Slot: L27 + L28 **Name: Swaranjana Nayak**
Date: 06/11/2020 **Reg. No.: 19BCE0977**

CSE2004 LAB FAT

Aim:
To solve the given problems by implementing in SQL.

QUESTION NO. 3

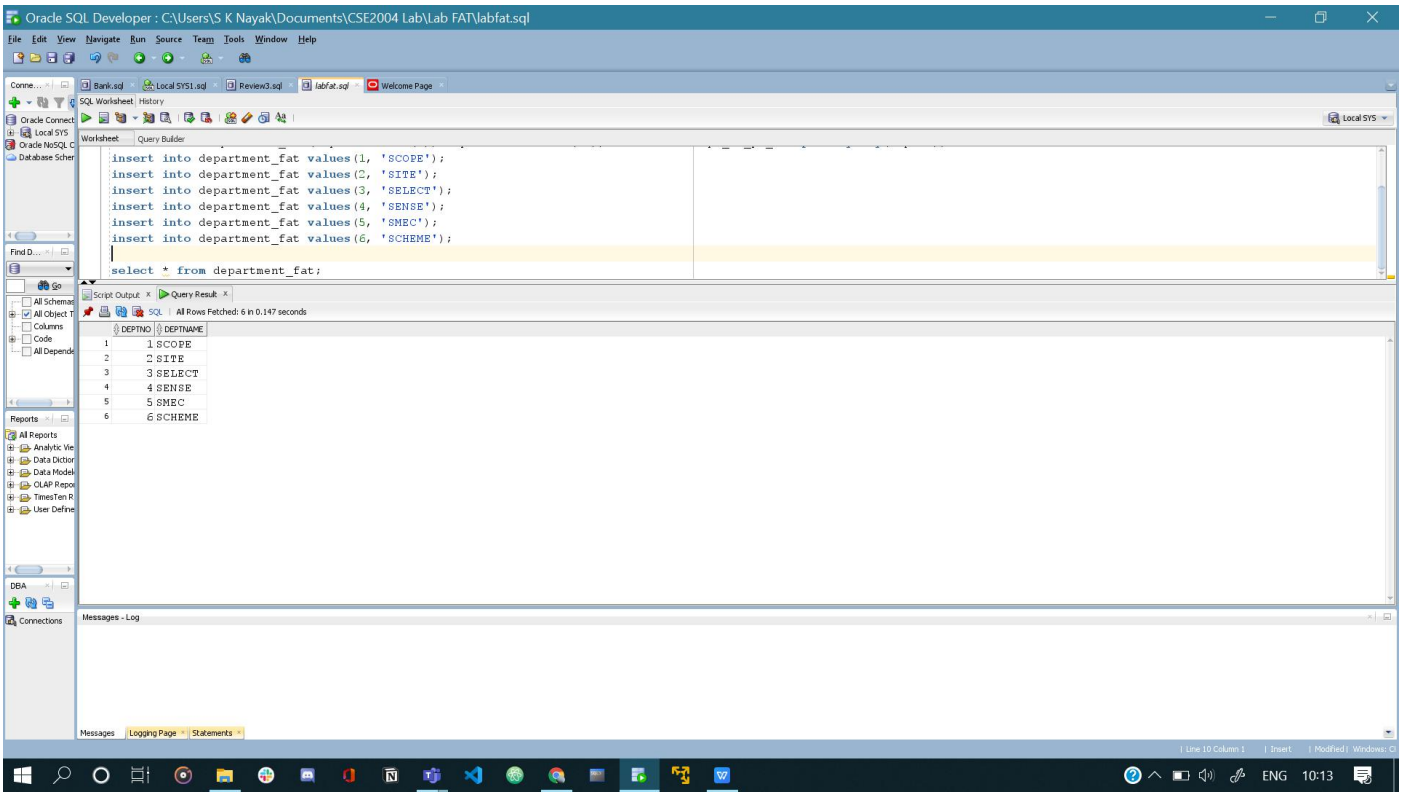
Note: The following queries are performed in Oracle's SQL Developer software, instead of command line interface. Rest all queries are same, just output will be in the software's GUI.

Note: There were already such named tables in LOCAL SYS, hence tablename_fat was used to name the table.

