**SOLUTIONS OF SQL TASK FOR INTERN:**

QUES:1:

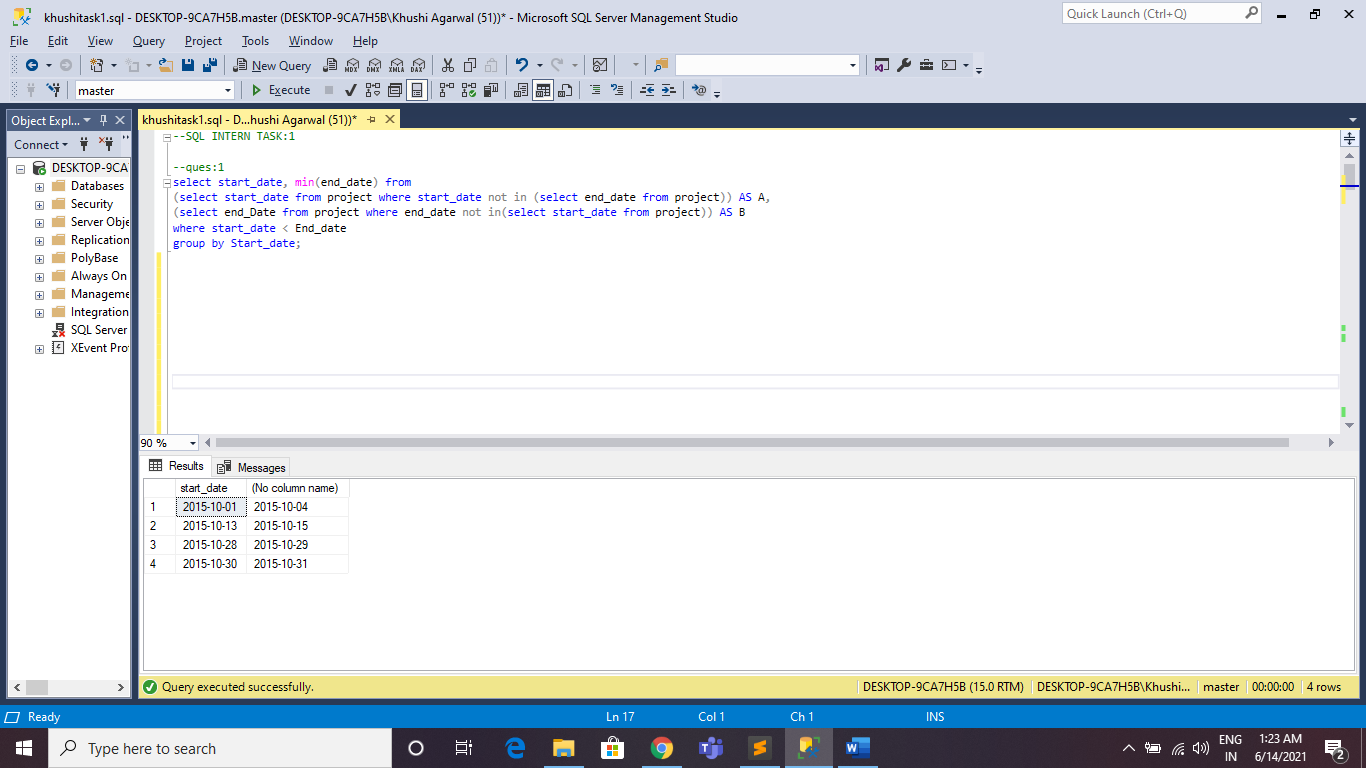
select start\_date, min(end\_date) from

(select start\_date from project where start\_date not in (select end\_date from project)) AS A,

(select end\_Date from project where end\_date not in(select start\_date from project)) AS B

where start\_date < End\_date

group by Start\_date;



QUES:2:

select top 3 \* from students as s1

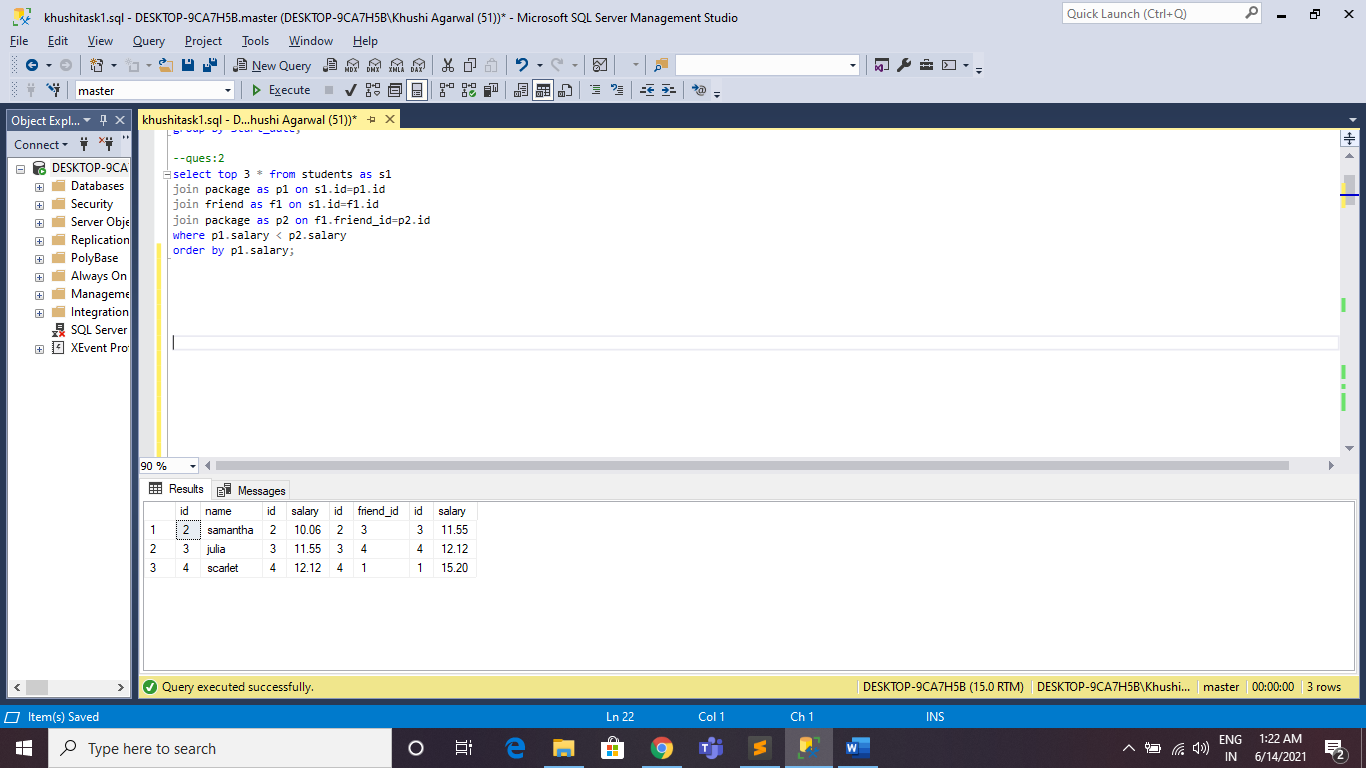
join package as p1 on s1.id=p1.id

join friend as f1 on s1.id=f1.id

join package as p2 on f1.friend\_id=p2.id

where p1.salary < p2.salary

order by p1.salary;



QUES:3:

