1. **Create database myrollno.**
2. **use myrollno;**
3. Create the following table with the given set of attributes.
   1. Course (course\_id varchar(5), course\_name varchar(15),credits int) **constraints**: course\_id primary key.

ans: **create table course(course\_id varchar(5),**

**course\_name varchar(15),**

**credits int,**

**primary key(course\_id));**

* 1. Student (s\_id int,s\_name varchar(25),branch varchar(10),course\_id varchar(5))

**constraints:** s\_id primary & course\_id foreign key refers Course table.

Ans: **create table student (sid int primary key,**

**s\_name varchar(25),**

**branch varchar(10),**

**course\_id varchar(5),**

**constraint fk\_course\_cid foreign key(course\_id) references course(course\_id));**

* 1. Describe Course and student tables.

**desc course; desc student;**

* 1. Add a column dob column of type date after s\_name in student table.

**alter table student add dob date after s\_name;**

* 1. Add NOT NULL constraint to credits in Course table .

**alter table course modify credits int not null;**

* 1. Decrease the width of s\_name attribute of Student table by 5.

**alter table student modify s\_name varchar(20);**

* 1. Insert the following rows into each table.

**insert into course values('DB101','DBMS',3);**

**insert into course values('CO102',CO',3);**

**insert into course values('M3,'Maths3',2);**

**insert into student values (121,'Rajesh','1999-03-12','cse','DB101');**

**select \* from course;**

**select \* from student;**

**course table**

| **Course\_id** | **Course\_name** | **credits** |
| --- | --- | --- |
| DB101 | DBMS | 3 |
| CO102 | CO | 3 |
| M3 | Maths3 | 2 |

**Student table**

| **s\_id** | **s\_name** | **dob** | **branch** | **course\_id** |
| --- | --- | --- | --- | --- |
| 121 | Rajesh | 1999-12-09 | cse | DB101 |
| 122 | Gautam Kumar | 2000-03-04 | ece | CO102 |
| 123 | Allen | 1998-04-12 | cse | DB101 |
| 124 | Vivekananda | 2001-03-14 | ece | M3 |
| 125 | Daneil | 1997-03-18 | it | DB101 |

**select \* from course;**

**select \* from student;**

* 1. List all students from ‘cse’ branch .

**select \* from student where branch='cse';**

* 1. Display all students whose name starts with character A.

**select \* from student where s\_name like 'A%';**

* 1. Display all students whose name has exactly 5 characters.

**select \* from student where s\_name like '\_\_\_\_\_';**

* 1. Count the number of students from each dept.

**select count(\*) from student group by branch;**

* 1. Rename course\_name attribute to c\_name.

**alter table course change column course\_name c\_name varchar(15);**

* 1. Drop column dob from student table.

**alter table student drop column dob;**

* 1. Add one credit to DBMS course.

**update course set credits=credits+1 where course=’DBMS’;**

* 1. List sid,course\_id,course\_name and credits.
  2. Delete all the students from cse branch.
  3. drop foreign key constraint from student table.

ANSWERS

1. create table course(course\_id varchar(5),

course\_name varchar(15),

credits int,

primary key(course\_id));

1. create table student (sid int primary key,

s\_name varchar(25),

branch varchar(10),

course\_id varchar(5),

constraint fk\_course\_cid foreign key(course\_id) references course(course\_id));

1. desc course; desc student;
2. **alter table student add dob date after s\_name;**
3. **alter table course modify credits int not null;**
4. **alter table student modify s\_name varchar(20);**
5. **insert into course values('DB101','DBMS',3);**

**insert into student values (121,'Rajesh','1999-03-12','cse','DB101');**

1. **select \* from student where branch='cse';**
2. **select \* from student where s\_name like 'A%';**
3. **select \* from student where s\_name like '\_\_\_\_\_';**
4. **select count(\*) from student group by branch;**
5. **update course set credits=credits+1 where course=’DBMS’;**
6. **alter table course change column course\_name c\_name varchar(15);**
7. **alter table student drop column dob;**
8. delete from student;

ALTER TABLE `User` ADD CONSTRAINT `user\_properties\_foreign` FOREIGN KEY (`properties`) REFERENCES `Properties` (`ID`) ON DELETE NO ACTION ON UPDATE NO ACTION;