use A

select \* from sys.tables

--DDL

create table Employee (id int constraint Pk primary key,

name varchar(20) not null , address varchar(20),

age int constraint AgeCon check (age between 20 and 40),

dept varchar(20) default 'HR' constraint DeptC check (dept In ('HR','Accts','Sales'))

)

--DML

insert into Employee values(1,'Ajay', 'Delhi', 21, 'Accts'),

(2,'Ajay', 'Delhi', 21, 'Accts'),

(3,'Ajay', 'Delhi', 21, 'Accts'),

(4,'Ajay', 'Delhi', 21, 'Accts')

--DML

Update Employee set dept='Accts' , age = age + 2

where id Between 2 And 4

select \* from Employee

-- To add Column/ Modify / Delete

alter table Employee add salary int constraint SalCon check (salary between 12000 AND 50000)

--DML / DQL

select \* from Employee

--remove column Address

alter table Employee drop column address

-- Modify column , change with of Name

--DDL

alter table employee alter column name varchar(30)

-- Rename a column

sp\_rename 'Employee.address' ,'EmployeeAddress' ,'column'

-- Remove a table

drop table Employee --DDL

-- Delete all records

delete Employee -- DML

-- Delete is DML command

-- where clause can be used

-- Delete all records

truncate table Employee

-- Truncate table is a ddl command

-- cannot use where

Stored Procedures

A set of SQL commands

Reusability

alter proc SelectEmployee

AS

Begin

select \* from employee

End

exec SelectEmployee

create proc InsertEmployee

As

Begin

insert into Employee (id, name, EmployeeAddress, age, dept, salary)

values (5,'Deepak','Delhi',23,'Sales',23000)

End

**--- THIS IS CALLING PART OF PROCEDURE**

exec InsertEmployee

create proc InsertEmployeeWithPara(@id int, @name varchar(30),

@address varchar(20), @age int , @dept varchar(20), @salary int)

As

Begin

insert into Employee (id, name, EmployeeAddress, age, dept, salary)

values (@id, @name, @address, @age,@dept,@salary)

End

**--- THIS IS CALLING PART OF PROCEDURE**

exec InsertEmployeeWithPara 6,'Lalit','O Delhi', 24, 'Sales',30000

create proc UpdateEmployee(@id int, @name varchar(30),

@address varchar(20), @age int , @dept varchar(20), @salary int)

As

Begin

Update Employee set name =@name , EmployeeAddress =@address, age =@age,

dept=@dept, salary=@salary where id=@id

End

**--- THIS IS CALLING PART OF PROCEDURE**

exec UpdateEmployee 3 ,'New Name', 'New Address', 22, 'Sales', 23000

create proc DeleteEmployee(@id int)

As

Begin

Delete from Employee where id=@id

End

-- Procedures can return values

-- 1. By using return statement

-- 2. By using output parameters

-- When you return somethong by using return statement, we can only return 1 value

-- and too is integer

-- When ypu ant to return more than one values , we shud use

-- output paramters

**--- PROCEDURE Returning value by using Return statement**

alter proc InsertEmployeeWithPara(@id int, @name varchar(30),

@address varchar(20), @age int , @dept varchar(20), @salary int)

As

Begin

if(exists(select \* from employee where id=@id))

return 0

else

begin

insert into Employee (id, name, EmployeeAddress, age, dept, salary)

values (@id, @name, @address, @age,@dept,@salary)

return 1

End

end

**--- THIS IS CALLING PART OF PROCEDURE**

-- Begin {

-- end }

declare @flag int

exec @flag= InsertEmployeeWithPara 8,'Lalit','O Delhi', 24, 'Sales',30000

if (@flag=0)

print 'Record wuth this ID is already there'

else

print 'record inserted'

**--- PROCEDURE Returning value by using Output statement**

create proc EmployeeDetailsById(

@id int ,

@name varchar(30) output,

@dept varchar(20) output,

@salary int output)

AS

Begin

Select @name = name , @dept = dept, @salary = salary from employee

where id=@id

End

**--- THIS IS CALLING PART OF PROCEDURE**

declare @name varchar(30)

declare @dept varchar(20)

declare @salary int

exec EmployeeDetailsById 5 , @name output , @dept output , @salary output

print @name

print @dept

print @salary

Index , Joins , Group By , Functions