)

insert into branch values('mumbai',2)

insert into branch values('delhi',4)

insert into branch values('delhi',5)

select a.T\_Name,a.Qualification,a.salary,b.S\_Name,b.class,b.fees,c.City from teachers as a

inner join students as b

on a.T\_id = b.T\_id

inner join branch as c

on b.S\_id = c.S\_id

create view teachers\_details

as

select a.T\_Name,a.Qualification,a.salary,b.S\_Name,b.class,b.fees,c.City from teachers as a

inner join students as b

on a.T\_id = b.T\_id

inner join branch as c

on b.S\_id = c.S\_id

select \* from teachers\_details

update teachers\_details set class = '9th' where S\_Name = 'rakesh singh'

triggers

dml trigger data manipulate language insert update delete

ddl trigger data definition language create alter

DML Triggers insert

After trigger

create trigger tr\_student

on students

after insert

as

begin

select \* from inserted

end

select \* from students

insert into students values('radha singh','8th',3000,5)

7 radha singh 8th 3000 5

DML Triggers delete

alter trigger tr\_studentDel

on students

after delete

as

begin

select \* from deleted

end

delete students where S\_id = 6

6 pooja singh 8th 3000 4

Inseted info save on other table get from inserted table

create table audit\_info(audit\_id int identity primary key, audit\_info varchar(max),user\_id int)

alter trigger tr\_info

on students

after insert

as

begin

declare @id int

declare @Name varchar(max)

select @id = S\_id, @Name = S\_Name from inserted

insert into audit\_info

values('New Student Inserted with the Id ' + cast(@id as varchar(50)) + ' and Name ' + cast(@Name as varchar(50)) + ' is added at ' +cast(getdate() as varchar(50)),@id)

end

insert into students values('radha singh','8th',3000,5)

select \* from audit\_info

4 New Student Inserted with the Id 12 and Name radha singh is added at Mar 11 2022 9:58PM 12

deleted info save on other table get from deleted table

create table delete\_info(audit\_id int identity primary key, audit\_info varchar(max),user\_id int)

create trigger tr\_info\_deleted

on students

after delete

as

begin

declare @id int

declare @Name varchar(max)

select @id = S\_id, @Name = S\_Name from deleted

insert into delete\_info

values('Student Deleted with the Id ' + cast(@id as varchar(50)) + ' and Name ' + cast(@Name as varchar(50)) + ' at ' +cast(getdate() as varchar(50)),@id)

end

delete from students where S\_id = 12

select \* from delete\_info

2 Student Deleted with the Id 12 and Name radha singh at Mar 11 2022 10:01PM 12

Trigger for update

create trigger tr\_updated

on students

after update

as

begin

select \* from inserted

select \* from deleted

end

instead of triggers

create trigger tr\_instead

on students

instead of insert

as

begin

print 'not allow to insert into student'

select \* from inserted

end

select \* from students

insert into students values('rakesh yadav','8th',3000,4)

instead update

create trigger tr\_instead\_update

on students

instead of update

as

begin

print 'not allowed to update '

end

update students set S\_Name = 'rakesh singh' where S\_id = 13

instead of delete

create trigger tr\_instead\_delete

on students

instead of delete

as

begin

print 'not allowed to delete '

end

delete from students where S\_id = 13

drop trigger <trigger name>

create table student\_audit(audit\_id int primary key identity(5,5),audit\_info varchar(max))

select \* from student\_audit

create trigger tr\_student\_audit

on students

instead of insert

as

begin

insert into student\_audit values('someone try to insert data in this table at '+ cast(getdate() as varchar(50)))

end

select \* from student\_audit

instead of update

create trigger tr\_student\_audit\_update

on students

instead of update

as

begin

insert into student\_audit values('someone try to update data in this table at '+ cast(getdate() as varchar(50)))

end

update students set S\_Name = 'ram pratap' where S\_id = 13

select \* from students

select \* from student\_audit

instead of delete

create trigger tr\_student\_audit\_delete

on students

with encryption

instead of delete

as

begin

insert into student\_audit values('someone try to Delete data in this table at '+ cast(getdate() as varchar(50)))

end

delete from students where S\_id = 13

select \* from students

select \* from student\_audit

instead of delete with views all row

create table heros(H\_id int primary key identity,H\_name varchar(50),H\_power int)

insert into heros values('super man',2000)

insert into heros values('bat man',1000)

insert into heros values('ant man',500)

select \* from heros

create table speed (S\_id int primary key identity,Speed nvarchar(50),H\_id int unique)

drop table speed

insert into speed values('20km/ps',1)

insert into speed values('1km/ps',2)

insert into speed values('2km/ps',3)

select \* from speed

create view vw\_heros

as

select a.H\_id,a.H\_name,a.H\_power,b.Speed from heros as a

full outer join speed as b

on

a.H\_id = b.H\_id

select \* from vw\_heros

create trigger tr\_heros

on vw\_heros

instead of delete

as

begin

delete from heros where H\_id in(

select H\_id from deleted

)

delete from speed where H\_id in(

select H\_id from deleted

)

end

delete from vw\_heros where H\_id = 3

DDL Triggers

Triggers for create, alter and drop

create database tr\_trigger

use tr\_trigger

create trigger tr\_createtable

on database

for CREATE\_TABLE

as

begin

print 'table is created'

end

create table stable(id int primary key identity)