SELECT f1.X, f1.Y FROM Functions AS f1

WHERE f1.X = f1.Y AND

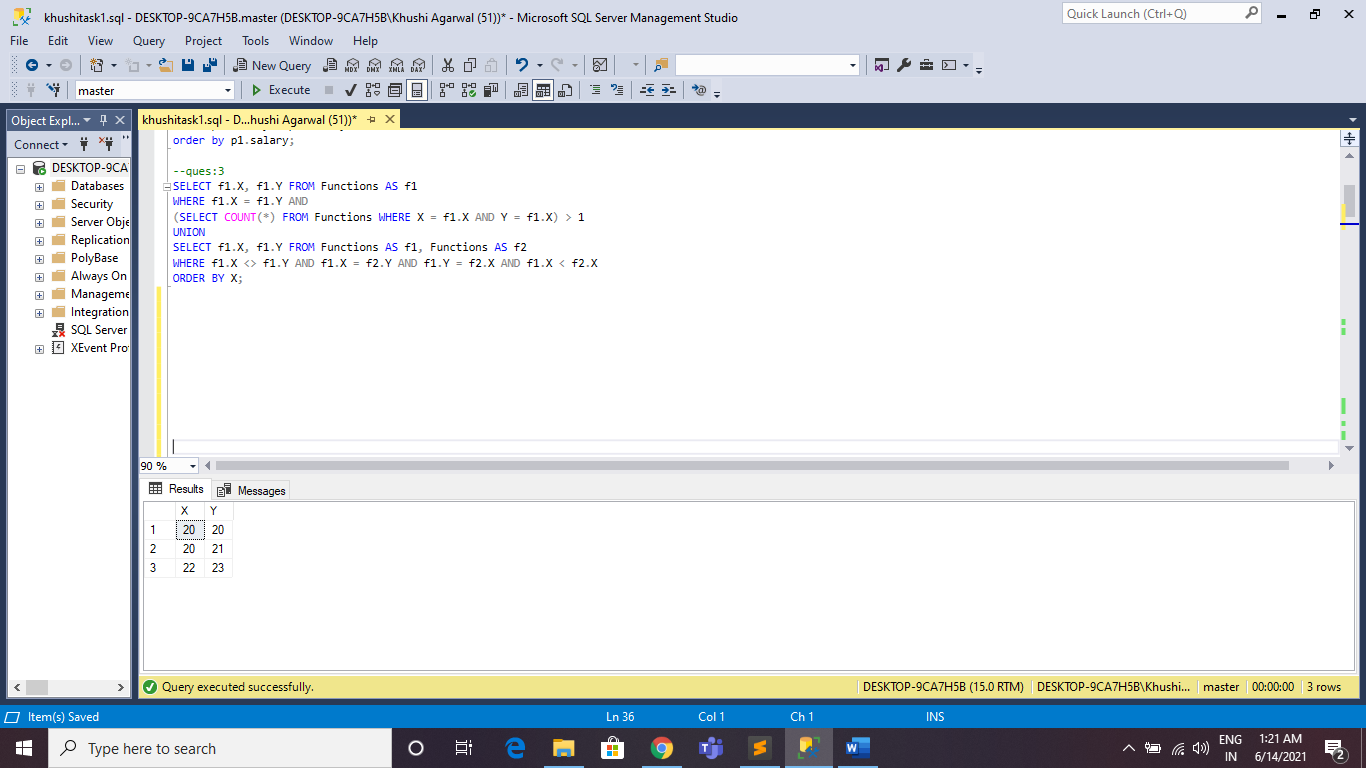
(SELECT COUNT(\*) FROM Functions WHERE X = f1.X AND Y = f1.X) > 1

UNION

SELECT f1.X, f1.Y FROM Functions AS f1, Functions AS f2

WHERE f1.X <> f1.Y AND f1.X = f2.Y AND f1.Y = f2.X AND f1.X < f2.X

ORDER BY X;



QUES:4:

SELECT con.contest\_id, con.hacker\_id, con.name,

SUM(sg.total\_submissions), SUM(sg.total\_accepted\_submissions),

SUM(vg.total\_views), SUM(vg.total\_unique\_views)

FROM Contests AS con

JOIN College AS col ON con.contest\_id = col.contest\_id

JOIN Challenges AS cha ON cha.college\_id = col.college\_id

LEFT JOIN

(SELECT ss.challenge\_id, SUM(ss.total\_submissions) AS total\_submissions, SUM(ss.total\_accepted\_submissions) AS total\_accepted\_submissions

FROM Submission\_Stats AS ss GROUP BY ss.challenge\_id) AS sg

ON cha.challenge\_id = sg.challenge\_id

LEFT JOIN

(SELECT vs.challenge\_id, SUM(vs.total\_views) AS total\_views, SUM(vs.total\_unique\_view) AS total\_unique\_views

FROM View\_Stats AS vs GROUP BY vs.challenge\_id) AS vg

ON cha.challenge\_id = vg.challenge\_id

GROUP BY con.contest\_id, con.hacker\_id, con.name

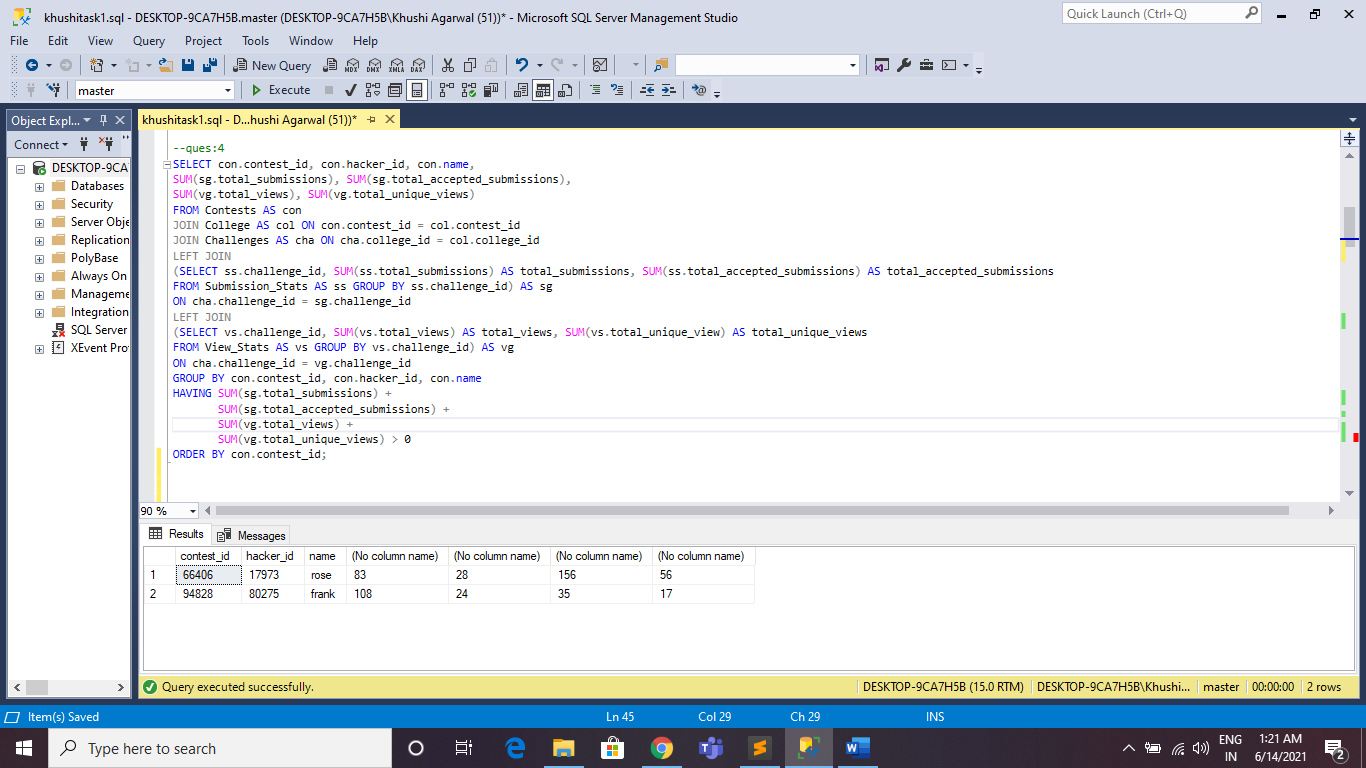
HAVING SUM(sg.total\_submissions) +

SUM(sg.total\_accepted\_submissions) +

SUM(vg.total\_views) +

SUM(vg.total\_unique\_views) > 0

ORDER BY con.contest\_id;



QUES:5:

WITH Q1 AS (

SELECT sdate, COUNT(DISTINCT h\_id) unique\_count

FROM (

SELECT DISTINCT(T0.h\_id), T0.sdate,

ROW\_NUMBER() OVER(PARTITION BY T0.h\_id ORDER BY T0.sdate) subdate\_rowno

FROM (SELECT sdate, h\_id FROM Submission GROUP BY sdate, h\_id )T0

) T1

WHERE T1.subdate\_rowno >= DAY(T1.sdate)

GROUP BY sdate),

Q2 AS (

SELECT sdate, submission\_count, h\_id,

ROW\_NUMBER() OVER (PARTITION BY sdate ORDER BY submission\_count DESC, h\_id ASC) Rank

