EXP 6: COMPLEX SQL QUERIES-I

B1. Consider the following relational schema:

PERSON (SS#, NAME, ADDRESS) CAR (REGISTRATION_NUMBER, YEAR, MODEL) ACCIDENT (DATE, DRIVER, CAR_REG_NO)

OWNS (SS#, LICENSE)

(i) Find the names of persons who are involved in an accident.

(ii) Find the registration number of cars which were not involved in any accident.

create table person(ss# int not null,name varchar(15),address varchar(20), CONSTRAINT ss PRIMARY KEY (ss#));

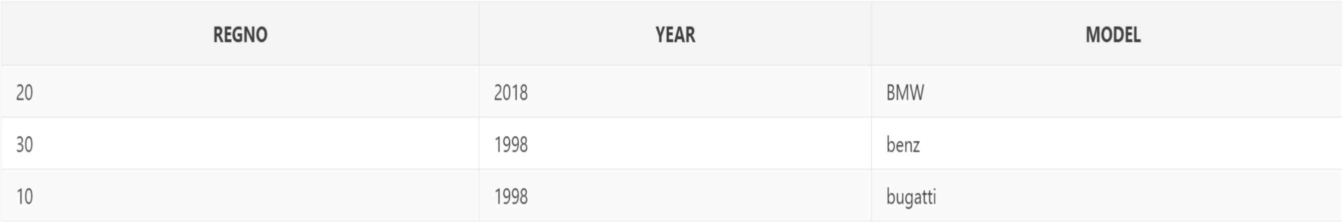
create table car(regno int primary key,year int,model varchar(20));

create table accident(license int primary key,regno int);

alter table accident add (constraint ac foreign key(regno) references car(regno));

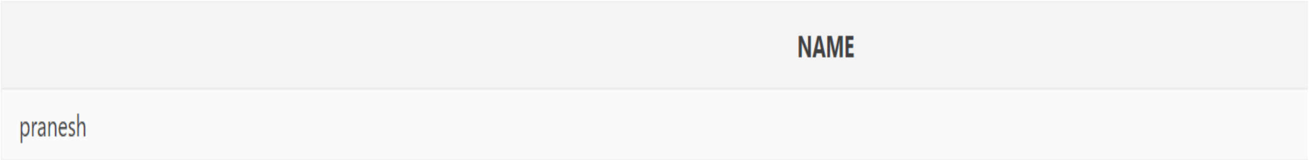
alter table owns add(constraint ac2 foreign key(license) references accident(license),constraint ac3 foreign key(ss#) references person(ss#));

i)select * from car;



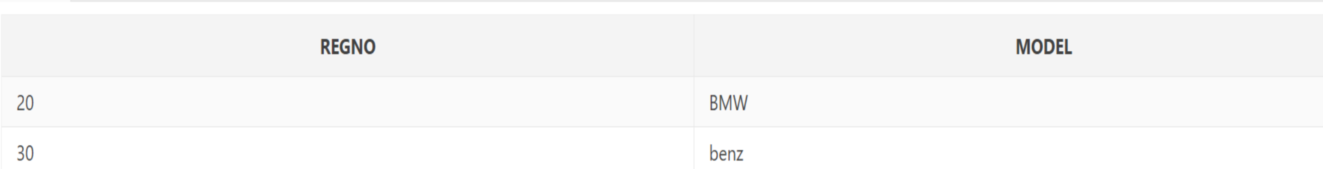
REGNO	YEAR	MODEL
20	2018	BMW
30	1998	benz
10	1998	bugatti

select name from person where ss# in (select owns.ss# from accident inner join owns on accident.license=owns.license)



NAME
pranesh

ii) select regno,model from car where regno not in (select regno from accident);



