use A

select \* from employee

select EmployeeAddress, count(\*) from Employee

group by EmployeeAddress

select EmployeeAddress, count(\*) from Employee

where salary is not null

group by EmployeeAddress

having count(\*) > 1

Triggers : Special type of stored procedure which is called automatically when we perform

Some DML or DDL command

DML Triggers : Triggers which are called when you perform DML (insert update delete)

DDL Triggers : Triggers which are called when you perform DDL (create alter drop)

DML after(for) , instead of

create trigger T1 on employee

for insert

as

begin

print 'rceord inserted'

end

insert into employee(id,name) values(11, 'Ajay')

select \* from salesdata

sp\_helptext salesdata

create view salesdata

As

Select \* from employee where dept = 'Sales'

insert into salesdata(id,name) values(11, 'Ajay')

alter trigger T1 on employee

for insert , update, delete

as

begin

rollback

print 'table not modified'

end

alter trigger T1 on employee

for insert , update, delete

as

begin

select \* from inserted

select \* from deleted

rollback

print 'table not modified'

end

alter trigger T1 on employee

for insert , update, delete

as

begin

--select \* from inserted

--select \* from deleted

print 'table not modified'

end

insert into employee(id,name) values(15, 'Ajay')

select \* from employee

select \* from salesdata

sp\_helptext salesdata

create view salesdata

As

Select \* from employee where dept = 'Sales'

insert into salesdata(id,name) values(11, 'Ajay')

declare @no integer

set @no=10

print 'Your marks are ' + Cast(@no As varchar(20))

create table AuditData(msg varchar(max))

select \* from AuditData

CREATE TRIGGER trInsertEmployee

ON Employee

FOR INSERT

AS

BEGIN

Declare @Id int

SELECT @Id = Id from inserted

INSERT INTO AuditData

VALUES ('New employee with Id = ' + CAST(@Id AS VARCHAR(10)) + ' is added at ' + CAST(Getdate() AS VARCHAR(22)))

END