FROM (SELECT sdate, COUNT(sdate) as submission\_count, h\_id FROM Submission GROUP BY sdate, h\_id) T3

)

SELECT Q2.sdate, Q1.unique\_count, Q2.h\_id, H.name

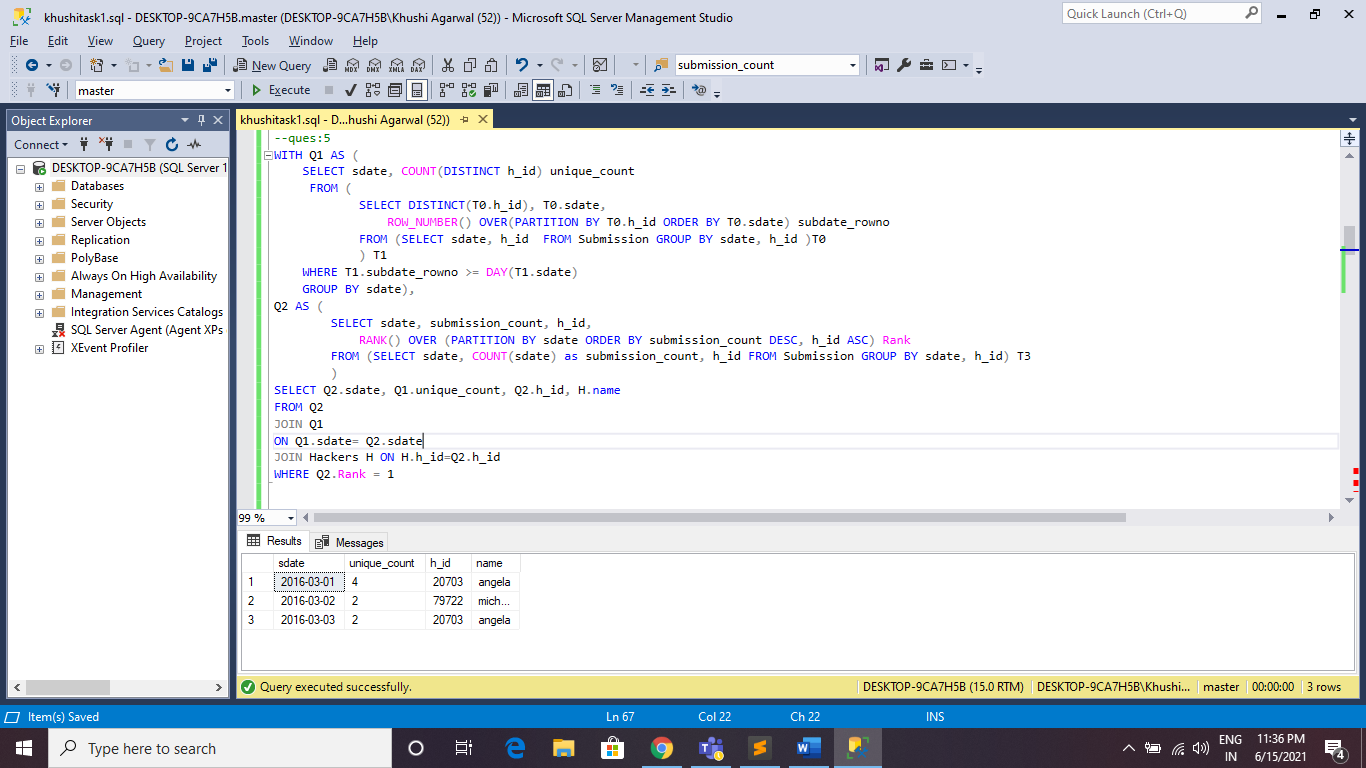
FROM Q2

JOIN Q1

ON Q1.sdate= Q2.sdate

JOIN Hackers H ON H.h\_id=Q2.h\_id

WHERE Q2.Rank = 1



QUES:6:

SELECT ROUND(ABS(MIN(lat\_n)-MAX(lat\_n)) + ABS(MIN(long\_w)-MAX(long\_w)), 4) FROM STATION;



QUES:7:

DECLARE @OUT TABLE (NUMBER VARCHAR(256));

DECLARE @B VARCHAR(256);

DECLARE @FLAG int = 0;

DECLARE @I int = 2;

WHILE @I<=1000

BEGIN

DECLARE @J INT= @I-1

SET @FLAG=1

WHILE @J>1

BEGIN

IF @I % @J =0

BEGIN

SET @FLAG=0

END

SET @J= @J-1

END

IF @FLAG =1

BEGIN

DECLARE @TMP VARCHAR(4) = CAST(CONCAT(@I,'&') AS VARCHAR)

SET @B = CONCAT(@B,@TMP)

--INSERT @OUT VALUES (@I)

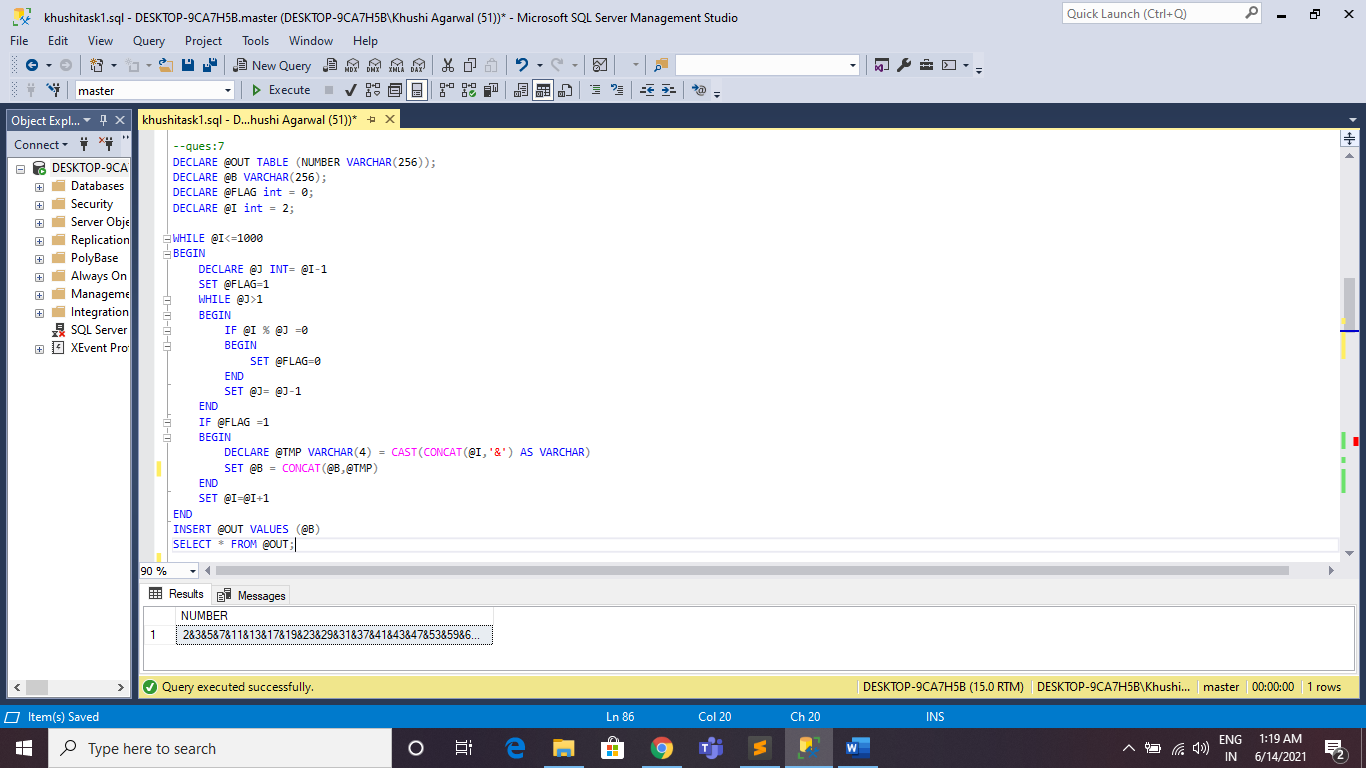
END

SET @I=@I+1

END

INSERT @OUT VALUES (@B)

SELECT \* FROM @OUT;