1. & 2. Table creation and insertion

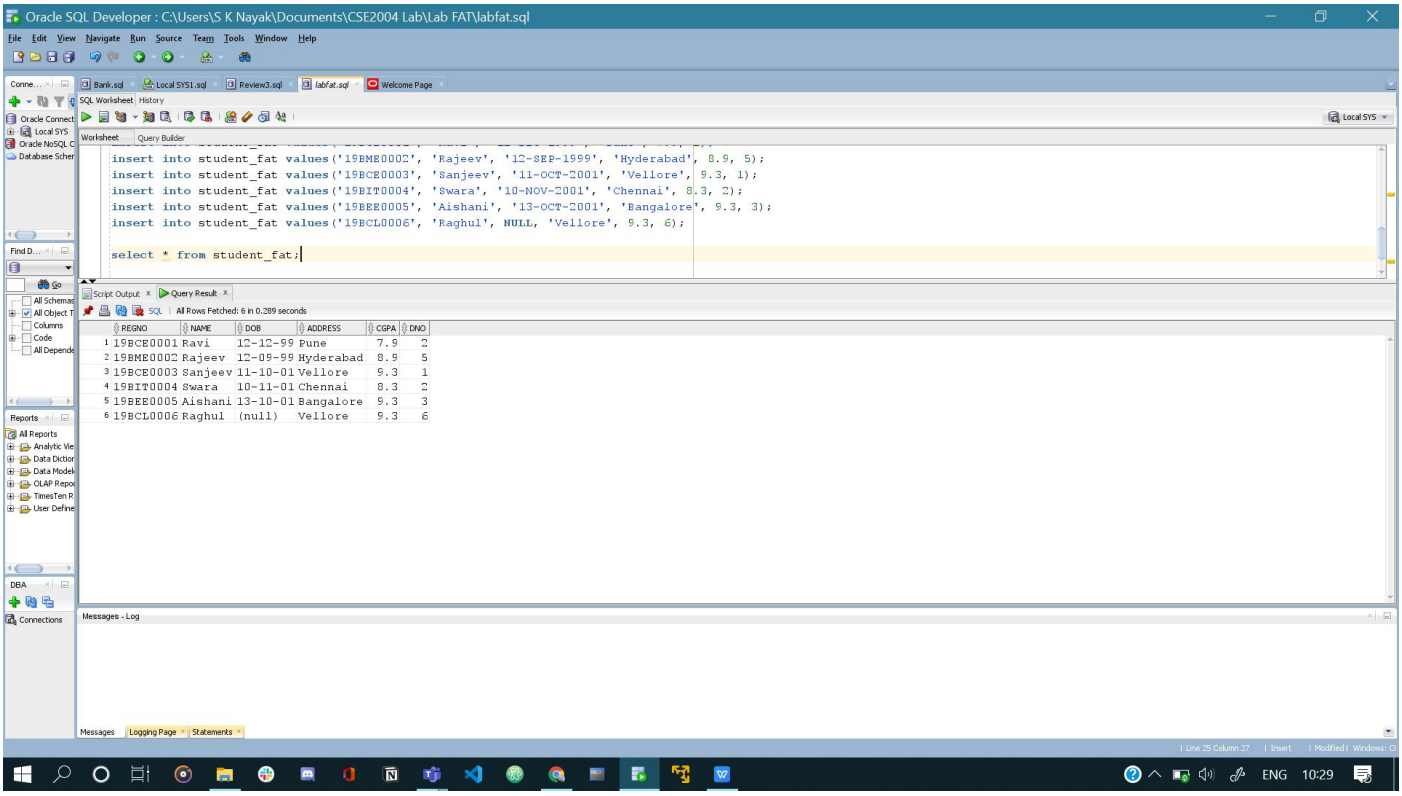
Department table

```
create table department_fat(deptno number(3), deptname varchar2(20), constraint dept_no_pk_fat primary key(deptno));
insert into department_fat values(1, 'SCOPE');
insert into department_fat values(2, 'SITE');
insert into department_fat values(3, 'SELECT');
insert into department_fat values(4, 'SENSE');
insert into department_fat values(5, 'SMEC');
insert into department_fat values(6, 'SCHEME');

select * from department_fat;
```



Student table

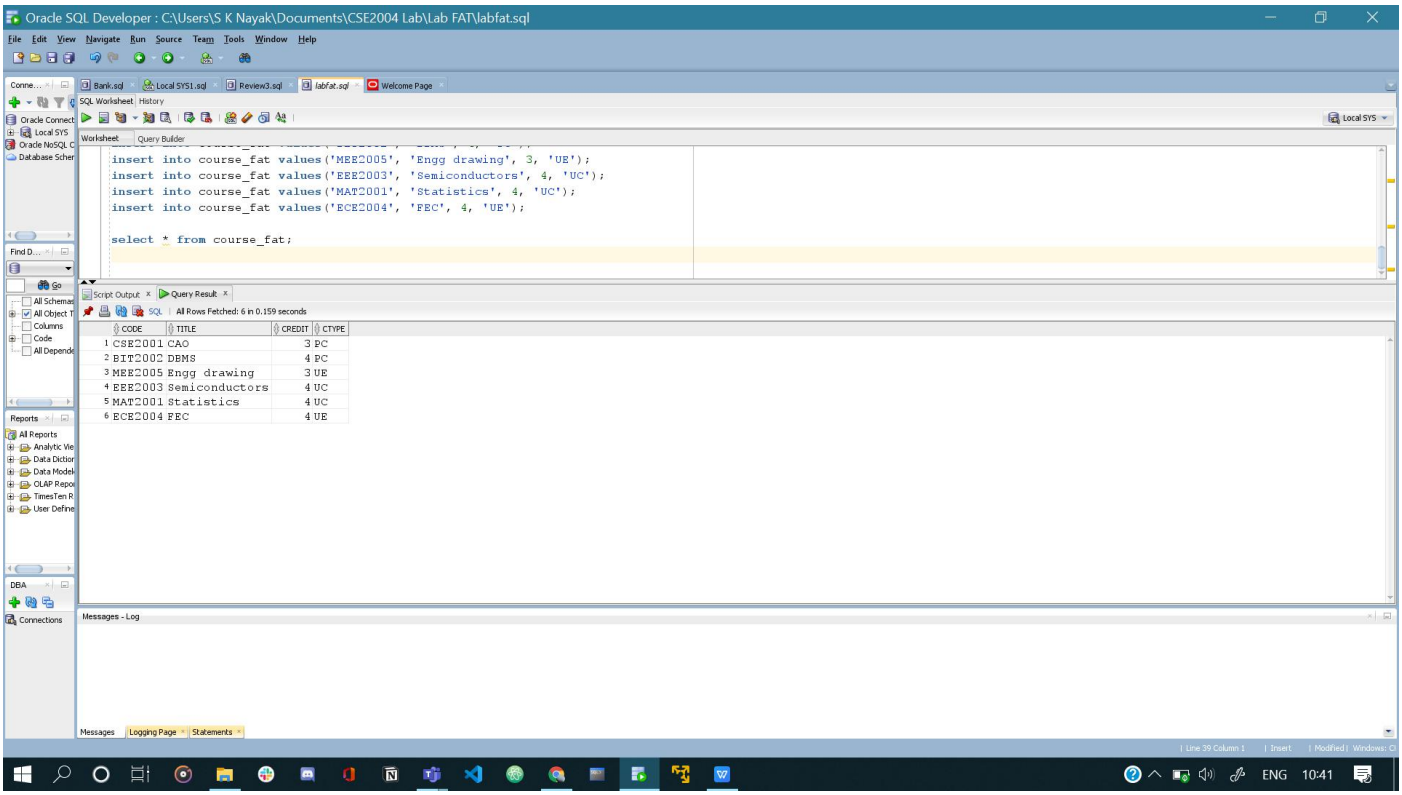
```
create table student_fat(regno varchar2(10), name varchar2(20), dob date, address
varchar2(20), cgpa number(10, 5), dno number(3),
constraint reg_no_pf_fat primary key(regno), constraint dno_fk_fat foreign key(dno)
references department_fat(deptno));

alter table student_fat add constraint cgpa_stu_chk check(cgpa between 2.5 and 10);

insert into student_fat values('19BCE0001', 'Ravi', '12-DEC-1999', 'Pune', 7.9, 2);
insert into student_fat values('19BME0002', 'Rajeev', '12-SEP-1999', 'Hyderabad', 8.9, 5);
insert into student_fat values('19BCE0003', 'Sanjeev', '11-OCT-2001', 'Vellore', 9.3, 1);
insert into student_fat values('19BIT0004', 'Swara', '10-NOV-2001', 'Chennai', 8.3, 2);
insert into student_fat values('19BEE0005', 'Aishani', '13-OCT-2001', 'Bangalore', 9.3, 3);
insert into student_fat values('19BCL0006', 'Raghul', NULL, 'Vellore', 9.3, 6);

select * from student_fat;
```