REGNO	MODEL
20	BMW
30	benz

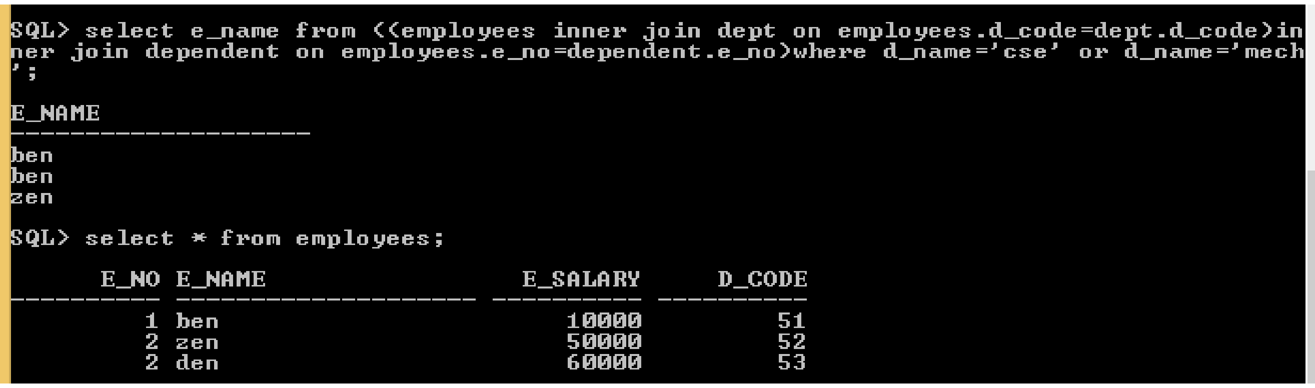
EXP 6: COMPLEX SQL QUERIES-I

B2)employee (e_no, e_name, e_salary, d_code), dept (d_code, d_name) dependent (depndt_name, e_no, relation)

student (std_id, std_name, date_of_birth, phone, dept_name). (i) Show the names of employees in purchase and accounts departments with at least one dependent.

(ii) A constraint named less_than_20 was defined on the field date_of_birth of table student. Delete this constraint.

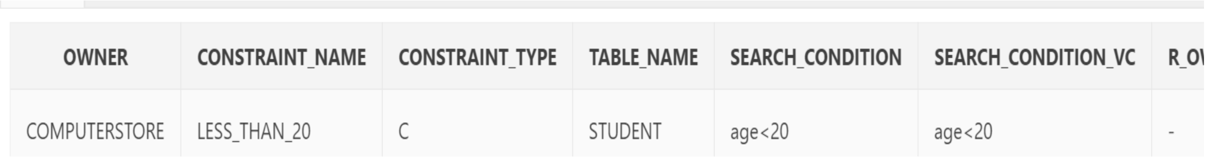
(iii) Consider the table student and list names of students in the departments other than maths and computer.



```
SQL> select e_name from ((employees inner join dept on employees.d_code=dept.d_code)in
ner join dependent on employees.e_no=dependent.e_no)where d_name='cse' or d_name='mech
';

E_NAME
--------------------
ben
ben
zen

SQL> select * from employees;

      E_NO E_NAME                 E_SALARY     D_CODE
---------- -------------------- ---------- ----------
         1 ben                       10000         51
         2 zen                       50000         52
         2 den                       60000         53
```

ii) alter table student add (constraint less_than_20 check(age<20));

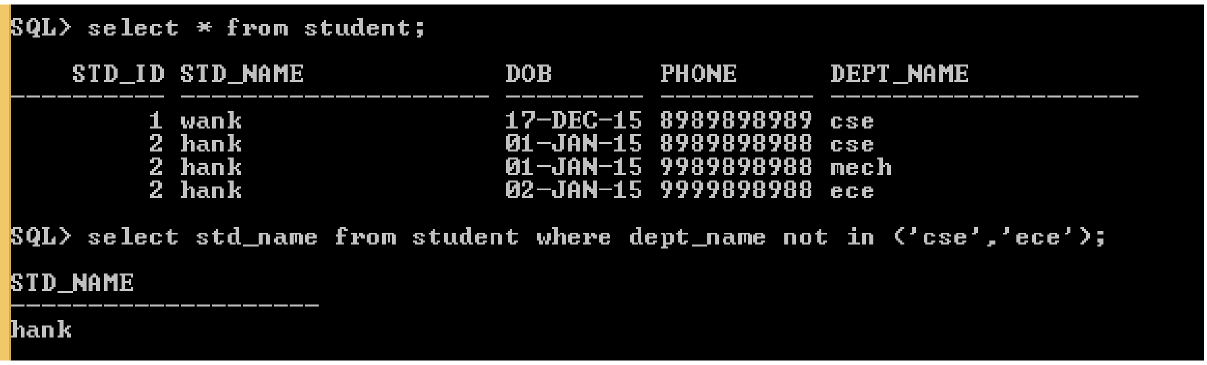
SELECT * FROM user_constraints WHERE table_name ='STUDENT';