QUES:8:

select doctor,professor,singer,actor from

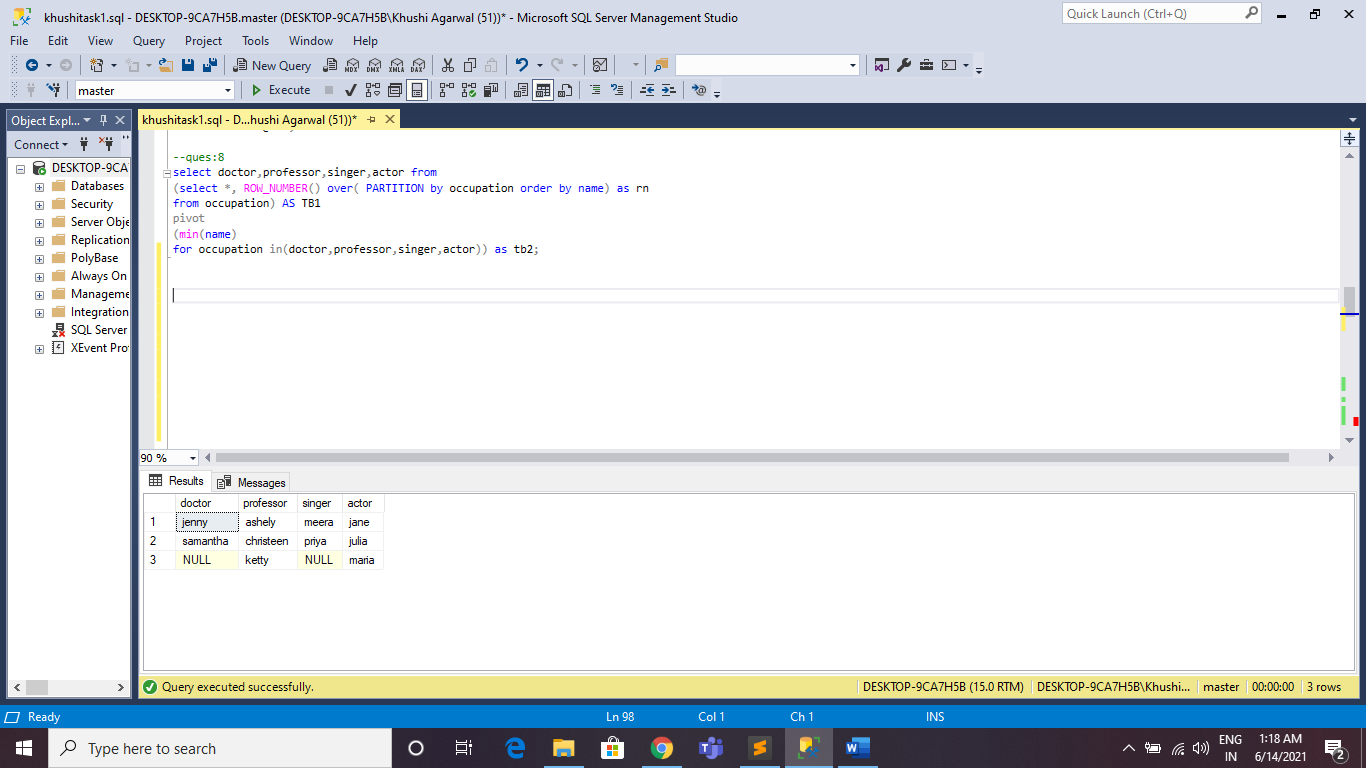
(select \*, ROW\_NUMBER() over( PARTITION by occupation order by name) as rn

from occupation) AS TB1

pivot

(min(name)

for occupation in(doctor,professor,singer,actor)) as tb2;



QUES:9:

SELECT CASE

WHEN P IS NULL THEN CONCAT(N, ' Root')

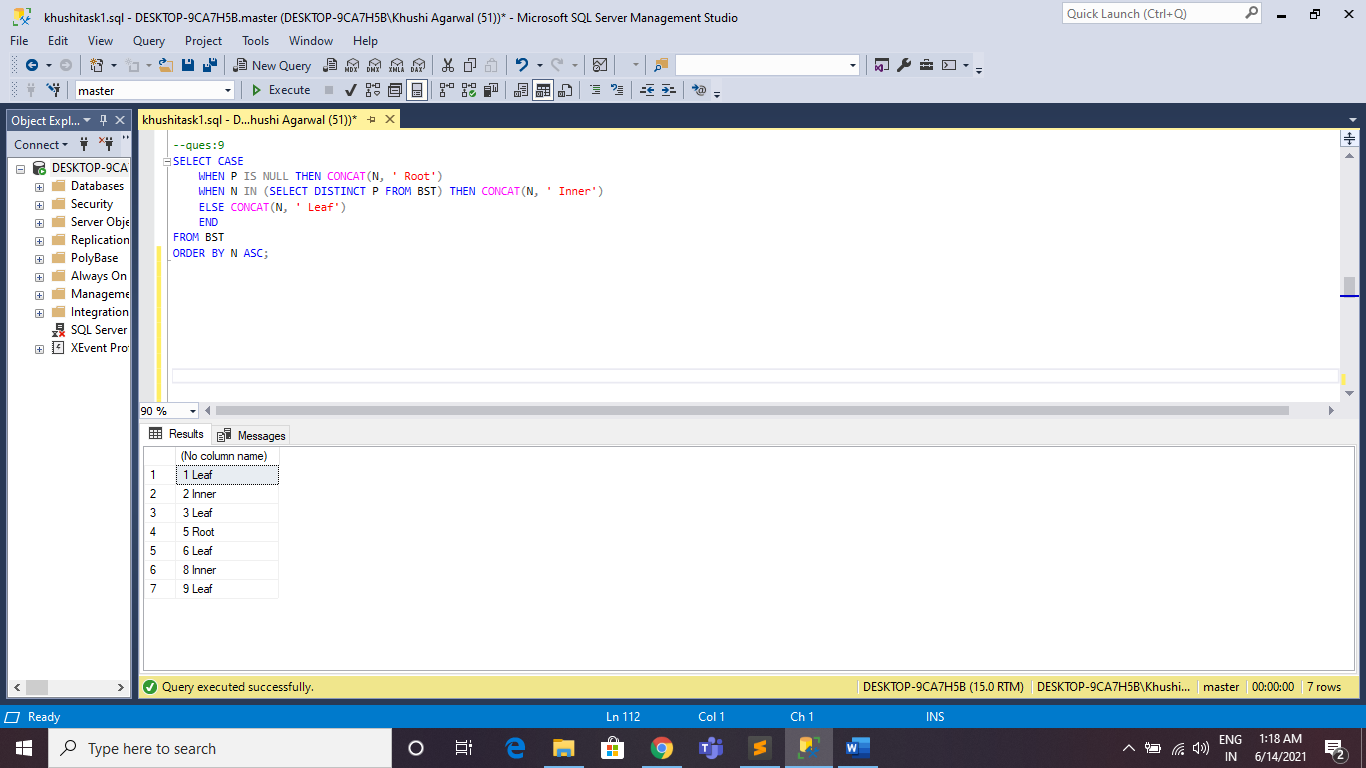
WHEN N IN (SELECT DISTINCT P FROM BST) THEN CONCAT(N, ' Inner')

ELSE CONCAT(N, ' Leaf')

END

FROM BST

ORDER BY N ASC;