Course table

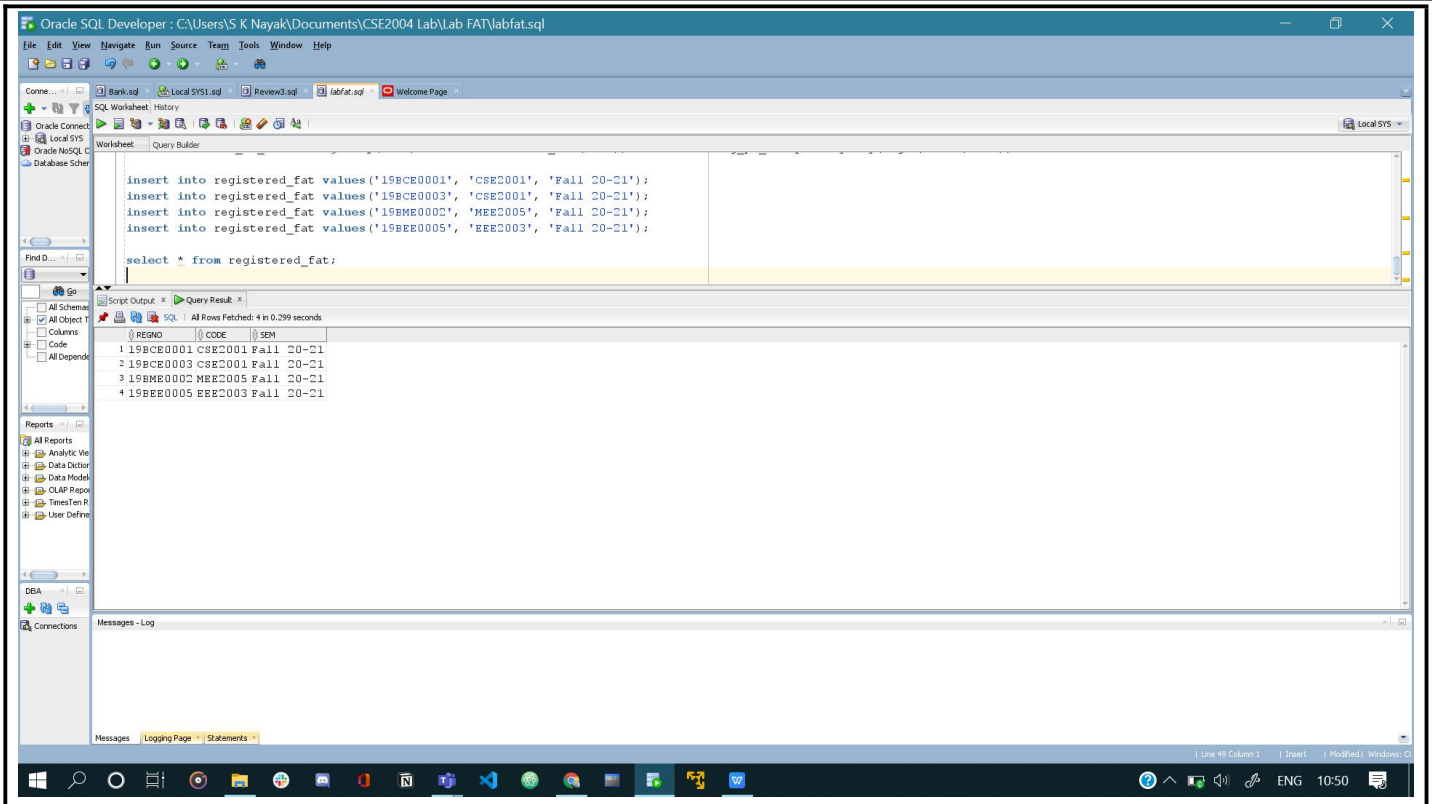
```
create table course_fat(code varchar2(10), title varchar2(30) unique, credit number(2),
ctype varchar2(10), constraint code_pk_fat primary key(code),
constraint ctype_chk_fat check(ctype in ('PC', 'UC', 'UE')));

insert into course_fat values('CSE2001', 'CAO', 3, 'PC');
insert into course_fat values('BIT2002', 'DBMS', 4, 'PC');
insert into course_fat values('MEE2005', 'Engg drawing', 3, 'UE');
insert into course_fat values('EEE2003', 'Semiconductors', 4, 'UC');
```

```
insert into course_fat values('MAT2001', 'Statistics', 4, 'UC');
insert into course_fat values('ECE2004', 'FEC', 4, 'UE');

select * from course_fat;
```

