--1)creating a databse

create database d

--2)use a databse

use d

--3)creating a table

create table pw(id int,idd int, sa int,nam varchar(90))

drop table pw

--4) insert into table

insert into pw(id,idd,sa,nam)

values(1,2,3, 'jdfi'),

(4,5,6,'diwjk'),

(7,8,9, 'eoke'),

(1,2,3, 'jdfi'),

(4,5,6,'diwjk'),

(7,8,9, 'eoke'),

(1,2,3, 'jdfi'),

(4,5,6,'diwjk'),

(7,8,9, 'eoke')

--5) print all the table values

select \*from pw

--6)drop a table

drop table pw

-- 7)drop a database

create database dd

drop database d

--8) inserting more values into table

insert into pw(id,idd,sa,nam)

values(11,12,13, 'fiff')

select \*from pw

--9) changing a value of the tables using update and set keyword

update pw

set id=2

where idd= 2

update pw

set id=1

where idd=2

select \*from pw

update pw

set sa = 6

where idd=2

update pw

set sa = 3

where idd=2

--10)deleting row using delete keyword

select \*from pw

delete from pw

where id = 1

delete from pw

where id=4

delete from pw

where id =7

delete from pw

where id=11

delete from pw

--11)delete a column using delete alter drop

select \*from pw

alter table pw

drop column id

alter table pw

drop column idd

alter table pw

drop column sa

alter pw

drop column nam

-- 12)delete all values from the tables delete

delete from pw

--13)select one particular column from the table

select id,idd ,sa ,nam from pw

select idd from pw

select nam from pw

-- 14)selecting values by adding the condtion like =,<,>,<=>=

select \*from pw

where id=1

select\* from pw

where id >5

select \*from pw

where id = 1

select \*from pw

where id <20

select \*from pw

where id>=4

-- 15)order all column by asc desc

select \*from pw

order by id desc

select id from pw

order by id desc

select idd from pw

order by idd desc

-- 16)aggregate functions sum,count,avg,min,max,etc

select sum(id) as sum from pw

select \*from pw

select avg(id) as avg from pw

select min(id) as min from pw

select count(id) as count from pw

select max(id) as max from pw

select count(\*) as count from pw

-- 17) grouy by - always use aggrative function we want to grouop more colimns

select id from pw

group by id

select idd from pw

group by idd

select id ,sum(idd) from pw

group by id

select id, min(idd) from pw

group by id

select id, max(idd) from pw

group by id

select id,avg(idd) from pw

group by id

select id, count(idd) from pw

group by id

-- 18) having - fltering out adding condition <,>,=,<=,>=

select id, sum(idd) from pw

group by id

having id =1

select id, sum(idd) from pw

group by id

having id<20

-- 19) join,left join ,right join,full join

create table t1(id int,idd int,sa int)

insert into t1(id,idd,sa)

values(1,2,3),

(4,5,6),

(16,17,18)

create table d2(id int,idd int, sa int)

insert into d2(id,idd,sa)

values(1,2,3),

(4,5,6),

(7,8,9)

select \*from t1

select t.id, d.id

from t1 t

join d2 d

on t.id=d.id

select t.id,d.id

from t1 t

left join d2 d

on t.id=d.id

select t.id,d.id

from t1 t

right join d2 d

on t.id=d.id

select t.id,d.id

from t1 t

full join d2 d

on t.id= d.id

--20)case statement

select id,

case

when id=1 then 'gbye'

when id=2 then 'thend'

when id>1 then 'end'

else 'lastday'

end

from pw

select id,

case

when id=1 then 'goodbye'

when id>10 then 'thend'

when id>5 then 'end'

when id<4 then 'bye'

else 'ended'

end

from pw

select\*,

case

when id=1 then 'gbye'

when id>7 then 'thend'

when id>9then 'end'

else 'lastday'

end

from kz

select id,

case

when id=1 then 'ended'

when id>5 then 'theend'

when id>10 then 'thelatday'

else 'endingtoady'

end

from pw

--21) and operator and or oepator and = 2 pass, or =1 or 2 pass

select \* from pw

where id =1 and idd=2

select \*from pw

where id=4 and idd =5

select \*from pw

where id=1 and idd=2

select \*from pw

where id=1 and idd=2

select \*from pw

where id= 1 or idd=8

select \*from pw

where id=4 or idd =8

select \*from kz where id= 1 and idd=2

--22) lower,upper,len,replace,trim,cast,convert

select lower(nam), upper(nam) from pw

select len('euej')