QUES:10:

select co.code,founder,count(distinct l.code) as 'no. of lead m', count(distinct s.s\_code) as 'no. of senior m',

count(distinct m.m\_code) as 'no. of manager', Count(distinct e\_code) as 'no. of emp'

from company as co

join leadmanager as l on co.code= l.code

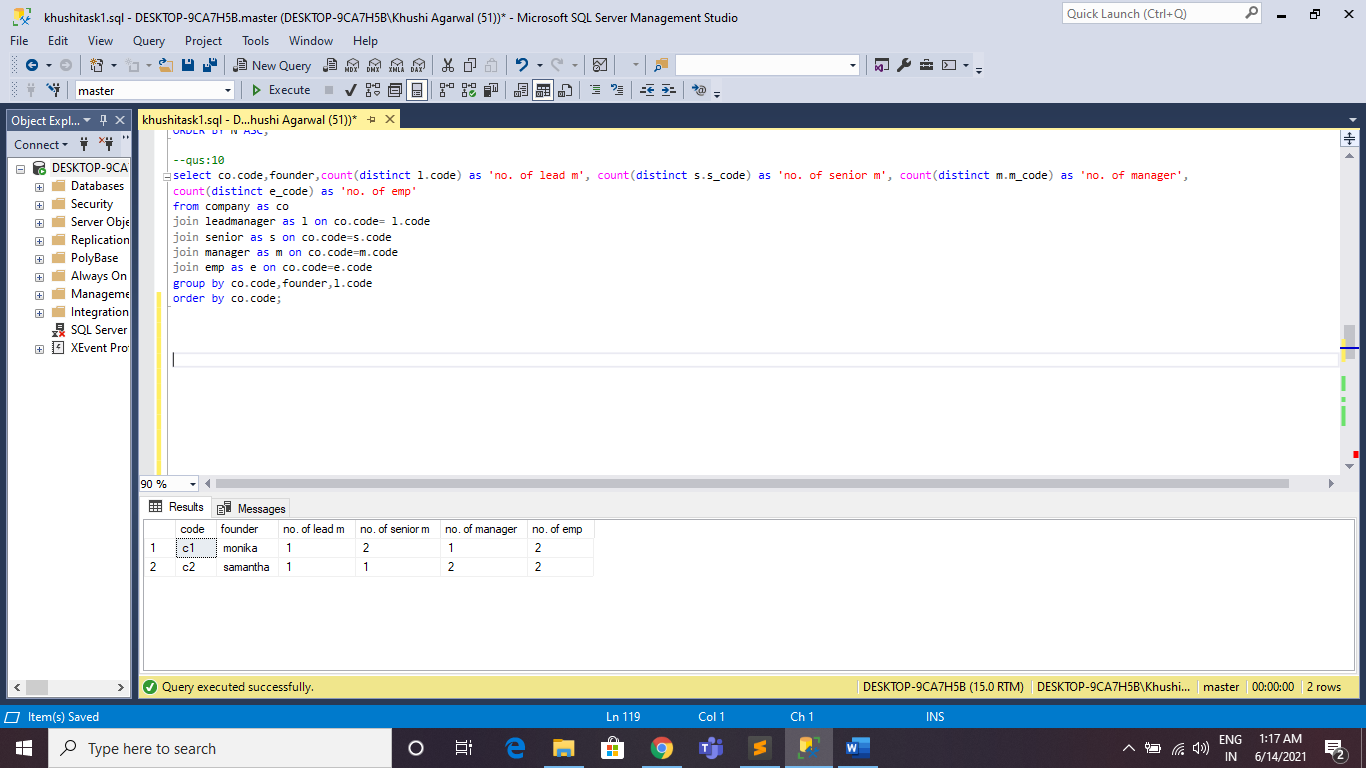
join senior as s on co.code=s.code

join manager as m on co.code=m.code

join emp as e on co.code=e.code

group by co.code,founder,l.code

order by co.code;



QUES:11:

select name from students as s1

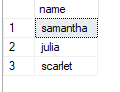
join package as p1 on s1.id=p1.id

join friend as f1 on s1.id=f1.id

join package as p2 on f1.friend\_id=p2.id

where p1.salary < p2.salary

order by p1.salary;



QUES:12:

DECLARE @TABLE TABLE(NUMBER INT);

DECLARE @AB INT= (SELECT max(salary) FROM employee);

DECLARE @N INT =1;

WHILE @N<=5

BEGIN

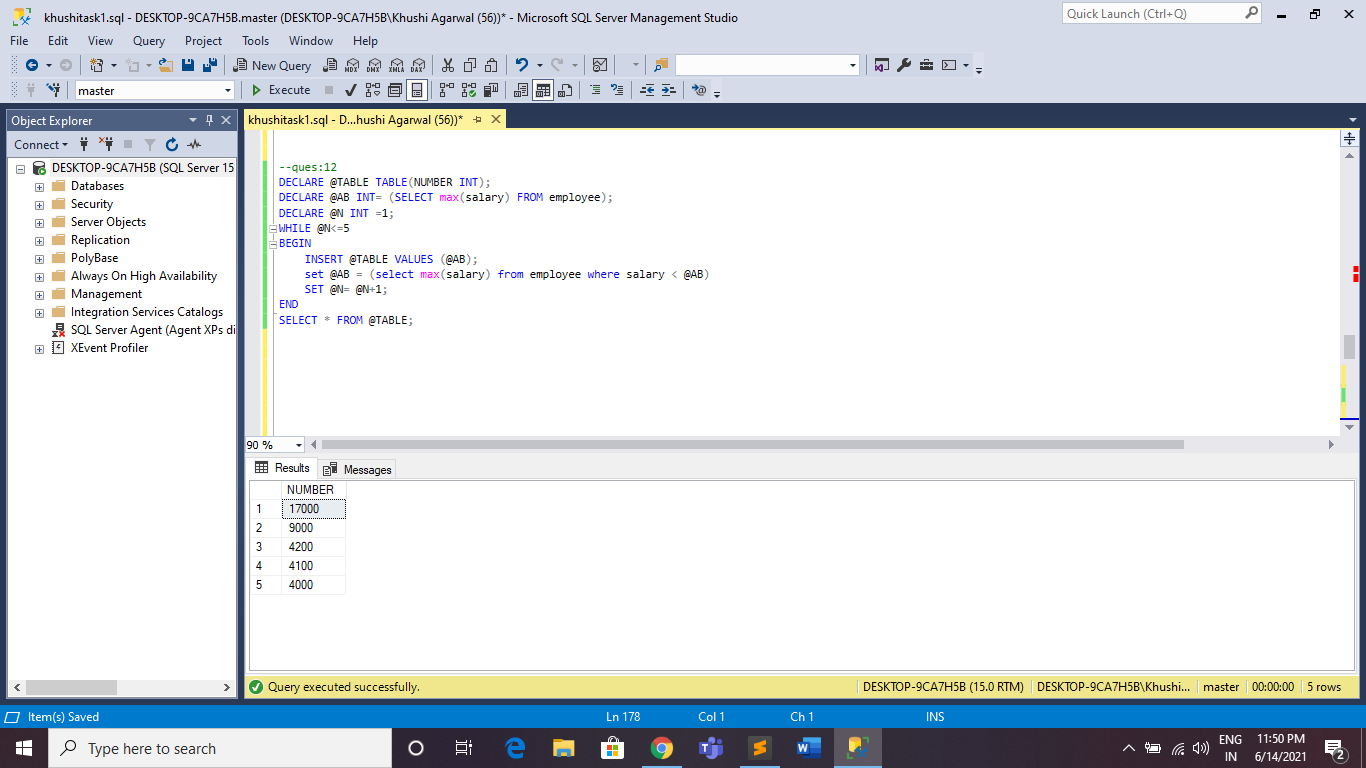
INSERT @TABLE VALUES (@AB);

set @AB = (select max(salary) from employee where salary < @AB)

SET @N= @N+1;

END

SELECT \* FROM @TABLE;

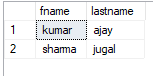


QUES:13:

update name

set fname=lastname, lastname=fname;

select \* from name;



QUES:15:

select cast(CEILING(avg(cast(salary as float)) - avg(cast(replace(salary,0,'') as float))) as int) from employees;



QUES:16:

