## Part A - Question 3:

STUDENT (rollno, name, class, birthdate)

COURSE (courseno, coursename, max\_marks, pass marks)

SC (rollno, courseno, marks)

1. Create the above three tables along with key constraints.
2. Write an Insert script for insertion of rows with substitution variables and insertappropriate data.
3. Add a constraint that the marks entered should strictly be between 0 and 100.
4. While creating SC table, composite key constraint was forgotten. Add the compositekeynow.
5. Display details of student who takes 'Database Management System' course.
6. Display the names of students who have scored more than 70% in Computer Networksand have not failed in any subject.
7. Display the average marks obtained by each student.
8. Select all courses where passing marks are more than 30% of average maximum mark.
9. Display details of students who are born in 1980 or 1982.
10. Create a view that displays student courseno and its corresponding marks

## Part A - Answer 3:

## STUDENT, COURSE and SC Tables (MySQL Server)

### Create the Tables with Constraints

#### STUDENT Table

* + rollno is **Primary Key**
  + birthdate must not be **NULL**

CREATE TABLE STUDENT ( rollno INT PRIMARY KEY,

name VARCHAR(100) NOT NULL, class VARCHAR(20) NOT NULL,

birthdate DATE NOT NULL

);

#### COURSE Table

* + courseno is **Primary Key**
  + max\_marks must be **greater than 0**
  + pass\_marks must be **greater than 0** and **less than or equal to max\_marks**

CREATE TABLE COURSE (

courseno INT PRIMARY KEY, coursename VARCHAR(100) NOT NULL,

max\_marks INT CHECK (max\_marks > 0), pass\_marks INT,

CHECK (pass\_marks > 0 AND pass\_marks <= max\_marks)

);

#### SC Table (Student-Course Relationship Table)

* + **Composite Primary Key** (rollno, courseno)
  + marks must be **between 0 and 100**
  + rollno is a **Foreign Key** referencing STUDENT(rollno)
  + courseno is a **Foreign Key** referencing COURSE(courseno)

CREATE TABLE SC (rollno INT, courseno INT,

marks INT CHECK (marks BETWEEN 0 AND 100),

PRIMARY KEY (rollno, courseno),

FOREIGN KEY (rollno) REFERENCES STUDENT(rollno) ON DELETE CASCADE,

FOREIGN KEY (courseno) REFERENCES COURSE(courseno) ON DELETE CASCADE );

### Insert Sample Data into the Tables

#### Insert Students

INSERT INTO STUDENT (rollno, name, class, birthdate) VALUES (1, 'Alice', 'MCA', '1998-05-10'),

(2, 'Bob', 'MCA', '1997-08-15'),

(3, 'Charlie', 'BSc', '1999-02-20');

COMMIT;

Insert Courses

INSERT INTO COURSE (courseno, coursename, max\_marks, pass\_marks) VALUES

(101, 'Database Management System', 100, 40),

(102, 'Computer Networks', 100, 35),

(103, 'Data Structures', 100, 50);

COMMIT;

Insert Student-Course Marks

INSERT INTO SC (rollno, courseno, marks) VALUES

|  |  |  |
| --- | --- | --- |
| (1, | 101, | 85), |
| (1, | 102, | 72), |
| (2, | 101, | 40), |
| (2, | 103, | 55), |
| (3, | 102, | 30), |

(3, 103, 90);

COMMIT;

### Add Constraint to Ensure Marks are Between 0 and 100

(Already added in the SC table using CHECK (marks BETWEEN 0 AND 100))

### Add Composite Key Constraint to SC Table

(Already defined in SC table: PRIMARY KEY (rollno, courseno))

### Display Details of Students Taking 'Database Management System'

SELECT s.rollno, s.name, s.class, s.birthdate, c.coursename, sc.marks

FROM STUDENT s

JOIN SC sc ON s.rollno = sc.rollno

JOIN COURSE c ON sc.courseno = c.courseno

WHERE c.coursename = 'Database Management System';

### Display Students Who Scored More Than 70% in 'Computer Networks' and Have Not Failed in Any Subject

SELECT s.rollno, s.name FROM STUDENT s

JOIN SC sc ON s.rollno = sc.rollno

JOIN COURSE c ON sc.courseno = c.courseno

WHERE c.coursename = 'Computer Networks' AND sc.marks > (c.max\_marks \* 0.7)

AND s.rollno NOT IN (

SELECT sc.rollno FROM SC sc

JOIN COURSE c ON sc.courseno = c.courseno WHERE sc.marks < c.pass\_marks

);

### Display the Average Marks Obtained by Each Student

SELECT s.rollno, s.name, AVG(sc.marks) AS avg\_marks FROM STUDENT s

JOIN SC sc ON s.rollno = sc.rollno GROUP BY s.rollno, s.name;

### Select All Courses Where Passing Marks Are More Than 30% of Maximum Marks

SELECT \* FROM COURSE WHERE pass\_marks > (max\_marks \* 0.3);

### Display Details of Students Born in 1980 or 1982

SELECT \* FROM STUDENT WHERE YEAR(birthdate) IN (1980, 1982);

### Create a View That Displays Student Roll Number, Course Number, and Marks

CREATE VIEW Student\_Course\_Marks AS SELECT rollno, courseno, marks FROM SC; To fetch data from the view:

SELECT \* FROM Student\_Course\_Marks;