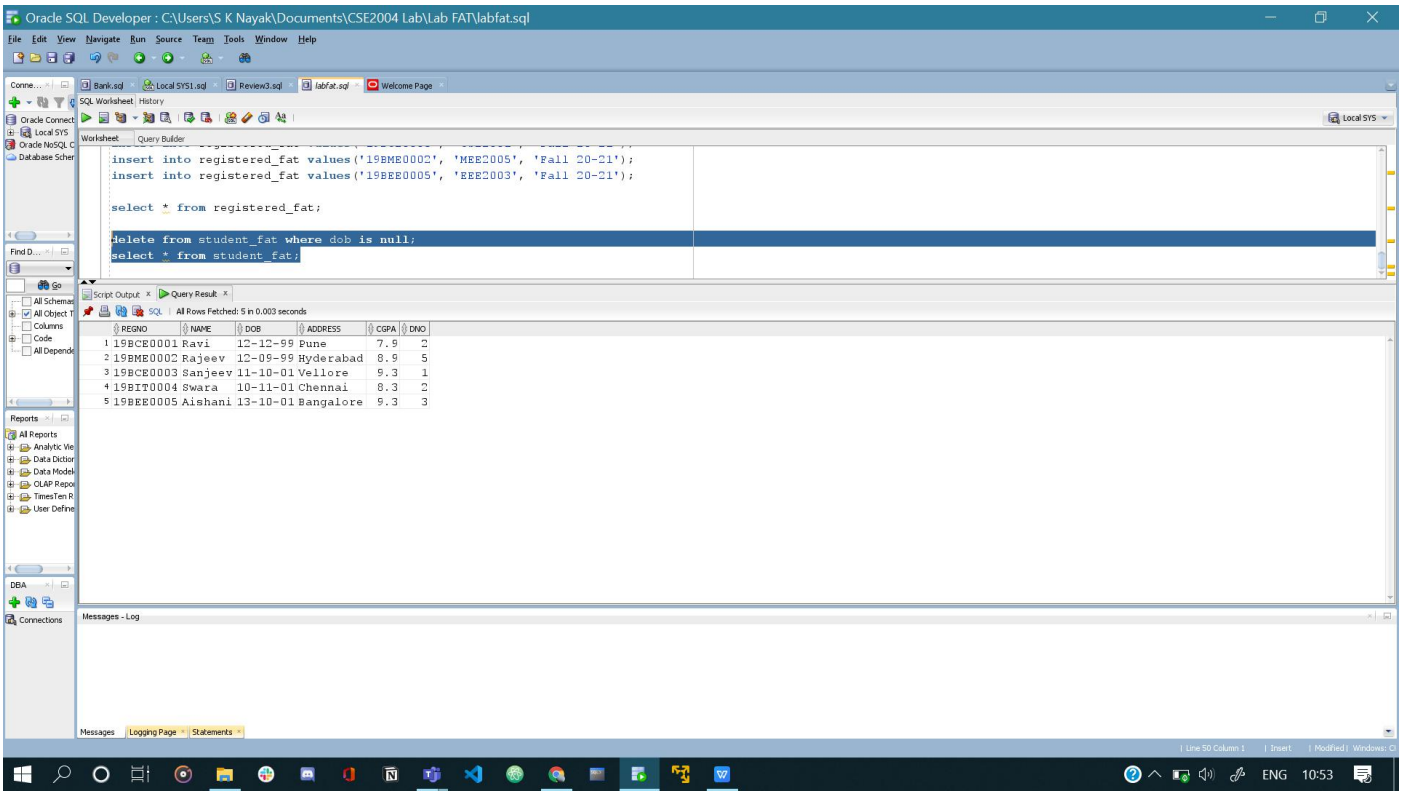


Registered table

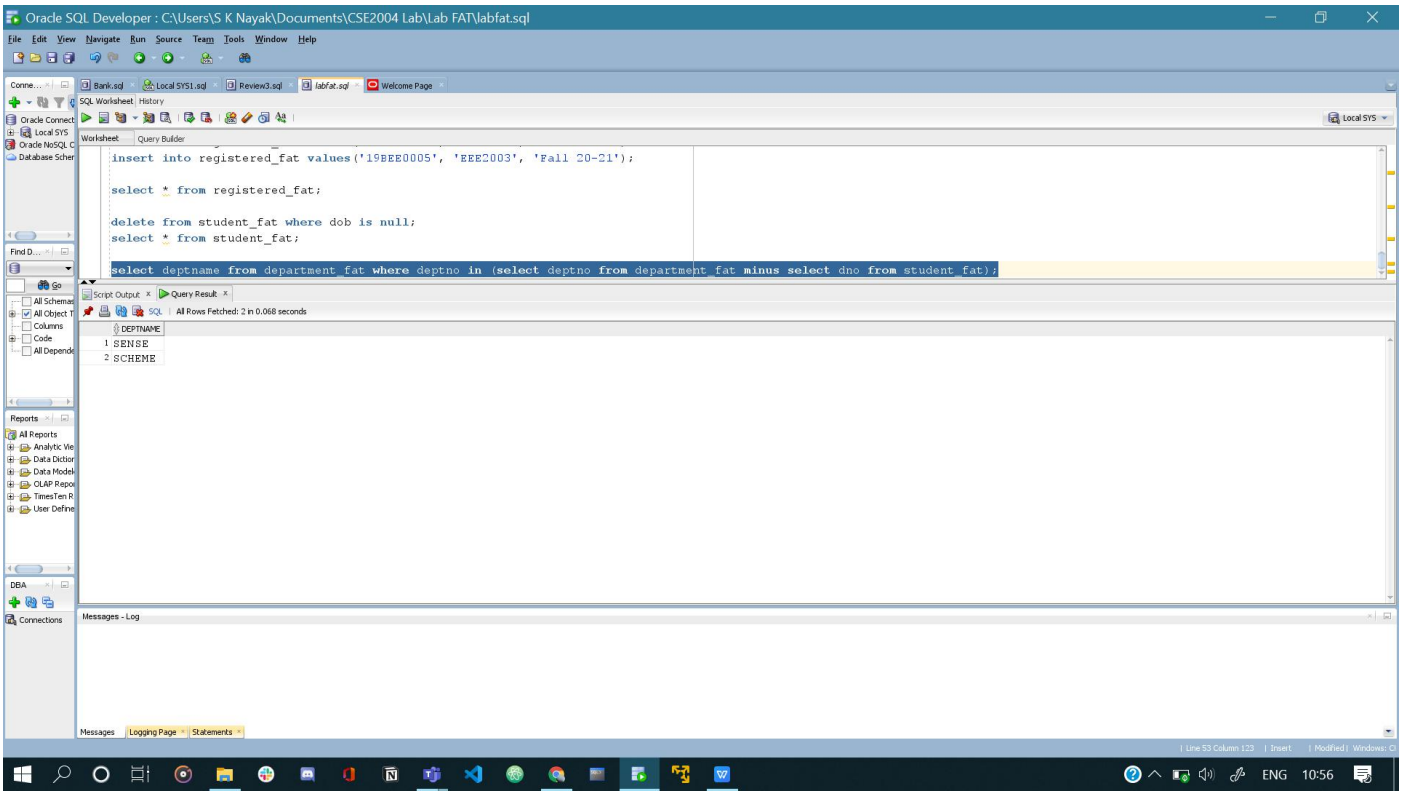
```
create table registered_fat(regno varchar2(10), code varchar2(10), sem varchar2(20),
constraint regno_fk_fat foreign key(regno) references student_fat(regno),
constraint code_fk_fat foreign key(code) references course_fat(code), constraint
reg_pk_fat primary key(regno, code, sem));