alter table

create trigger tr\_altertable

on database

for ALTER\_TABLE

as

begin

print 'table is altered'

end

alter table stable add name varchar(50)

drop table

create trigger tr\_droptable

on database

for DROP\_TABLE

as

begin

print 'table is droped'

end

drop table stable

create trigger tr\_droptable

on database

for DROP\_TABLE,CREATE\_TABLE,ALTER\_TABLE

as

begin

print 'table is droped,create,altered'

end

create procedure not allowed

create trigger tr\_sp\_nottocreateproc

on database

for create\_procedure

as

begin

rollback

print 'you cannot create store procedure'

end

create proc hey

as

begin

print 'my proc'

end

rollback command is use for deny the actions like create alter or drop

is also use for view, function , procedure

you can DISABLE a trigger by this command

disable trigger tr\_sp\_nottocreateproc on database

stop renaming table and column

create trigger tr\_rename

on database

for rename

as

begin

rollback

print 'you cannot rename table'

end

execute sp\_rename 'tb\_id' ,'tb\_name'

drop trigger of ddl on database trigger

drop trigger tr\_rename on database

all server dll trigger

create trigger tr\_allserver

on all server

for create\_table

as

begin

rollback

print 'cannot create table '

end

create table t\_sam(id int)

drop trigger tr\_rename on all server

disable trigger tr\_sp\_nottocreateproc on all server

execution of dml triggers

order vise trigger

exec sp\_settriggerorder

@triggername = 'trigger\_1',

@order = 'first',

@stmttype = 'insert';

GUID

create table table\_customer(id uniqueidentifier primary key default newid(), name varchar(50))

insert into table\_customer values(default,'mangal singh')

select \* from table\_customer

7DA34946-9CB4-4FBF-886D-446609A397F8 mangal singh

Copy data on database to other database table

Like this

insert into customer

select \* from tr\_trigger.dbo.table\_customer

union all

select \* from [user].dbo.employees

**Composite Key**

**Create Multiple primary key in many colums**

create table emp\_det(

EMP\_ID int not null,

DEP\_ID int not null,

Order\_id int not null,

Product\_id int not null,

Sales\_id int not null,

E\_name varchar(50),

primary key(EMP\_ID,DEP\_ID,Order\_id,Product\_id,Sales\_id)

)

alter table sales\_tbl add primary key(Cus\_id,Sales\_id)

string function

declare @start int

set @start = 65

while (@start <= 90)

begin

print char(@start)

set @start = @start +1

end

A

B

C

D

E

F

G

H

I

J

K

L

M

N

O

P

Q

R

S

T

U

V

W

X

Y

Z

select ascii('m')

select char(109)

Left trim or right trim

select firstname,middlename, lastname, rtrim(ltrim(firstname))+ ' '+ middlename + ' '+ lastname as fullname from employee\_tbl

upper case value

select upper(middlename) from employees

length count

select len(firstname) from employees

select len(ltrim(firstname)) from employees

create index

**create index on single colum**

**for quick result set**

select \* from employees

create index IN\_salary

on

employees(salary asc)

select \* from employees

where salary >10000 and salary <= 50000

for drop index

drop index employees.IN\_salary;

clustard index

not clustered index

create nonclustered index NINX\_name

on employees (name asc)

create index NINX\_name

on employees (name asc)

mention nonclustered is optional you can directly create index its automatically generate non clustered index

COMPUTED COLUMNS

create table gross\_salary(

id int unique identity,

name varchar(50),

basicpay int,

housepay int,

conveincepay int,

grosspay as basicpay+housepay+conveincepay

)

insert into gross\_salary values('mangal singh',2000,400,400)

select \* from gross\_salary

constant colum

alter table gross\_salary add company\_name as 'abc'

select \* from gross\_salary

**1 mangal singh 2000 400 400 2800 abc**

**Function computed colum**

alter table gross\_salary add date\_ as getdate()

**1 mangal singh 2000 400 400 2800 abc 2022-03-13 22:24:42.733**

**There is two type of computed column**

**Persisted and non persisted**

create table gross\_salary(

id int unique identity,

name varchar(50),

basicpay int,

housepay int,

conveincepay int,

grosspay as basicpay+housepay+conveincepay perasited

)

**Drop the computed column**

alter table gross\_salary drop column grosspay

create table product (

p\_id int not null primary key,

p\_name varchar(50),

p\_price int,

P\_quantity int,

totalcost as P\_price \* P\_quantity persited

Group by With cube

Cube is use for grand total

select city, gender, sum(salary) from employees

group by cube(city,gender)

select city, gender, sum(salary) from employees

group by city,gender with cube

Group by With rollup

select city, gender, sum(salary) from employees

group by city,gender with rollup

grouping sets

select city, gender, sum(salary) from employees

group by

grouping sets(

(city,gender),

(city),

(gender),

()

)

select city, gender, sum(salary) from employees

group by

grouping sets(

(city,gender),

(city),

(gender),

()

)

order by grouping(city),grouping(gender) asc

merge statement in sql

**Duplicate data**

select Category, COUNT(Category) from Amazon

group by Category

having COUNT(Category) > 1

transation sql

begin transaction

update student\_tbl set name = 'rakesh' where id = 1

commit transaction -- permanent update

commit or rollback the transaction

begin transaction

update student\_tbl set name = 'rakesh' where id = 1

rollback transaction -- undo the update