select \*from pw

select len(id) from pw

select len(nam) from pw

select replace(nam, 'd','n') from pw

select replace(id,1,2) from pw

select trim (' dhdhd ')

select rtrim (' drkrk' )

select ltrim( ' hdhddn')

select cast('10' as int) \*20

select convert (int, '10') \*20

--23)view

create view v1

as

select id,idd from pw

select \*from v1

create view v2

as

select id from pw

select\*from v2

create view vh as select

id,idd,sa from kz

--24)functions scalar,inline

--scalr

create function f1(@n1 int, @n2 int)

returns int

as

begin

return @n1+@n2

end

go

select dbo.f1(1,2)

select dbo.f1(1,2)

--inline

create function f2(@n1 int, @n2 int)

returns table

as

return(select @n1+@n2 as sum);

go

select\*from dbo.f2(1,2)

create function yd(@n1 int, @n2 int)

returns table

as

return

(select @n1+@n2 as sum);

go

select\*from dbo.f2(6,7)

select\*from dbo.f2(1,2)

--25)distinct - removes repated values from the coulmn

select\*from pw

select distinct(idd) from pw

select distinct(id) from pw

-- 26)iif -true or false

select iif(0>10 , 'true', 'false');

select iif(0<20 , 'true', 'false') as o

-- 27) transction rollback - removed function from the main table we can add back tran roolback

select\*from pw

begin tran demo

delete from pw where id=1

rollback tran demo

begin tran demo1

delete from pw where id=1

rollback tran demo1

-- 28)triiger

create table anuu(id int,idd int,sa int)

insert into anuu(idd,id,sa)

values(1,2,3),

(4,5,6),

(7,8,9)

create table qo(id int,idd int,sa int)

create trigger ql

on anuu

after delete

as

begin

insert into qo(id,idd,sa)

select deleted.id,deleted.idd,deleted.sa

from deleted;

end

delete from anuu where id=1

select\*from qo

select\*from anuu

delete from anuu where id=2

CREATE TRIGGER lo

ON q1

AFTER DELETE

AS

BEGIN

INSERT INTO qd (id, idd, sa)

SELECT DELETED.id, DELETED.idd, DELETED.sa

FROM DELETED;

END;

delete from q1 where id=1

-- 29) stored procedure

create proc gk @id int

as

select \*from pw where id=@id

go

exec gk @id =1

create proc gkk

as

select \*from pw

go

exec gkk

create proc po @id int,@idd int

as

select \*from pw where id=@id and idd=@idd

go

--30) union,uion all,except,inersect

select id,idd,sa, nam from pw where id=1

union

select id,idd,sa,nam from pw where id in(1,4)

select id,idd,sa,nam from pw where id=1

union all

select id,idd,sa,nam from pw where id in(1,7)

select id,idd,sa,nam from pw where id=1

except

select id,idd,sa,nam from pw where id in(4,7)

select id,idd,sa,nam from pw where id=1

intersect

select id,idd,sa,nam from pw where id in(1,7)

--31)rank,dense\_rank,row\_number,ntile(1)

select id,idd,rank() over(order by id desc) as rank from pw

select id,idd,sa,dense\_rank() over(order by id desc)as dense\_rank from pw

select id,idd,sa, row\_number() over(order by id desc) as row\_numer from pw

select id,idd,sa, ntile(3) over(order by id desc) as ntile from pw

select id,idd,sa,

--32)between , is not null,is null

select \*from pw where id between 0 and 20

select \*from pw where id is not null

select \*from pw where id is null

create database slipper

use slipper;

create database lastday;

drop database lastday;

create table bye(id int, namee varchar(60);

alter table bye

add lname varchar(50);

select \*from bye;

alter table bye

drop column lname;

select \*from bye;

insert into bye

select 1, 'john';

select \*from bye;

insert into bye

values(2,'fv'), (8, 'iu');

select \*from bye;

alter table bye

add las varchar(50);

select \*from bye;

insert into bye

values(10, 'j','h')

select \*from bye;

insert into bye

(name)

select 'tarsh';

select \* from bye;

select \* from bye

where id=1;

update bye

set name = 'bn'

where id =10

select \*from bye

update bye

set id = 200;

select \*from bye;

alter table bye

add eg int identity(1,2)

delete from bye where id = 200;

select \*from bye

update bye

set id = 600;

select \*from bye

truncate table bye;