merge into TAB2 as Target

using TAB1 as Source

on Target.id = Source.id

when matched and (Target.name <> Source.name or Target.AGE <> Source.AGE) then

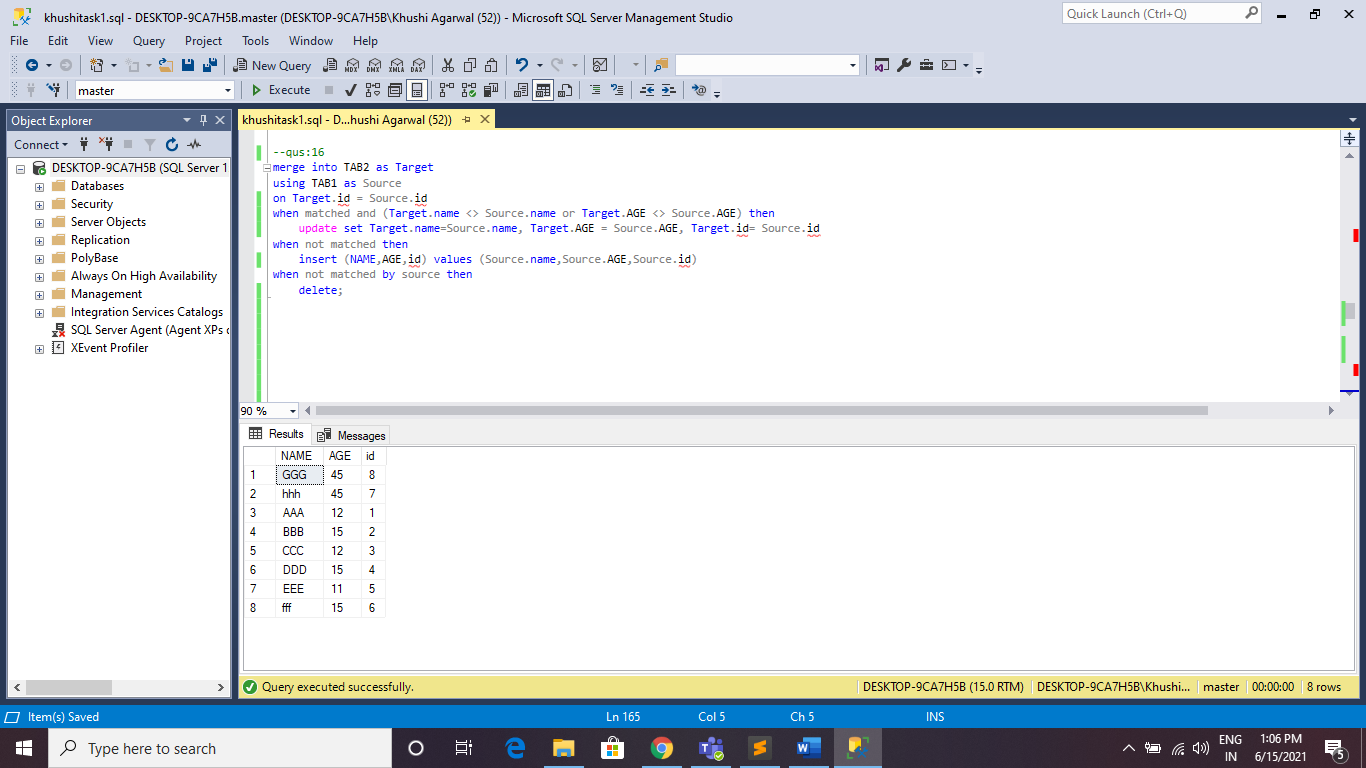
update set Target.name=Source.name, Target.AGE = Source.AGE, Target.id= Source.id

when not matched then

insert (NAME,AGE,id) values (Source.name,Source.AGE,Source.id)

when not matched by source then

delete;



QUES:17:

CREATE PROCEDURE DIMDATE AS

drop table dimdate1

create table dimdate1

(date date,

month int,

year int);

DECLARE @StartDate date = '2021-06-17';

DECLARE @CutoffDate date = DATEADD(DAY, 1, DATEADD(YEAR, 25, @StartDate))

declare @i int = 1

while @i <= (SELECT DATEDIFF(day, @StartDate,@CutoffDate))

begin

DECLARE @UP DATE = dateadd(day,@i,@StartDate)

insert into dimdate1

values (@UP, DATEPART(MONTH,@UP), DATEPART(YEAR,@UP))

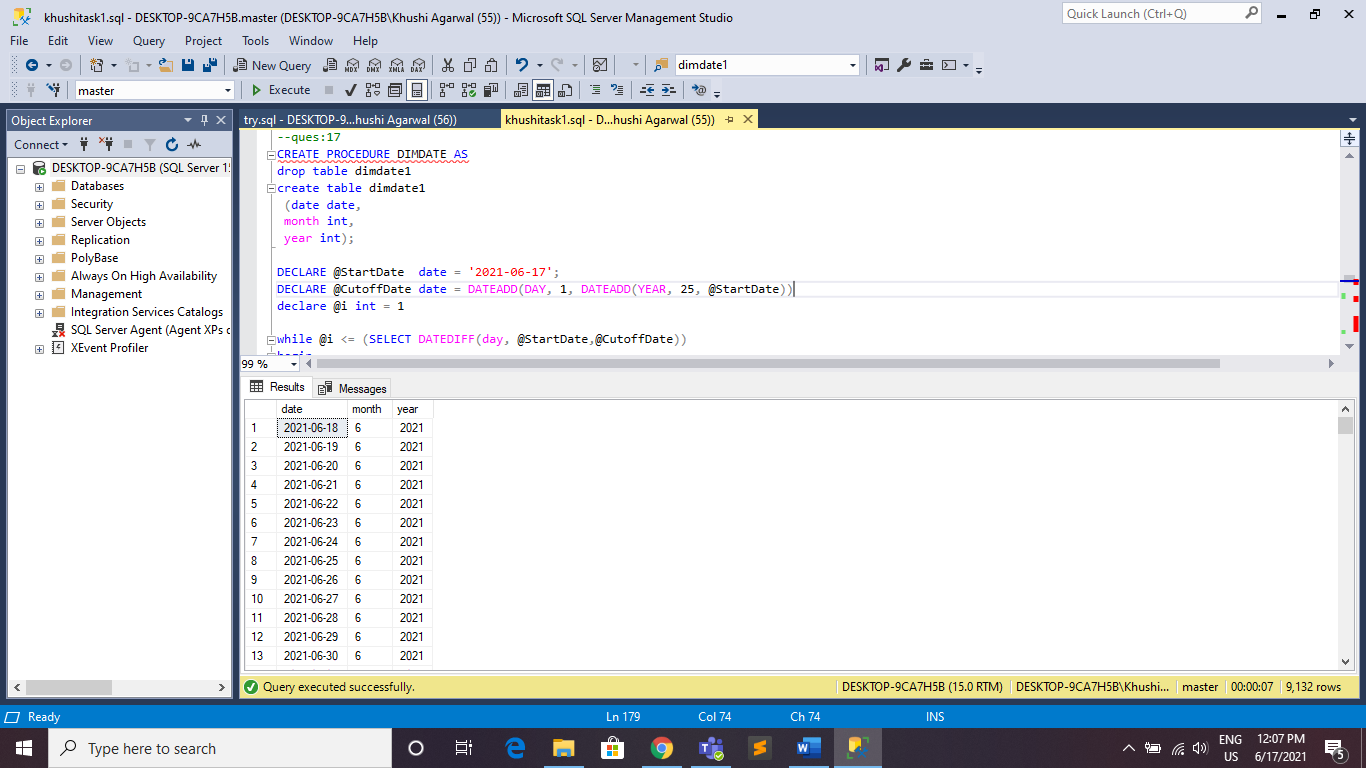
set @i = @i +1

end

select \* from dimdate1

--stored procedure called

EXEC DIMDATE;



QUES:18:

create procedure dt as

drop table dimtable1;

create table dimtable1

(time int,

timealt time,

hour int,

minute int,

second int);

declare @start time(0) = '00:00:00'

declare @end time(0) = '23:00:00'

declare @tk time(0)= @start

declare @interval int = 1

declare @i int =0;

while @i < 24

begin

insert into dimtable1

values(@interval,@tk, datename(hh,@tk), datename(minute,@tk), datename(second,@tk))--convert(varchar(10),@tk)

select @tk = dateadd(hh,1,@tk)

