*-- Q1: Consider the following relational schema for the*

*-- Office of the Controller of Examinations Application.*

*-- Student (Rollno, Name, Dob, Gender, Doa, Bcode);*

*-- Implement a check constraint for*

*-- Gender*

*-- Date of Admission*

*-- Branch (Bcode, Bname, Dno);*

*-- Department (Dno, Dname);*

*-- Course (Ccode, Cname, Credits, Dno);*

*-- Branch\_Course (Bcode, Ccode, Semester);*

*-- Enrolls (Rollno, Ccode, Sess, Grade);*

*-- For Example,*

*-- SESS can take values ‘APRIL 2013’, ‘NOV 2013’*

*-- Implement a check constraint for grade Value Set (‘S’, ‘A’, ‘B’, ‘C’, ‘D’, ‘E’, ‘U’*

*-- );*

*-- Students are admitted to Branches and they are offered by Departments. A*

*-- branch is offered by only one department.*

*-- Each branch has a set of Courses (Subjects). Each student must enroll during*

*-- a semester. Courses are offered by Departments. A course is offered only by*

*-- one department. If a student is unsuccessful in a course he/she must enroll for*

*-- the course during next session. A student has successfully completed a*

*-- course if the grade obtained by is from the list (A, B, C, D, and E).*

*-- A student is unsuccessful if he/she have grade ‘U’ in a course.*

*-- Primary Keys are underlined.*

create table department(

dno INT NOT NULL,

dname VARCHAR(20) NOT NULL,

PRIMARY KEY(dno)

);

create table branch(

bcode VARCHAR(20) NOT NULL PRIMARY KEY,

bname VARCHAR(20) NOT NULL,

dno INT,

foreign key (dno) references department(dno)

);

create table student (

rollno int not null,

name varchar(20) not null,

dob date not null,

gender varchar(20) not null,

doa date not null,

bcode VARCHAR(20) NOT NULL,

check(gender in ('MALE', 'FEMALE')),

check(doa > '2000-07-07'),

primary key(rollno),

foreign key(bcode) references branch(bcode)

);

create table course(

ccode VARCHAR(20) NOT NULL PRIMARY KEY,

cname VARCHAR(20) NOT NULL,

credits INT NOT NULL,

dno INT NOT NULL,

foreign key (dno) references department(dno)

);

create table branch\_course(

bcode VARCHAR(20) NOT NULL,

ccode VARCHAR(20) NOT NULL,

semester INT NOT NULL,

PRIMARY KEY(bcode, ccode),

foreign key (bcode) references branch(bcode),

foreign key (ccode) references course(ccode)

);

create table enrolls(

rollno INT NOT NULL,

ccode VARCHAR(20) NOT NULL,

sess VARCHAR(20) NOT NULL,

grade VARCHAR(20) NOT NULL,

PRIMARY KEY(rollno, ccode, sess),

check(grade in ('S', 'A', 'B', 'C', 'D', 'E', 'U')),

check(sess in ('APRIL 2013', 'NOV 2013')),

foreign key (rollno) references student(rollno),

foreign key (ccode) references course(ccode)

);

insert into department

values (1, 'ENGINEERING'),

(2, 'HUMANITIES'),

(3, 'COMMERCE');

insert into branch

values ('B01', 'IT', 1),

('B02', 'ACCOUNTS', 3),

('B03', 'HISTORY', 2);

insert into student

values (

1,

'Ramesh',

'1987-02-12',

'MALE',

'2001-10-10',

'B01'

),

(

2,

'Suresh',

'1989-03-03',

'MALE',

'2002-10-12',

'B01'

),

(

3,

'Chameli',

'1984-03-07',

'FEMALE',

'2004-09-10',

'B02'

),

(

4,

'Mahesh',

'1992-12-12',

'MALE',

'2003-12-12',

'B02'

),

(

5,

'Jessica',

'1990-01-05',

'FEMALE',

'2004-10-10',

'B03'

);

insert into course

values ('C01', 'MATHEMATICS', 3, 1),

('C02', 'PHYSICS', 3, 2),

('C03', 'CHEMISTRY', 3, 2),

('C04', 'ACCOUNTENCY', 3, 3),

('C05', 'COMPUTER SCIENCE', 3, 1),

('C06', 'COMPUTER ARCHI', 3, 1);

insert into branch\_course

values ('B01', 'C01', 1),

('B01', 'C02', 2),

('B01', 'C03', 3),

('B02', 'C04', 1),

('B02', 'C05', 2),

('B02', 'C06', 3);

insert into enrolls

values (1, 'C01', 'APRIL 2013', 'A'),

(1, 'C02', 'APRIL 2013', 'B'),

(1, 'C03', 'APRIL 2013', 'C'),

(2, 'C01', 'APRIL 2013', 'A'),

(2, 'C02', 'APRIL 2013', 'B'),

(2, 'C03', 'APRIL 2013', 'C'),

(3, 'C01', 'APRIL 2013', 'A'),

(3, 'C02', 'APRIL 2013', 'B'),

(3, 'C03', 'NOV 2013', 'C'),

(4, 'C01', 'NOV 2013', 'A'),

(4, 'C02', 'NOV 2013', 'B'),

(4, 'C03', 'NOV 2013', 'C'),

(5, 'C01', 'APRIL 2013', 'A'),

(5, 'C02', 'APRIL 2013', 'B'),

(5, 'C03', 'APRIL 2013', 'C');

*-- select \**

*-- from enrolls;*

*-- select \**

*-- from student;*

*-- select \**

*-- from branch;*

*-- select \**

*-- from branch\_course;*

*-- select \**

*-- from department;*

*-- A. Develop a SQL query to list details of Departments that offer more than 3*

*-- branches.*

select \*

from department

where department.dno in (

select branch.dno

from branch

group by branch.dno

having **count**(branch.dno) > 3

);

*-- B. Develop a SQL query to list the details of Departments that offer more than 6*

*-- courses.*

select \*

from department

where department.dno in (

select branch\_course.dno

from branch\_course

group by branch\_course.dno

having **count**(branch\_course.dno) > 6

);

*-- C. Develop a SQL query to list the details of courses that are common for more*

*-- than 3 branches.*

select \*

from course

where course.ccode in (

select branch\_course.ccode

from branch\_course

group by branch\_course.ccode

having **count**(branch\_course.ccode) > 3

);

*-- D. Develop a SQL query to list students who got ‘S’ in more than 2 courses*

*-- during single enrollment.*

select \*

from student

where student.rollno in (

select enrolls.rollno

from enrolls

group by enrolls.rollno

having **count**(enrolls.rollno) > 2

and enrolls.grade = 'S'

);