insert into registered_fat values('19BCE0001', 'CSE2001', 'Fall 20-21');
insert into registered_fat values('19BCE0003', 'CSE2001', 'Fall 20-21');
insert into registered_fat values('19BME0002', 'MEE2005', 'Fall 20-21');
insert into registered_fat values('19BEE0005', 'EEE2003', 'Fall 20-21');

select * from registered_fat;
```



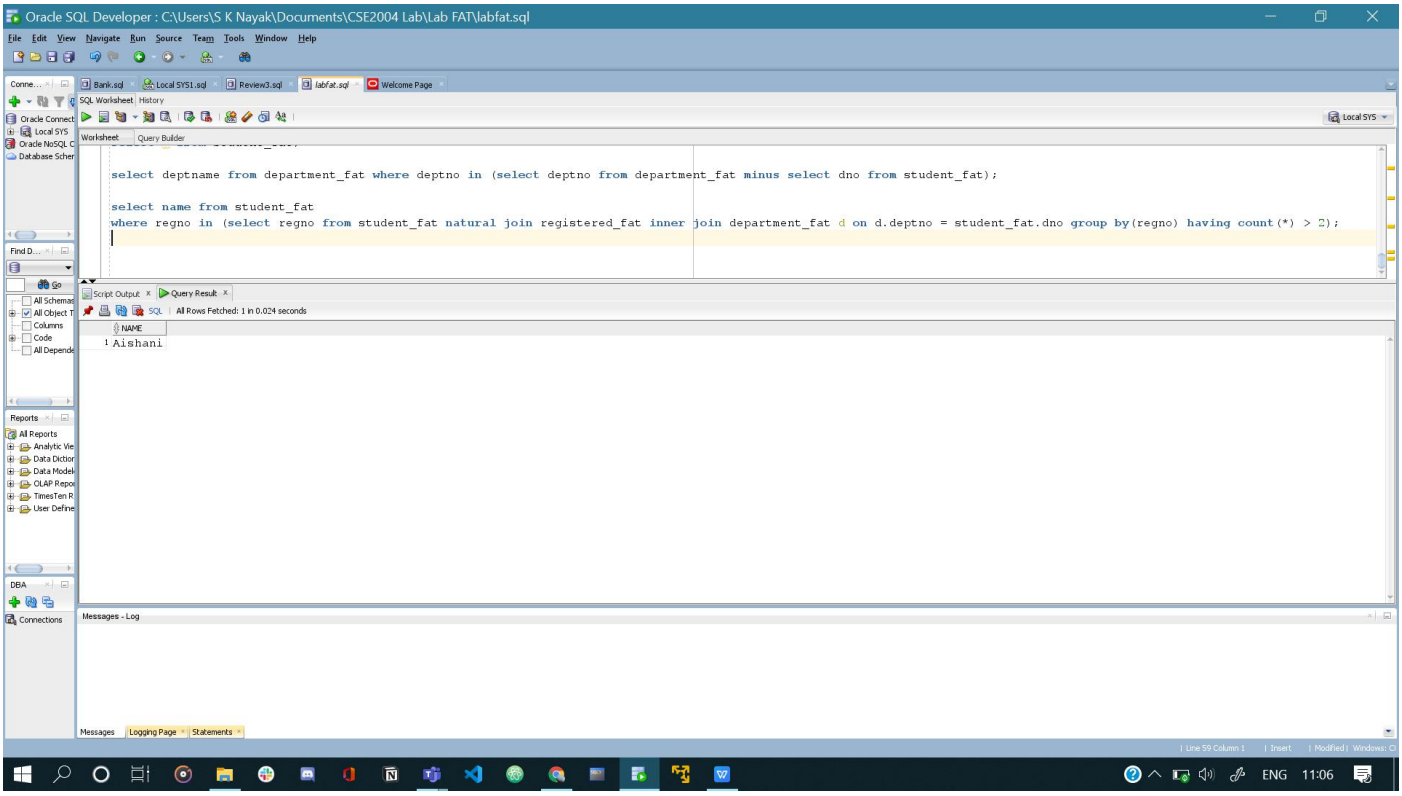
4. delete from student_fat where dob is null;
   select * from student_fat;