select \*from bye

create table hg

( roll int primary key,

namee varchar(50) default('hg'),

age int check(age>10),

addr varchar(100) unique,

[name] varchar(50) not null);

select \*from hg;

insert into hg

(roll, age, addr,[name])

values(2, 90, 'h', 'ol');

select \*from hg;

create table last

(roll int foreign key references hg(roll),

sub varchar(20))

insert into last

values(1,'eg'),

(9,'lp');

select \*from las

create table bgye

(id int , namee varchar(60), lastname text);

insert into bgye

(id, namee, lastname)

values(4,'g', 'gh');

select \*from bgye

select distinct id,namee from bgye

select distinct id from bgye

select \*from bgye where namee = 'og'

select \*from bgye where id not in (1,2,3,4)

select \*from bgye where id = 4 and namee= 'g';

select \*from bgye where id = 6 or namee = 'g'

insert into bgye

values(3, 'wd' , 'ul');

select \*from bgye where namee like '%h%'

select \* from bgye where namee like 'w'

select \*from bgye where id between 3 and 10;

select \*from bgye where id is null

create table ab

(fname varchar(50),

lname varchar(50),

salary int,

id int,

idd int,

d date

)

create table bb(fname varchar(50),lname varchar(50),salary int,id int ,idd int,d date)

create trigger ttzz

on ab

after delete

as

begin

insert into bb(fname,lname,salary,id,idd,d)

select deleted.fname,deleted.lname,deleted.salary,deleted.id,deleted.idd,deleted.d

from deleted;

end;

select \*from ab

delete from ab where id = 64

select \*from bb

insert into ab

values('cg', 'hf', 5000, 64, 93, '2024-10-01'),

('kl', 'h',689, 4, 23, '2024-10-01'),

('vd', 'kf', 6400, 3, 6, '2024-10-01'),

('bv','vnv',400, 6, 8,'2024-10-01')

select sum(idd) as a from ab

select \*from ab

create table ms

(id int,

dname varchar(80))

insert into ms

values(7, 'ao'),

(3, 'jn')

select \*from ms

select fname, a.id, dname from

ab as a

join ms m

on a.id= m.id

select fname, a.id, dname from

ab as a

right join ms m

on a.id = m.id

select fname, a.id,dname from

ab as a

full outer join ms m

on a.id = m.id

select fname, a.id, dname from

ab as a

join ms m

on a.id=m.id

left join newtable n

on a.id=m.id

select \*from fact

--fucctions

-- sytem define, user define

-- sytem define types

-- aggreatae function, string function , date function, windows function

--agrreate function - priovide

select \*from ab

select avg(salary) as averaye\_salary from ab

-- aggretive funtions means sggreated coulmns you cant add not aggreated column

--in the select query

select avg(salary) as records from ab

select max(salary) as records from ab

select min(salary) as cords from ab

select sum(salary) as records from ab

--group by

select id, avg(salary) avg\_slaray, count(fname) f\_anu from ab

group by id

create table fac(area int, id int , total int)

insert into fac

values(23,89,46)

select \*from fact

select id , count(total)

from fac

group by id

select id, sum(salary) av\_salarey, count(fname)

from ab

group by id;

select \*from fact

select id

from ab g

group by id s

select count(fname) as no

from ab a

left join ms m

on a.id = m.id

where idd <> 4

group by dname

having count(fname)>1

select trim( ' cg')

select fname from ab

selct left(fname, 2) from ab

select substring(fname,2,3) from ab

select len(fname) from ab

select fname, replace(fname,'a','e') from ab

select fname, upper(fname), lower(fname) from ab

select id=1, cast(id as varchar )+ 'ccc' from ab

select fname, convert(varchar(20),106) from ab

select \*from ab

create view vw\_ab

as

select fname,d,idd,lname from ab

select \*from vw\_ab

insert into ab

values

('anu', 'k', 567,56, 78,'2024-11-20')