OWNER	CONSTRAINT_NAME	CONSTRAINT_TYPE	TABLE_NAME	SEARCH_CONDITION	SEARCH_CONDITION_VC	R_OV
COMPUTERSTORE	LESS_THAN_20	C	STUDENT	age<20	age<20	-

alter table student drop constraint less_than_20;

iii)



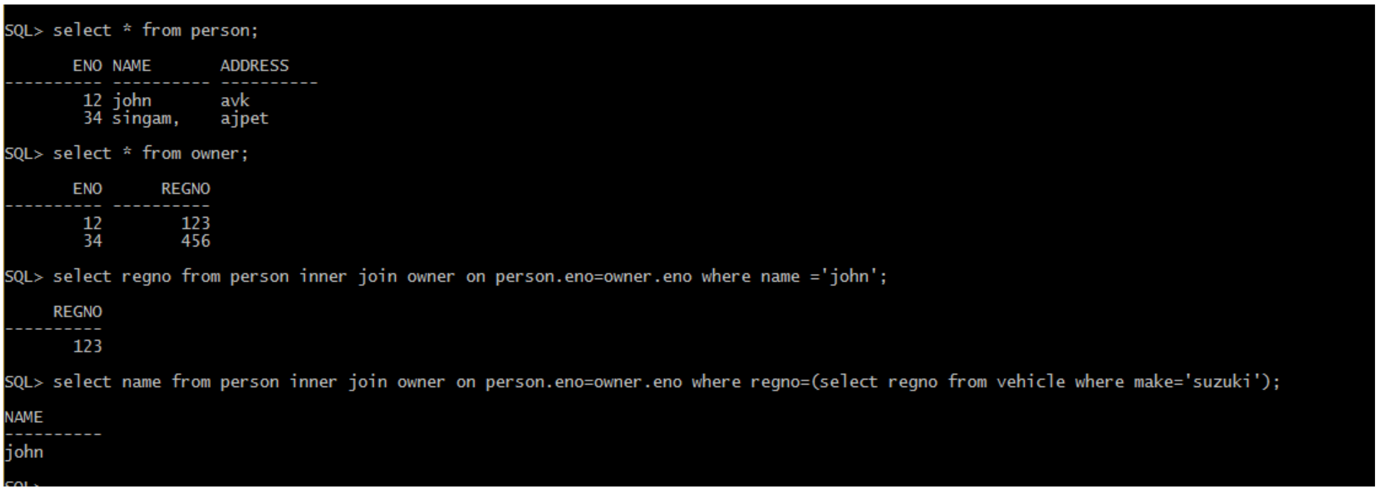
```
SQL> select * from student;

    STD_ID STD_NAME             DOB       PHONE      DEPT_NAME
---------- -------------------- --------- ---------- --------------------
         1 wank                 17-DEC-15 8989898989 cse
         2 hank                 01-JAN-15 8989898988 cse
         2 hank                 01-JAN-15 9989898988 mech
         2 hank                 02-JAN-15 9999898988 ece

SQL> select std_name from student where dept_name not in ('cse','ece');

STD_NAME
--------------------
hank
```

EXP 6: COMPLEX SQL QUERIES-I

B3) Given the following relations Vehicle (Reg_no, make, colour) Person(eno, name, address)

Owner(eno, reg_no) Write expressions in tuple calculus to answer the following queries:(i) List the reg_no of vehicles owned by John.

(ii) List the names of persons who own maruti cars.



```
SQL> select * from person;

       ENO NAME       ADDRESS
---------- ---------- ----------
        12 john       avk
        34 singam,    ajpet

SQL> select * from owner;

       ENO      REGNO
---------- ----------
        12        123
        34        456

SQL> select regno from person inner join owner on person.eno=owner.eno where name ='john';

     REGNO
----------
       123

SQL> select name from person inner join owner on person.eno=owner.eno where regno=(select regno from vehicle where make='suzuki');

NAME
----------
john

SQL>
```