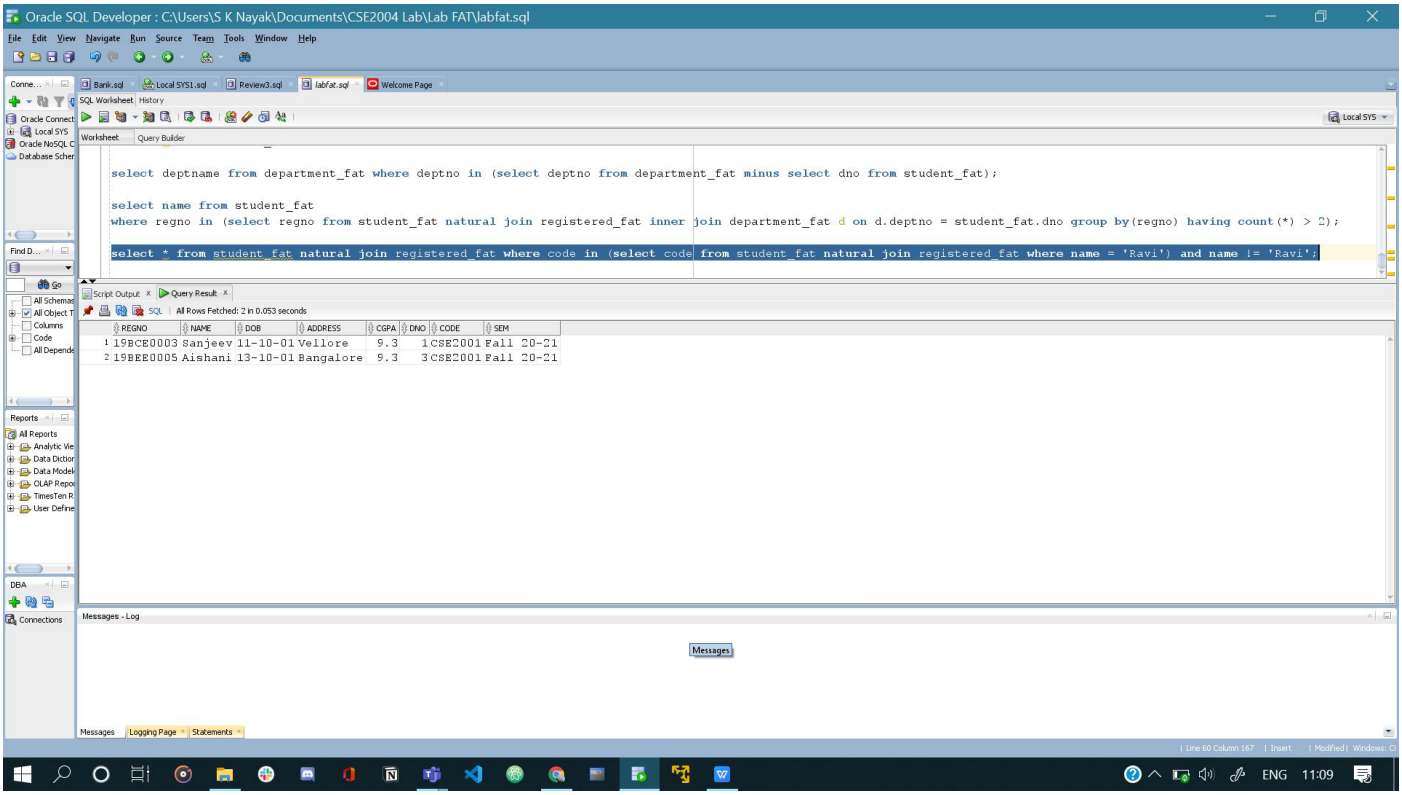
5. A) select deptname from department_fat where deptno in (select deptno from department_fat minus select dno from student_fat);