select @interval = @interval + 1

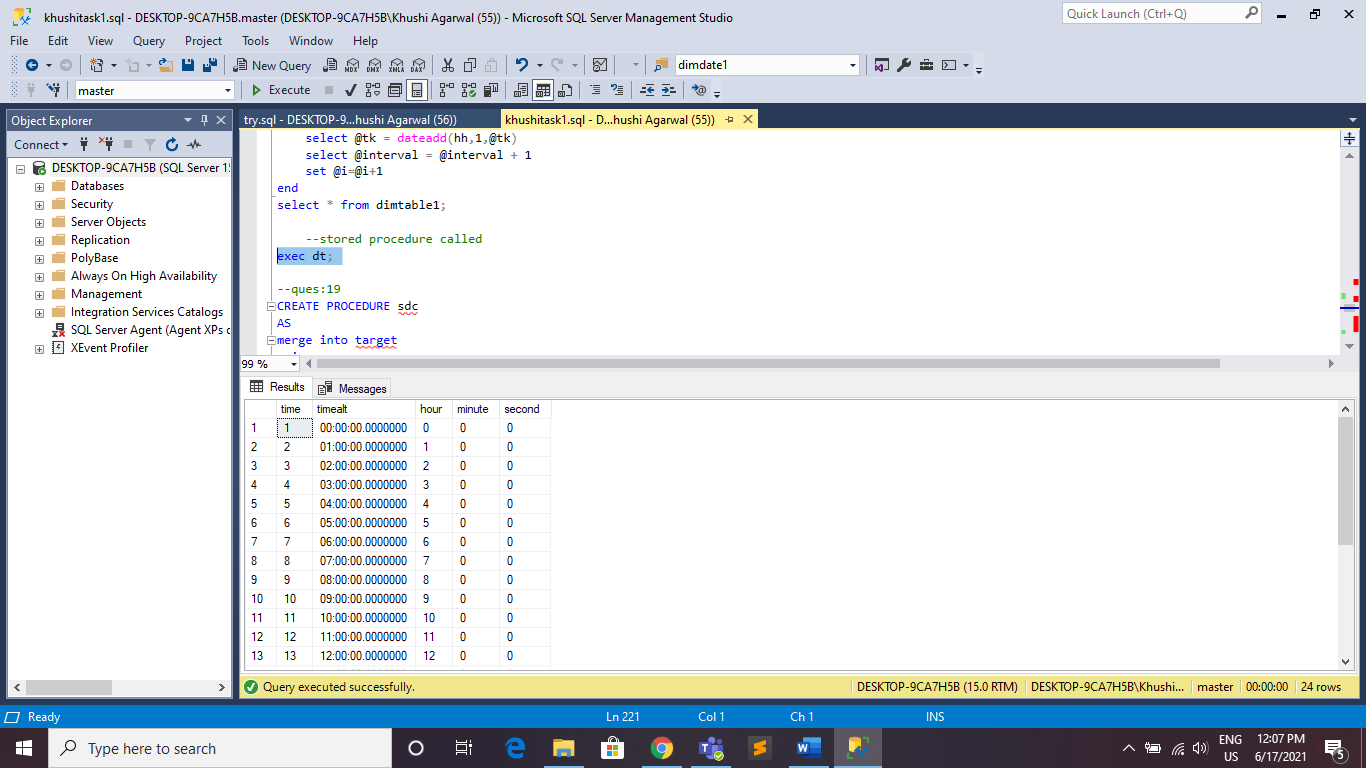
set @i=@i+1

end

select \* from dimtable1;

--stored procedure called

exec dt;



QUES:19:

CREATE PROCEDURE sdc

AS

merge into target

using source

on source.id = target.id

when matched and (target.name <> source.name or target.age <> source.age) then

update set target.id= source.id, target.name=source.name, target.age=source.age

when not matched then

insert (id,name,age) values(source.id, source.name, source.age)

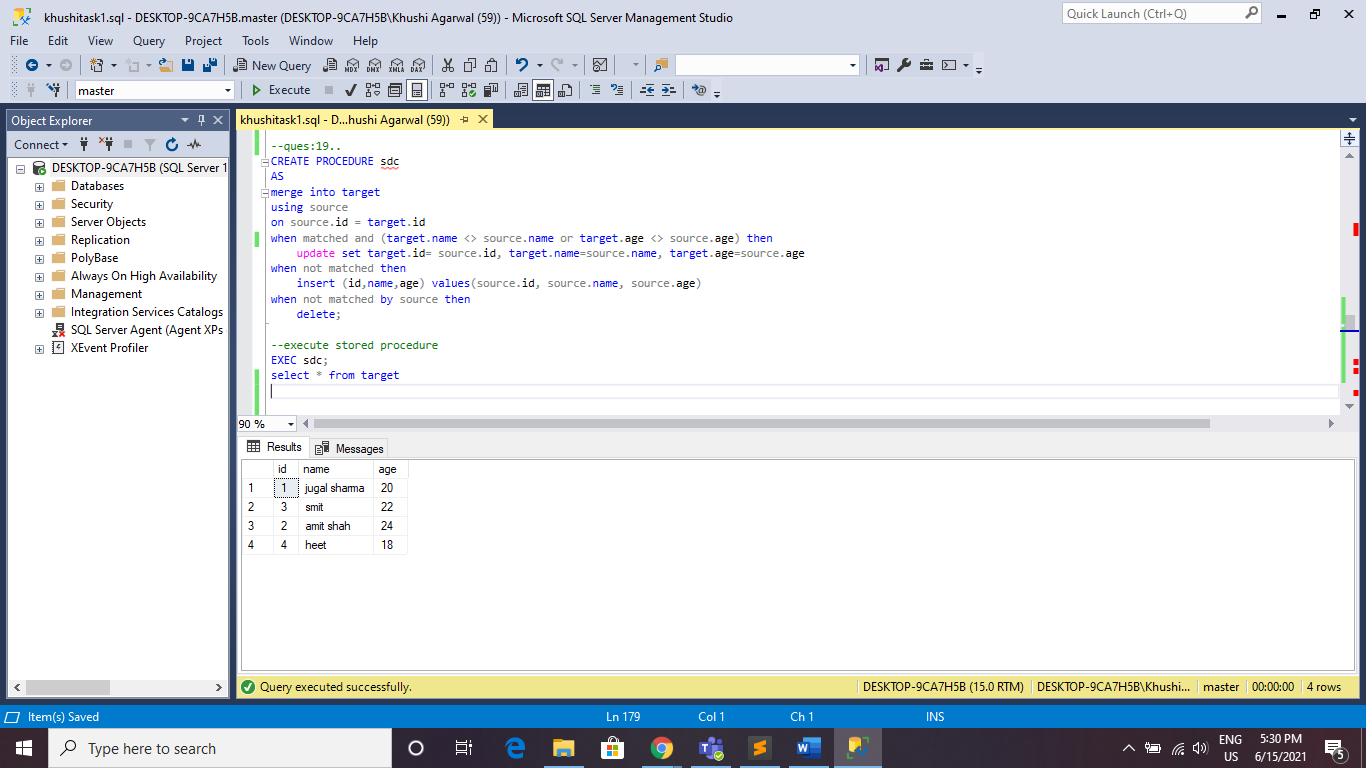
when not matched by source then

delete;

--execute stored procedure

EXEC sdc;

select \* from target



QUES:20:

CREATE PROCEDURE SCD2 AS

UPDATE D

SET D.FLAG ='N', D.TO\_DATE= DATEADD(DAY,-1,S.START\_DATE)

FROM SS AS S

INNER JOIN DW AS D

ON S.ID=D.ID AND D.flag = 'Y' AND ( S.SALARY <> D.SALARY OR S.OFFICE <> D.OFFICE )

INSERT INTO DW

SELECT SS.ID, SS.NAME, SS.SALARY, SS.OFFICE, SS.START\_DATE, NULL, 'Y'

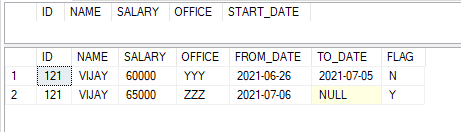
FROM SS

DELETE FROM SS

SELECT \* FROM SS; SELECT \* FROM DW;

--EXCEUTE STORED PROCEDURE

EXEC SCD2;



QUES:21:

select \* from configuration

declare @i int = @@rowcount

declare @dist table(id int identity(1,1), name varchar(20))

insert into @dist select distinct(tables) from configuration

--select \* from @dist

declare @c int = (select count(\*) from @dist)

declare @j int =1

while @i > 0 --main loop

begin

while @j <= @c --dist loop

begin

create table tmp (id int identity(1,1), newcol varchar(20), datatype varchar(20), tables varchar(20), tabletype varchar(20), tablestatus varchar(20), ifnew varchar(20), cli varchar(20), ncli varchar(20))

insert tmp select newcol,datatype,tables,tabletype,tablestatus, ifnew,cli,ncli from configuration where tables in (select name from @dist where id= @j)

select \* from tmp

declare @tablename varchar(100)= (select tables from tmp where id= 1)

declare @count int =(select count(newcol) from tmp where tablestatus= 'new')

if @count > 0

begin

declare @colname varchar(100) = ''

while @count > 0 --FOR COLONAME

begin

declare @append varchar(100)= (select concat(newcol,' ',datatype) from tmp where id=@count)

if @count > 1

begin

set @colname= (select concat(@append,', ',@colname))

end

else

begin

set @colname= (select concat(@colname,' ',@append))

end

set @count= @count-1

end

--PRIMARY KEY CHECK:

create table samp (id int identity(1,1), col1 varchar(20))

insert into samp select newcol from tmp where ifnew='yes'

