# **nested query**

**# Q course offered in fll 2009 and not in spring 2010**

select course

from t1

where semester =”fall” and year =”2009” and

course not in (

select course

from t1

where semester =”spring” and year =”2010”)

**# Q course offered in fll 2009 and n spring 2010**

select course

from t1

where semester =”fall” and year =”2009” and

course in (

select course

from t1

where semester =”spring” and year =”2010”)

#  **“some” in sql (salary must be greater than some “instructor”)**

Select name

From t1

Where salary > **some(**

Select salary

From t1

Where name=”instructor”)

**# select instructor such that salary must be greater than some cse dept instructor**

Select instructor

From t1

Where salary > **some** (

Select salary

From t1

Where instructor =”cse”)

**# select instructor such that salary must be greater than cse dept instructor**

Select instructor

From t1

Where salary > **all** (

Select salary

From t1

Where instructor =”cse”)

# **exists** In dbms

Select name

From t1

Where year=2010 and

**Exists** (

Select name

From t1

Where year =2029)

# **not exist**s In dbms

Select name

From t1

Where year=2010 and

**Not Exists** (

Select name

From t1

Where year =2029)

#  **unique in sql (to check dublicay…..if dublicate columns found return false bu=y unique phrase)**

select name

from t1

where **unique** (

select name

from t1

where salary =1200 )

#--------------------------------------------

# subqueries in “from” phrase

**# average instructor salary of those department whw=ere average salary is greater than 12000**

Select dept , avg

From (

select dept,avrg(slary)

From t1

Group by dept)

Where avrg > 120000

**# select all department with maximum budget**

select dept ,max(budget)

from (

select dept, budget

from t1

group by dept)

group by dept

# **scaler subquery,return only 1 value**

**#Q dept with no of instructor in each dept**

Select dept (select count(\*)

From department

Where department.dept=instructor.dept) as dept\_count

From instructor

# **Deletion in sql**

delete from table1 # delelete all table

delete from table1

where class==a1

**# Q: delete instructor who is associated with abc company**

Delete from table1

Where dept in (select dept

From table2

Where buiding=”abc )

# **delete all instructors where slary is less than average salary**

Delete instructor

From instructor

Where salary< (select avg(salary)

From instructor )

# **insert a value in given table**

Insert into table1 (col1,col2,col3,col4)

Values(val1,val2,val3,val4)

**# inserting null value**

Insert into table1

Values()val1,val2,val3,null)

# **insertion using select**

Insert into table1

Select col,id,contact

From table2

#  **updation using update**

**M!-------**

Update table1

Set salary=salary\*1.03

Where salary > 100000

Update table1

Set salary=salary\*1.05

Where salary <= 100000

**M2------------**

Update table1

Set salary=case

When salary<= 1000

Then salary\*1.05

Else salary \* 1.05

End

**# use of case**

**Method-1**

Sum(col1)

**Method2**

Case

When sum(col1) is not null

then sum(col1)

Else 0

#  **case in sql**

**Case**

**When** condition1

**Then** impplemntation1

**Else**

Implemetation2

**end**

**#**  **Create view**

Create view view1 as

Select col1,col2

From table1

**# create view of total salary**

Create view total\_salary(user,id) as

Select id, sum(salary)

From table1

Where clh=”NIT”

Group by “id”

**# practice**

Create view new\_view as

Seleclt col1,col2,col3

From table1

Where condition1 and /or condition2

# practice2

Create view view1 as

(select col1, col2

From (

Select col3,col4

From t1,t2

Where cond1,cond2)

Where cond3)

#  **check in sql**

Create table t1 (

Col1 varchar(3),

Id char(3),

Semester varchar(5),

Primary key (col1),

Check (semester in (“val1”,”val2”,”val3”))) # here check(predicate) will check either all

# **integrity maintain**

Create table t1(

Col1 varchar(3) not null,

Col2 char(3),

Foreign key (col1)

Reference table2

**On delete cascade**

**On update cascade**) # when any changes happen in foreign key table parallel changes will implemented here too.

# **creating index**

**# task-1 create table**

Create table t1(

Col1 varchar(3),

Col2 numeric(2,4) defalult 33,

Primary key (id) )

# task-2 , address create index

**Create index new\_index\_col on table1(id)**

**# Sql functions**

**# below fi=unction will help to count of valuews I column 1**

Create function **function1**(value1 varchar(100))

Return integer

Begin

Declare output integer;

Select count(\*) into output

From table1

Where col1=value1

Return output

End

Q: find department name of all department with more than 12 instructor

Select dept, salary

From table1

Where **function1(dept)** > 12

#  **table function**

Create function func1(var1 varchar(20))

Return table (

Col1 varchar(2),

Col2 char(33))

Return table(

Select col1,col2

From table\_old

Where col\_nth = “var1”)

# usage of above function

Select \*

From table function1(adarsh)

# **procedure formation in sql**

Create procedure proc1 (

In var1 char(3)

Out output1 integer)

Begin (

Select count(\*) into output1

From table1

Where col1=var1)

End

# calling procedure

Declare ouput1 integer

Call proc1(“code\_adarsh,output1)

# **Decomposition in DBMS**

Table1(col1,col2,col3,col3)

Into

Table2(col1,col2)

Table3(col3,col4)