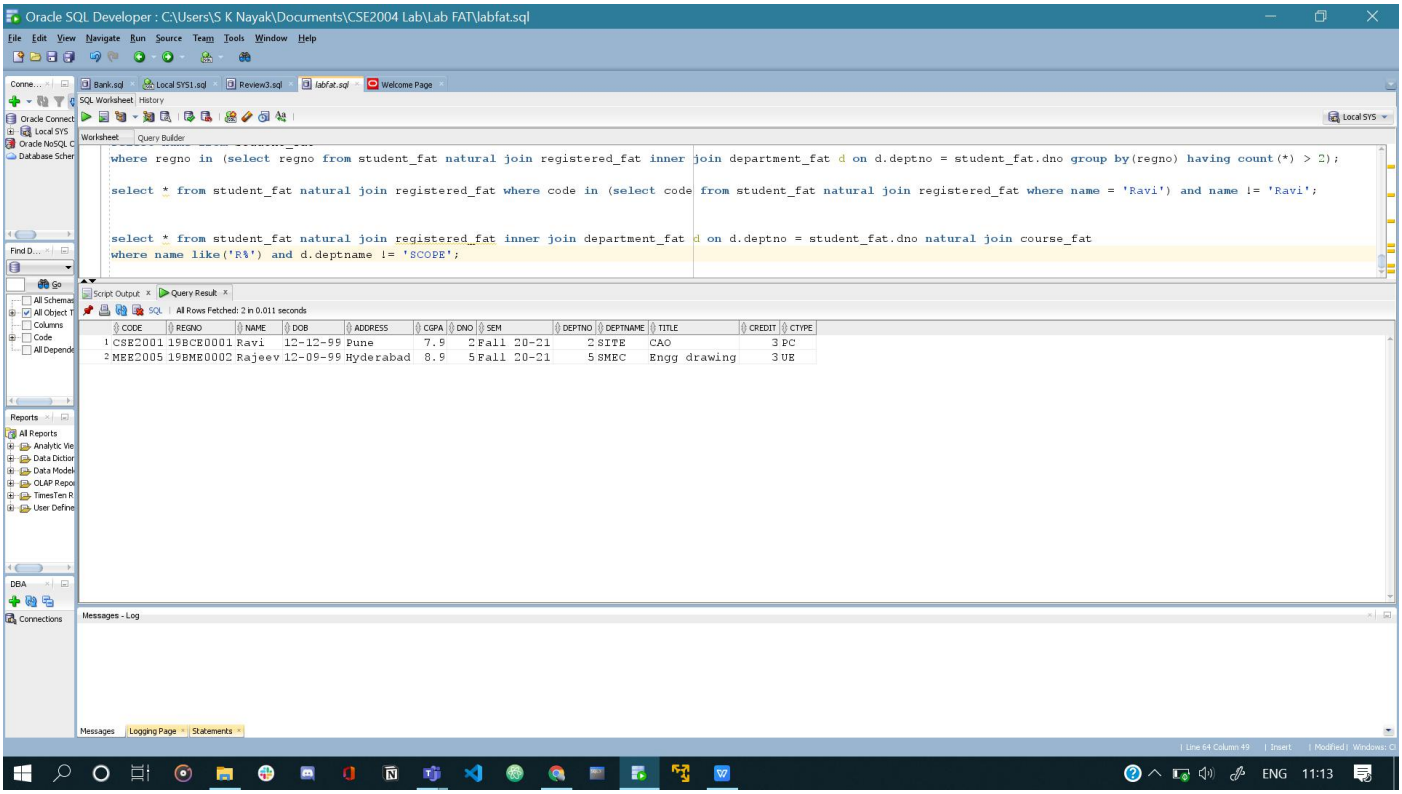
B) select name from student_fat where regno in (select regno from student_fat natural join registered_fat inner join department_fat d on d.deptno = student_fat.dno group by(regno) having count(*) > 2);



C) select * from student_fat natural join registered_fat where code in (select code from student_fat natural join registered_fat where name = 'Ravi') and name != 'Ravi';