DECLARE @CPK int=(SELECT count(newcol) FROM tmp where ifnew ='yes')

declare @pk varchar(100) =''

while @cpk > 0

begin

if @cpk > 1

begin

set @pk= (select concat(col1,', ',@pk) from samp where id =@cpk)

end

else

begin

set @pk= (select concat(@pk,' ',col1) from samp where id =@cpk)

end

set @cpk= @cpk -1

end

drop table samp

--CLUSTER INDEX CHECK:

create table sampCi(id int identity(1,1), col1 varchar(20))

insert into sampCi select newcol from tmp where CLI='yes'

DECLARE @CCI int=(SELECT count(newcol) FROM tmp where CLI ='yes' )

declare @CI varchar(100) = ''

while @CCI > 0

begin

if @CCI > 1

begin

set @CI= (select concat(col1,', ',@CI) from sampCi where id =@CCI)

end

else

begin

set @CI= (select concat(@CI,' ',col1) from sampCi where id =@CCI)

end

set @CCI= @CCI -1

end

drop table sampCi

--NON CLUSTERED INDEX:

create table sampNci (id int identity(1,1), col1 varchar(20))

insert into sampNci select newcol from tmp where NCLI='yes'

DECLARE @ncount int=(SELECT count(newcol) FROM tmp where NCLI ='yes' )

declare @NCI varchar(100) =''

while @ncount > 0

begin

if @ncount > 1

begin

set @NCI= (select concat(col1,', ',@NCI) from sampNci where id =@ncount)

end

else

begin

set @NCI= (select concat(@NCI,' ',col1) from sampNci where id =@ncount)

end

set @ncount= @ncount -1

end

drop table sampNci

--COMBINATION OF PK, CI, NCI

print('pk,ci,nci are:')

print(@pk)

print(@ci)

print(@nci)

declare @tableStr varchar(max)

if (@pk != '' and @ci != '' and @nci != '')

begin

set @tableStr = (select concat('create table ', @tablename,'(',@colname,', primary key(',@pk,'), index CLI clustered ('+@ci+'), index NCLI nonclustered ('+@nci+'))'))

end

else if(@pk != '' and @ci != '' and @nci = '')

begin

set @tableStr = (select concat('create table ', @tablename,'(',@colname,', primary key(',@pk,'), index CLI clustered ('+@ci+'))'))

end

else if(@pk != '' and @ci = '' and @nci != '')

begin

set @tableStr = (select concat('create table ', @tablename,'(',@colname,', primary key(',@pk,'), index NCLI nonclustered ('+@nci+'))'))

end

else if(@pk != '' and @ci = '' and @nci = '')

begin

set @tableStr = (select concat('create table ', @tablename,'(',@colname,', primary key(',@pk,'))'))

end

else if(@pk = '' and @ci != '' and @nci != '')

begin

set @tableStr = (select concat('create table ', @tablename,'(',@colname,', index CLI clustered ('+@ci+'), index NCLI nonclustered ('+@nci+'))'))

end

else if(@pk = '' and @ci != '' and @nci = '')

begin

set @tableStr = (select concat('create table ', @tablename,'(',@colname,', index CLI clustered ('+@ci+'))'))

end

else if(@pk = '' and @ci = '' and @nci != '')

begin

set @tableStr = (select concat('create table ', @tablename,'(',@colname,', index NCLI nonclustered ('+@nci+'))'))

end

else if(@pk = '' and @ci = '' and @nci = '')

begin

set @tableStr = (select concat('create table ', @tablename,'(',@colname,')'))

end

exec (@tableStr)

print(@tableStr)

end

--for old :alter

declare @countold int =(select count(newcol) from tmp where tablestatus= 'old')

if @countold > 0

begin

declare @colnameold varchar(100) = ''

while @countold > 0

begin

declare @appendold varchar(100)= (select concat(newcol,' ',datatype) from tmp where tablestatus='old')

if @countold > 1

begin

set @colnameold= (select concat(@appendold,', ',@colnameold)) --a= typevar , idvarcahr , type

end

else

begin

set @colnameold= (select concat(@colnameold,' ',@appendold))

end

set @countold= @countold-1

end

declare @tableOld varchar(max)= (select concat('alter table ', @tablename, ' add ', @colnameold ))

print(@tableOld)

exec (@tableOld)

--flag operation:

insert into tmp select newcol,datatype, tables, tabletype, tablestatus, ifnew,cli,ncli from configuration

where tables in (select name from @dist) and flag='N'

update configuration set flag='Y' where newcol in (select newcol from tmp)

end

drop table tmp

set @j = @j +1

end

set @i = @i -1

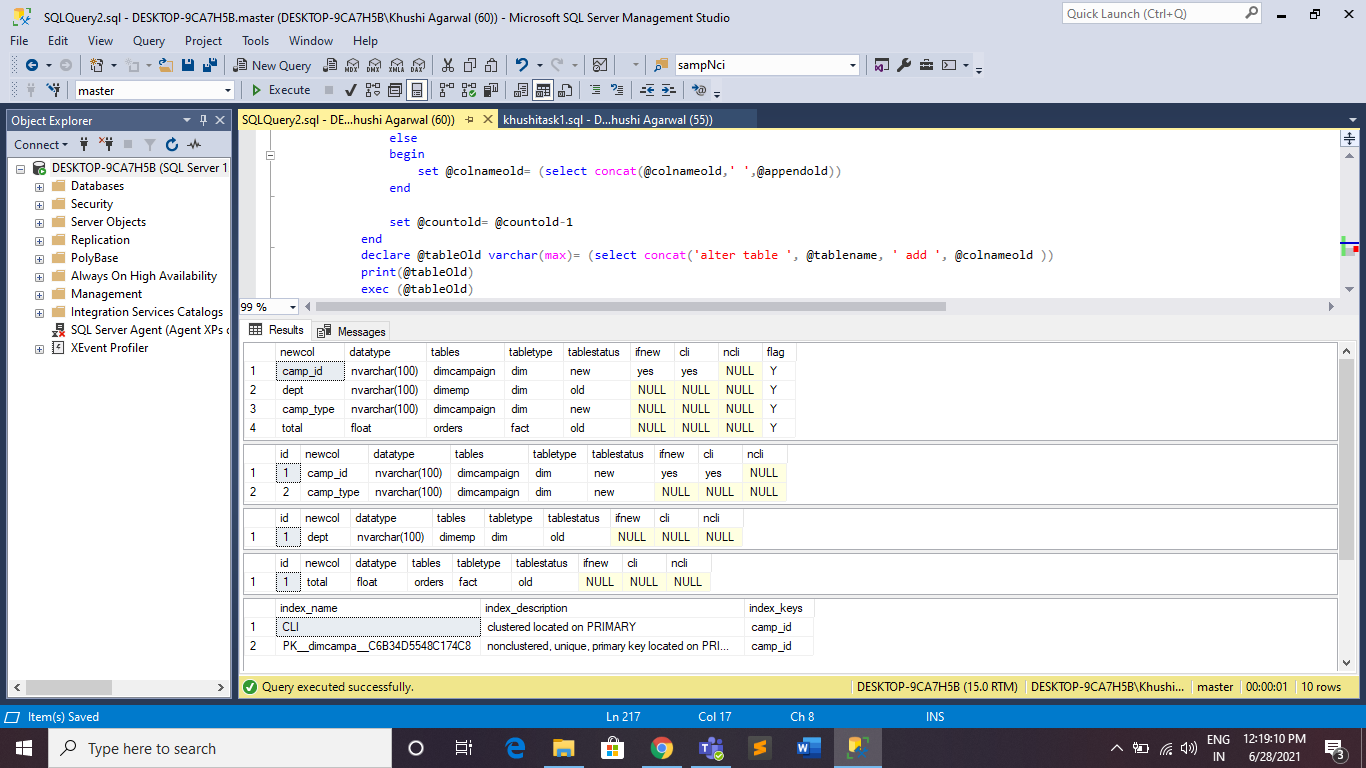
end

EXECUTE sp\_helpindex dimcampaign

drop table dimcampaign

alter table dimemp drop column dept

alter table orders drop column total



QUES:22:

select id, min(checkin\_out) as 'check-in',max(checkin\_out) as 'check-out'

from cust\_details

group by id order by id;

