EXERCISE 3

Date: 05/03/2024

A **UNIVERSITY** database for maintaining information concerning students, courses, and grades in a university environment is given below.

The **STUDENT** file stores data on each student, the **COURSE** file stores data on each course, the **SECTION** file stores data in each section of a course, the **GRADE_REPORT** file stores the grades that students receive in the various section they have completed, and the **PREREQUISITE** files stores the prerequisites of each course.

STUDENT

Name	Student_number	Class	Major
Smith	17	1	CS
Brown	8	2	CS

COURSE

Course_name	Course_number	Credit_hours	Department
Intro to Computer Science	CS1310	4	CS
Data Structures	CS3320	4	CS
Discrete Mathematics	MATH2410	3	MATH
Database	CS3380	3	CS

SECTION

Section_identifier	Course_number	Semester	Year	Instructor
85	MATH2410	Fall	07	King
92	CS1310	Fall	07	Anderson
102	CS3320	Spring	08	Knuth
112	MATH2410	Fall	08	Chang
119	CS1310	Fall	08	Anderson
135	CS3380	Fall	08	Stone

GRADE_REPORT

Student_number	Section_identifier	Grade
17	112	B
17	119	C
8	85	A
8	92	A
8	102	B
8	135	A

PREREQUISITE

Course_number	Prerequisite_number
CS3380	CS3320
CS3380	MATH2410
CS3320	CS1310

Table of STUDENT:-

Name	Student_number	Class	Major
Brown	8	2	CS
Smith	17	1	CS

Table of COURSE:-

Course_name	Course_number	Credit_hours	Department
Intro to Computer Science	CS1310	4	CS
Data Structures	CS3320	4	CS
Database	CS3380	3	CS
Discrete Mathamatics	MATH2410	3	MATH

Table of SECTION:-

Section_identifier	Course_number	Semester	Year	Instructor
85	MATH2410	Fall	7	King
92	CS1310	Fall	7	Anderson
102	CS3320	Spring	8	Knuth
112	MATH2410	Fall	8	Chang
119	CS1310	Fall	8	Anderson
135	CS3380	Fall	8	Stone

Table of PREREQUISITE:-

Course_number	Prerequisite_number
CS3320	CS1310
CS3380	CS3320
CS3380	MATH2410

Table of GRADE REPORT:-

Student_number	Section_identifier	Grade
8	85	A
8	92	A
8	102	B
8	135	A
17	112	B
17	119	C

SQL QUERIES: -

1) Write appropriate MYQL DDL statements to define UNIVERSITY database.

- Create Database

- create database UNIVERSITY;
- use UNIVERSITY;

- Create Table: STUDENT

- create table STUDENT (Name varchar(10), Student_number int(2) primary key, Class int(2), Major varchar(20));

- Create Table: COURSE

- create table COURSE (Course_name varchar(30),Course_number varchar(20) primary key, Credit_hours int(2), Department varchar(15));

- Create Table: SECTION

- create table SECTION (Section_identifier int(3) primary key,Course_number varchar(20), Semester varchar(10),Year int(2),Instructor varchar(15),foreign key (Course_number) references COURSE(Course_number));

- Create Table: GRADE_REPORT

- create table GRADE_REPORT (Student_number int(2),Section_identifier int(3),Grade varchar(2),foreign key (Student_number) references STUDENT(Student_number),foreign key (Section_identifier) references SECTION(Section_identifier),primary key (Student_number,Section_identifier));

- Create Table: PREREQUISITE

- create table PREREQUISITE (Course_number varchar(20) ,Prerequisite_number varchar(20),foreign key (Course_number) references COURSE(Course_number),foreign key (Prerequisite_number) references COURSE(Course_number),primary key (Course_number,Prerequisite_number));

2) Write queries to insert values in all the five tables.

- Inserting Data to Table: STUDENT

- insert into STUDENT values("Smith",17,1,"CS"),("Brown",8,2,"CS");

- Inserting Data to Table: COURSE

- insert into COURSE values("Intro to Computer Science","CS1310",4,"CS"),("Data Structures","CS3320",4,"CS"),("Discrete Mathamatics","MATH2410",3,"MATH"), ("Database","CS3380",3,"CS");

- Inserting Data to Table: SECTION

- insert into SECTION values (85,"MATH2410","Fall",07,"King") ,(92,"CS1310","Fall",07, "Anderson"),(102,"CS3320","Spring",08,"Knuth"),(112,"MATH2410","Fall",08,"Chang"), (119,"CS1310","Fall",08,"Anderson"),(135,"CS3380","Fall",08,"Stone");