==triggers

create trigger tr

on ab

for insert

as begin

declare @fname varchar(30)

select @fname = fname from inserted

select 'new record with fanme' +@fname+ 'is added to en tanle'

end

create trigger tr\_datelte

on ab

for delete

as begin

declare @fname varchar(30)

select @fname = fname from deleted

select 'record with fanme' +@fname+ 'is deleted froma ab'

end

delete from ab where id =3

select \*from ab

create trigger tr\_u

on ab

for update

as begin

declare @fanme varchar(30)

select @fname = fname from ins

insert @oldsal = salary from ins

select @oldsal = salary from deleted

select 'employe' +@fname+ 'salary is incorreted from rs' + cast (oldsal as varchar)+ to rs +cast(@oldsal as varchar)

end

update ab

set salary = 15000

where salary=400

select \*from ab order by salary desc

select top 2 \* from ab order by salary desc

--limit cross

select \* from ab order by salary desc limit 2

select \*from ab

where salary > (select avg(salary) from ab)

select \*from ab

update ab

set salary = 15000

where salary=400

select \*from ab order by salary desc

select top 2 \* from ab order by salary desc

--limit cross

select \* from ab order by salary desc limit 2

select \*from ab

where salary > (select avg(salary) from ab)

select \*from ab

-- iif

select \*,iif(salary >125000, 'group a', ' group b') salary

from ab

select \*, iif(salary>125000, salary +salary\*0.05, iif(salary> 110000, salary+salary\*0.08, salary+salary\*0.1)) new from ab

-- case statenment

select \*,

case

when salary >125000

then salary+salary\*0.5

when salary >110000

then salary+salary\*0.8

else salary+salary\*0.1

end as new\_salary

from ab

-- transction, rollback and commit

begin tran demo

delete from ab where lname='h'

select \*from ab

rollback tran demo

commit tran demo

select\*from ab

-- exception handling

select salary/0 from ab

begin try

select salary/0 from ab

end try

begin catch

select 'error'

end catch

--index

-- table scan, idex scan, and index seek

-- 2types of indexx : clusetred -1 and npon clustetred-999

--primsry key in atble , clsuetred is atomatcaially created for the column

create nonclustered index nci on ab(salary)

drop index nci2 on ab

sp\_helpindex ab

create clustered index ci on ab(salary)

drop index ci on ab

sp\_helpindex ab

select\*from ab

--set operaators

-- uion,uinon all, except,intersect

select \*from ab

--union all -- it will give 2 values are same in both thye stament then it will show two imes

select fname,salary,id,d,idd from ab where id = 64

union all

select fname,salary,id,d,idd from ab where id in (4,64)

-- union - if two values are same in both stament then it wont repeat it will give only once

select fname,salary,id,d,idd from ab where id =4

union

select fname,salary,id,d,idd from ab where id in(4,64)

-- whatever we give in 2nd stanment that will removed if that values is there in 1St stanment

select fname,salary,id,d,idd from ab where id in(4, 56)

except

select fname,salary,id,d,idd from ab where id= 56

select\*from ab

--intersect -- idf both values from single set matches then then it will show

select fname,salary,id,d, idd from ab where id in(4,56)

intersect

select fname,salary,id,d,idd from ab where id = 56

--cte(coomon table expression)

with em as(

select \*,

dense\_rank() over(order by salary) as salary\_rank

from ab)

select \*from em where salary\_rank=3

with w as(

select\*, dense\_rank() over(order by salary) as salary\_rank

from ab),

2ndcebame as (2nd cte),

3rdcetname as (3rd cte)

select \*from w where salary\_rank =4

WITH excte AS (

SELECT \*,

DENSE\_RANK()OVER(ORDER BY salary) salary\_rank

FROM ab),

2ndctename AS ( 2ND CTE QUERY),

3rdctename AS (3RD CTE QUERY)

SELECT \* FROM excte WHERE salary\_rank=3

--- order by particular column

select \*from ab

order by idd desc

-=extension caluses of group by

select idd,count(idd) from ab

group by id

select id,idd, count(salary) from ab

group by

grouping sets(id,idd)

select id,idd,count(salary) as col

from ab

group by

rollup(id,idd)

select id,idd ,count(salary)

from ab

group by

cube(idd,id)

--if else,while(condtion statements)

declare @dd int

set @dd=3

if @dd in(select id from ab)

begin

select \*from ab where id=@

end

else begin

select 'id does not exist'

end

------cast,convert

select cast( '10' as int) \*20 as cast\_

select convert(int, '10') \*20 as cat

select cast('10' as int)\*20 as cast

select convert(int ,'10') \*20 as cast

--while

declare @i int =5

while(@i>0)

begin

select\*from ab where id=@i

set @i=@i -1

end

--continue in while loop

--when the continue keyword is encouyrtred in a loop,

--the particular run is completed and the next run will start

---break in while loop

-when the break keyword is encoutered in a loop, the loop

declare @i int =5

while(@i>0)

begin

if @i=3

begin

set @i=@i-1

continue

end

else if @i=2

begin

break

end

else

begin

select \*from ab where id=@i

end

set @i=@i-1

end