Sample Schema;
Student schema (crn, name, address, phone, DOB)
Course schema (courseid, cname, duration, fee)
Enroll schema (enrollid, crn, coursed, enrolldt, completedt)

Write the SQL statements for the following:

1) Create a table student on student schema with crn as primary key and a table course on course schema with course id as primary key.

Ans:
```
create table student(
        crn varchar(20) primary key,
        name varchar(20),
        address varchar(20),
        phone varchar(20),
        dob date);
create table course(
        courseid varchar(20) primary key,
        cname varchar(20),
        duration float,
        fee int);
```

2) Insert the following tuples into relation student

crn	**Name**	**address**	**phone**	**dob**
066/bct/112	Anil	Lalitpur	Null	12-jan-1992
066/bct/115	Bibek	Bhaktapur	45356678	01-apr-1990
066/bct/166	Rohan	New Road	41244533	14-mar-1991
066/bct/188	Puja	Baneshwor	42322432	12-jul-1989
066/bct/154	John	Jorpati	51423234	18-dec-1993
066/bct/176	Sita	Lagankhel	51222765	10-jun-1989

Ans:
```
insert into student values('066/bct/112', 'Anil', 'Lalitpur', null, '1992-01-12');
insert into student values('066/bct/115', 'Bibek', 'Bhaktapur', 45356678, '1990-04-01');
...
```

3) Insert the following tuples into the relation course:

courseid	**coursename**	**duration**	**Fee**
C101	Java	5	12000
C102	Oracle	8	21000
C103	Linux	3	15000
C104	Cisco	2	22000

Ans:
```
insert into course values ('C101', 'Java', 5, 12000);
insert into course values ('C102', 'Oracle', 8, 21000);
insert into course values ('C103', 'Linux', 3, 15000);
insert into course values ('C104', 'Cisco', 2, 22000);
```

4) Create a relation enroll with enroll id as primary key. Attributes crn and course id are foreign keys referencing relation student and course respectively.

Ans:
```
create table enroll(
    enrollid varchar(10) primary key,
    crn varchar(20),
    courseid varchar(20),
    enrolldt date,
    completedt date,
    foreign key (crn) references student(crn),
    foreign key (courseid) references course(courseid));
```

5) Insert the following tuples into the relation enroll:

enrollid	**Crn**	**courseid**	**enrolldt**	**Completed**
E12	066/bct/112	C101	23-jan-2011	
E13	066/bct/115	C101	23-jan-2011	1-jun-2011
E14	066/bct/176	C103	01-jun-2012	
E15	066/bct/176	C102	01-jun-2012	
E18	066/bct/115	C104	01-mar-2012	
E20	066/bct/166	C101	01-apr-2012	
E30	066/bct/188	C101	23-apr-2012	

Ans:
```
insert into enroll (enrollid, crn, courseid, enrolldt) values('E12', '066/bct/112', 'C101', '2011-01-23');
  insert into enroll values('E13', '066/bct/115', 'C101', '2011-01-23', '2011-06-01' );
insert into enroll (enrollid, crn, courseid, enrolldt) values('E14', '066/bct/176', 'C103', '2012-06-01');
insert into enroll (enrollid, crn, courseid, enrolldt) values('E15', '066/bct/176', 'C102', '2012-06-01');
insert into enroll (enrollid, crn, courseid, enrolldt) values('E18', '066/bct/115', 'C104', '2012-03-01');
insert into enroll (enrollid, crn, courseid, enrolldt) values('E20', '066/bct/166', 'C101', '2012-04-01');
insert into enroll (enrollid, crn, courseid, enrolldt) values('E30', '066/bct/188', 'C101', '2012-04-23');
```

6) Insert a tuple into the relation enroll with enroll id e23, crn 066/bct/112 and course id C102. Comment.

Ans: insert into enroll (enrollid, crn, courseid) values('E23', '066/bct/112', 'C102');
Comment: all the values in tuple are valid. It doesn't violates any constraints.

7) Find the Cartesian product of student and enroll.

Ans: Implicitly=> select * from student, enroll;
OR Explicitly=> select * from student cross join enroll;

8) Find the natural join of student and enroll.

Ans: select * from student natural join enroll;
OR select * from enroll natural join student;
OR select * from student, enroll where student.crn = enroll.crn;
OR select * from student inner join enroll on student.crn = enroll.crn;

9) Find crn, names and enroll date of all students who have taken the course 'java'.

Ans: select crn, name, enrolldt from student natural join enroll natural join course where cname = 'Java';

10) Find the name and address of all the students who are enrolled on 01-jun-2012.
Ans: select distinct name, address from student natural join enroll where enrolldt = '2012-06-01';

11) Find the names and address of all the students who have taken both course java and linux.
Ans: select distinct name, address from student natural join enroll natural join course where cname in ('Java', 'Linux');

12) Delete the record of student having crn 066/bct/115. Comment.
Ans: delete from student where crn = '066/bct/115';
Comment: This query cannot be performed because crn is used as a primary key which is referenced to other table (enroll table).

13) Delete the record of the student having enroll id E12. Comment.
Ans:

```
create table temp (select * from (select crn from enroll where enrollid = 'E12') as temp);

delete from enroll where crn = (select * from (select crn from enroll where enrollid = 'E12') as
 temp);

delete from student where crn = (select crn from temp);
```

14) Delete the record of student who are enrolled in a course less than 15000.
Ans:

```
create table temp select * from (select crn from enroll where courseid in(select courseid from
course where fee < 15000)) as t;

 delete from enroll where crn in(select * from temp);

delete from student where crn in(select * from temp);
```

15) Create view with crn, name and address of student.
Ans: create view view as select crn, name, address from student;

16) Create a view ‘student_course’ having the attributes crn, name, phone, cname, enrolldt.
Ans: create view student_course as select distinct student.crn, name, phone, cname, enrolldt from student, course, enroll;

17) Create view with the crn and names of students who have taken course linux.
Ans:

```
create view temp as
select crn, name from student_backup where crn in (select distinct crn from
(select * from course_backup natural join enroll_backup) as tempo where cname = ‘Linux’);
```

18) Drop the view ‘student_course.
Ans: drop view student_course;

19) Drop the table student. Comment.
Ans: drop table student;