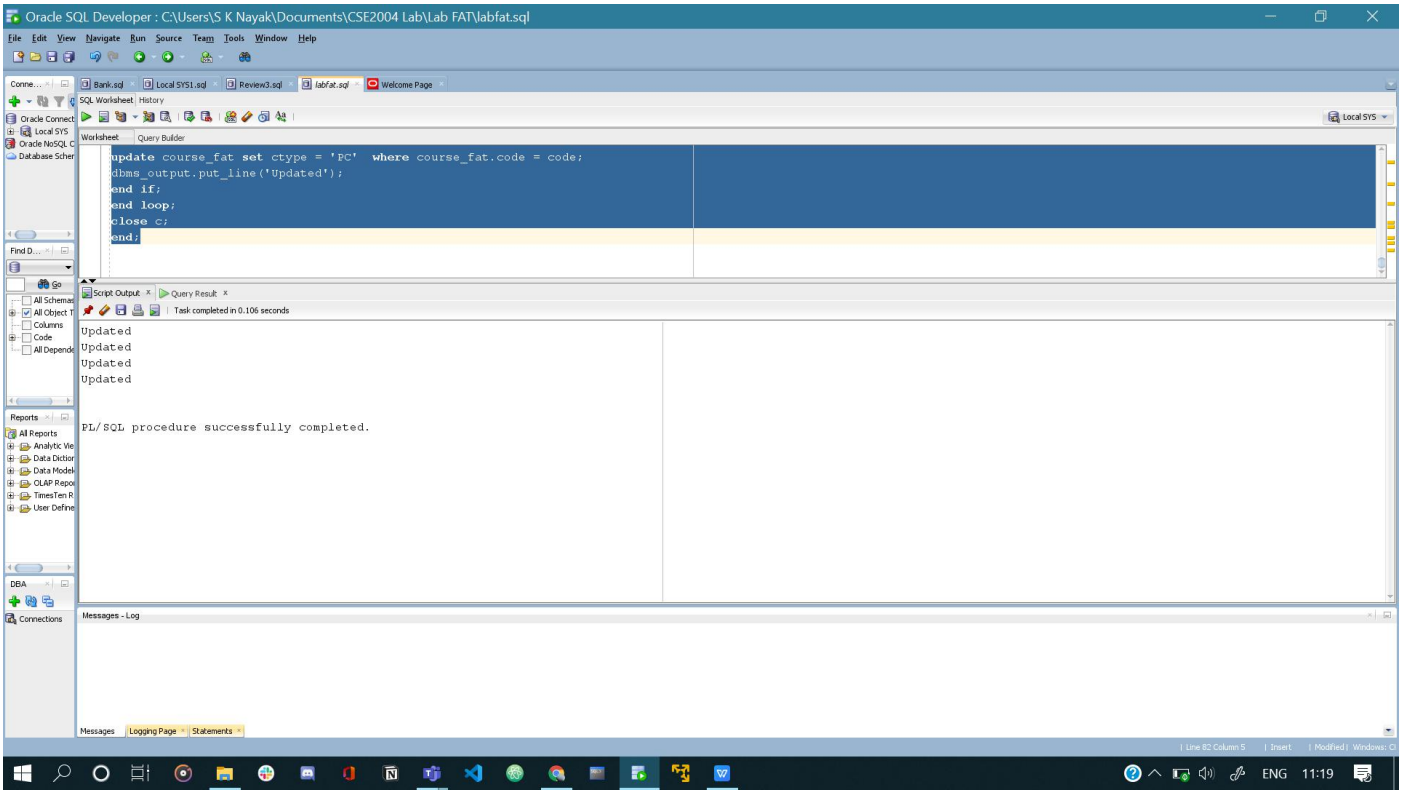


D) select * from student_fat natural join registered_fat inner join department_fat d on d.deptno = student_fat.dno natural join course_fat where name like('R%') and d.deptname != 'SCOPE';



6. Code:

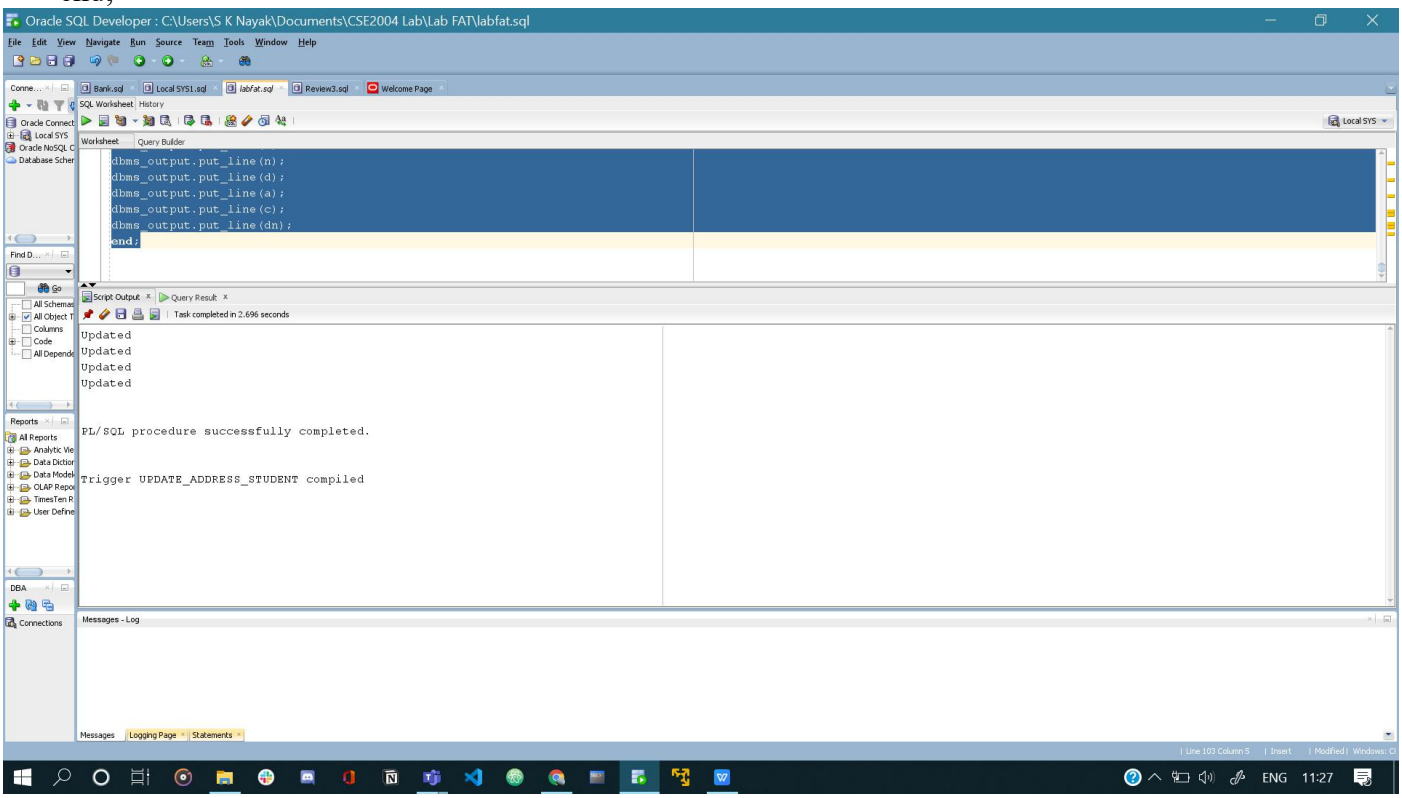
```
declare
cursor c is select code, credit from course_fat;
code course_fat.code%type;
cred course_fat.credit%type;
begin
open c;
loop
fetch c into code, cred;
exit when c%notfound;
if cred = 4 then
update course_fat set ctype = 'PC'  where course_fat.code = code;
dbms_output.put_line('Updated');
end if;
end loop;
close c;
end;
```



7. Code:

```
create or replace trigger update_address_student
after update on student_fat
for each row
enable
declare
r student_fat.regno%type := :new.regno;
n student_fat.name%type := :new.name;
d student_fat.dob%type := :new.dob;
a student_fat.address%type := :new.address;
```

```
c student_fat.cgpa%type := :new.cgpa;
dn student_fat.dno%type := :new.dno;
begin
dbms_output.put_line(r);
dbms_output.put_line(n);
dbms_output.put_line(d);
dbms_output.put_line(a);
dbms_output.put_line(c);
dbms_output.put_line(dn);
end;
```



3.alter table student_fat drop column address;