3) All Courses and grades of Smith

Course_name	Grade
Discrete Mathamatics	B
Intro to Computer Science	C

4) Names and grades of students who took 'Database' course offered in fall 2008

Name	Grade
Brown	A

5) Prerequisite for Database Course

Course_name
Data Structures
Discrete Mathamatics

6) Senior Students

Name
Brown

7) Courses taught by Professor King in 2007 and 2008

Course_name
Discrete Mathamatics

8) Details on section taught by King

Course_number	Semester	Year	No_of_student
MATH2410	Fall	7	1

9) Name and transcript of each senior majoring in CS

Name	Course_name	Course_number	Semester	Year	Grade
Brown	Discrete Mathamatics	MATH2410	Fall	7	A
Brown	Intro to Computer Science	CS1310	Fall	7	A
Brown	Data Structures	CS3320	Spring	8	B
Brown	Database	CS3380	Fall	8	A

- Inserting Data to Table: GRADE_REPORT

- insert into GRADE_REPORT values (17,112,"B"),(17,119,"C"),(8,85,"A"),(8,92,"A"), (8,102,"B"),(8,135,"A");

- Inserting Data to Table: PREREQUISITE

- insert into PREREQUISITE values("CS3380","CS3320"),("CS3380","MATH2410"), ("CS3320","CS1310");

3) Retrieve the list of all courses and grades of "Smith".

- select c.Course_name, g.Grade from STUDENT s inner join GRADE_REPORT g on s.Student_number = g.Student_number inner join SECTION se on g.Section_identifier= se.Section_identifier inner join COURSE c on se.Course_number=c.Course_number where s.Name="Smith";

4) List the names of students who took the section of 'Database' course offered in fall 2008 and their grades in that section.

- select s.Name,g.Grade from STUDENT s join GRADE_REPORT g on s.Student_number = g.Student_number inner join SECTION se on g.Section_identifier= se.Section_identifier inner join COURSE c on se.Course_number=c.Course_number where c.Course_name= "Database" and se.Semester="Fall" and se.Year=08 ;

5) List the prerequisites of the 'Database' course.

- select Course_name from COURSE where Course_number in (select p.Prerequisite_number from PREREQUISITE p join COURSE c on p.Course_number=c.Course_number where p.Course_number=(select Course_number from COURSE where Course_name="Database"));

6) Create a view to retrieve the names of all senior students majoring in 'CS' (computer science).

- create view seniors as select * from STUDENT where class=2;
- select Name from seniors;

7) Retrieve the names of all courses taught by Professor King in 2007 and 2008.

- select c.Course_name from COURSE c join SECTION s on c.Course_number = s.Course_number where s.Instructor="King";

8) For each section taught by Professor King, retrieve the course number, semester, year, and number of students who tool the section.

- select s.Course_number,s.Semester,s.Year,count(g.Student_number) as No_of_students from SECTION s join GRADE_REPORT g on s.Section_identifier=g.Section_identifier where s.Instructor="King" group by g.Section_identifier;

9) Retrieve the name and transcript of each senior student (Class=2) majoring in CS. A transcript includes course name, course number, credit hours, semester, year and grade for each course completed by the student.

- select s.Name,c.Course_name,c.Course_number,se.Semester,se.Year,g.Grade from student s join grade_report g on s.Student_number = g.Student_number join section se on g.Section_identifier = se.Section_identifier join course c on se.Course_number = c.Course_number where s.Class=2 and s.Major="CS";

10 a) Insert new student Johnson.

Name	Student_number	Class	Major
Johnson	25	1	Math

10 b) Update class of Smith to 2

Name	Student_number	Class	Major
Smith	17	2	CS

10 c) Inset new course Knowledge Engineering

Course_name	Course_number	Credit_hours	Department
Knowledge Engineering	CS4390	3	CS

10 d) Delete student Smith.

Name	Student_number	Class	Major
Brown	8	2	CS
Johnson	25	1	Math

10) Write SQL update statements to do the following on the database schema.

a) Insert a new student, < 'Johnson',25,1, 'Math'>, in the database.
- insert into STUDENT values("Johnson",25,1,"Math");
- select * from student where Student_number=25;

b) Change the class of student 'Smith' to 2.
- update STUDENT set Class=2 where Name="Smith";
- select * from student where Name="Smith";

c) Insert a new course, < 'Knowledge Engineering', 'CS4390', 3, 'CS'>.
- insert into COURSE values("Knowledge Engineering","CS4390",3,"CS");
- select * from course where Course_number="CS4390";

d) Delete the record for the student whose name is 'Smith' and whose student number is 17.
- delete from STUDENT where Student_name="Smith";
- select * from student;

RESULT: -

Queries are executed and output is verified.