IT2040-Database Management Systems

Semester 1,2021

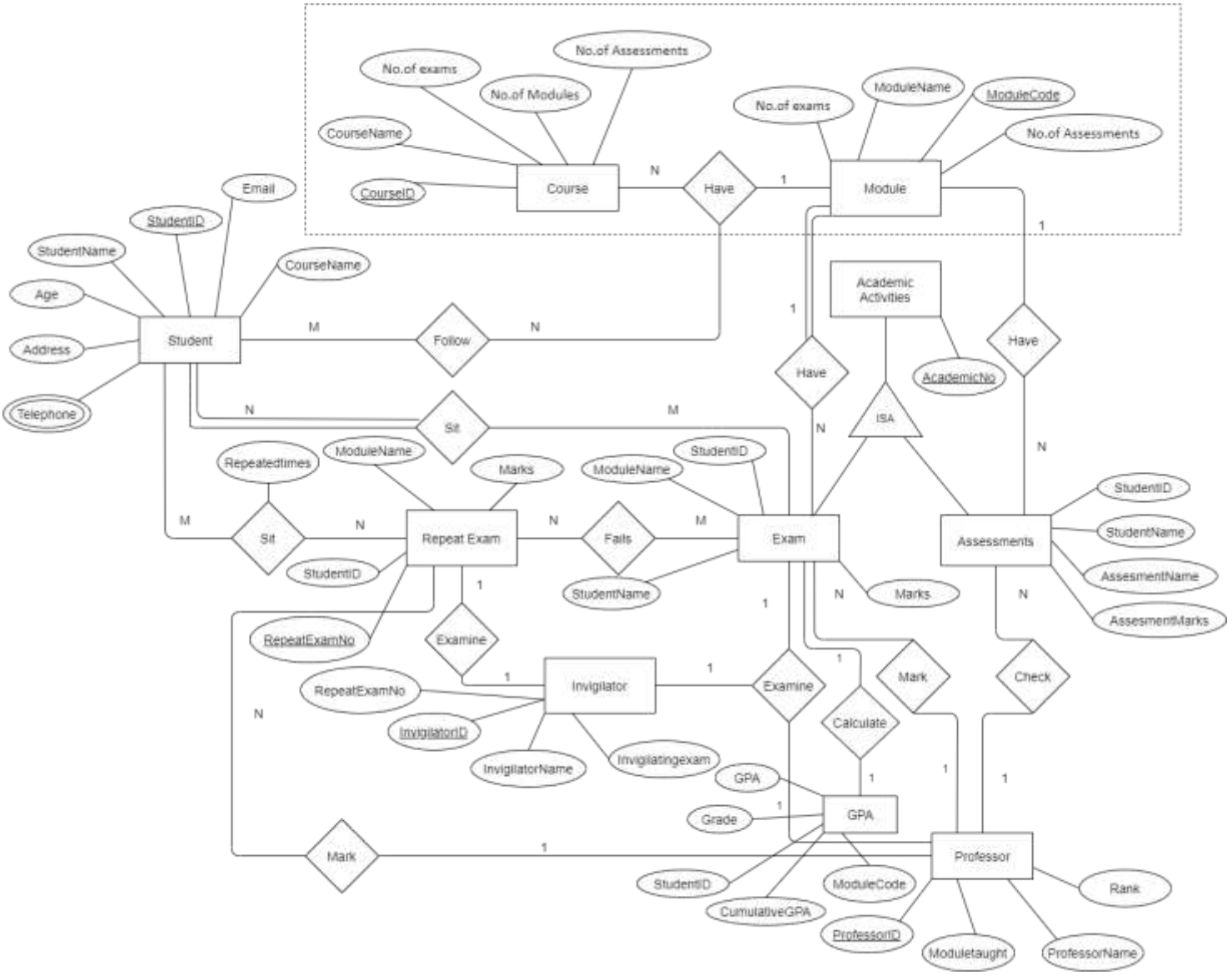
Exam and assessment information management system for a university

Student Registration Number : IT20219598

Group **:** Y2S1 8.1

Requirements

- A student has a Student ID, Student name, Age, Address, E-mail, Telephone, Course name.
- Course details are Course ID, Course name, No.of Modules, No.of exams, No.of Assessments.
- Professor has Professor ID, Modules taught, Professor name, Rank.
- Invigilator has Invigilator ID, Invigilator name, Invigilating exam, RepeatingExamNo.
- A repeating exam has RepeatingExamNo ,Student id, Module name, Marks.
- Exam has Student ID, Student name, Module name, Marks.
- Assessment has Student ID, Student name, Assessment name, Assessment marks.
- GPA has Student ID, Module code, Grade, gpa, cumulative gpa
- Module has Module code, Module name, No.of exams, No.of Assessments.
- A course can have many students but student can follow only one course.
- A student must sit for all examinations in the course. If he/she is unable to sit or fails, then can do the repeat examination.
- Students can do their assessments.
- A course can have several modules and same module can be in several courses.
- Module must have an examination.
- A module can have several assessments, but assessment has only one module.
- Professor can teach several modules and module can have several professors.
- A Professor can mark several examinations and assessments.
- An invigilator can examine one exam at a time.
- Professor can also examine the exam.
- Exam and Assessments do not cover all the academic activities in the University.



Student

StudentID	StudentName	Age	Address	E-mail	Coursename	AcademicNo

CONSTRAINT fk22 FOREIGN KEY(CourseName) References Course(CourseName),
CONSTRAINT fk1 FOREIGN KEY(AcademicNo) References Exam(AcademicNo)

Telephone

StudentID	Telephone

CONSTRAINT fk2 FOREIGN KEY(StudentID) References Student(StudentID

Professor

Professor ID	Modulestaught	Professor name	Rank

CONSTARINT fk21 FOREIGN KEY(Moduletaught) References Module(ModuleName),

Invigilator

InigilatorID	InvigilatorName	InvigilatingExam	RepeatExamNo

CONSTRAINT fk13 FOREIGN KEY(Invigilatingexam) References Exam(AcademicNo),
CONSTRAINT fk14 FOREIGN KEY(RepeatingExamNo) References RepeatingExam(RepeatingExamNo)

Exam

AcademicNo	StudentID	StudentName	ModuleCode	ModuleName	Marks	ProfessorID

CONSTARINT fk16 FOREIGN KEY(ModuleCode) References Module(ModuleCode),
CONSTRAINT fk12 FOREIGN KEY(AcademicNo) References AcademicActivities (AcademicNo),
CONSTRAINT fk10 FOREIGN KEY(ProfessorID) References Professor (ProfessorID)

Assessment

AcademicNo	StudentID	StudentName	ModuleCode	ProfessorID	Assesment Name	Assessment Marks

CONSTARINT fk15 FOREIGN KEY(ModuleCode) References Module(ModuleCode),
CONSTRAINT fk11 FOREIGN KEY(AcademicNo) References AcademicActivities (AcademicNo),
CONSTRAINT fk9 FOREIGN KEY(ProfessorID) References Professor (ProfessorID)

Module

ModuleCode	ModuleName	No.OfExams	No.OfAssessments

RepeatExam

RepeatExamNo	StudentID	ProfessorID	ModuleName	Marks

CONSTRAINT fk9 FOREIGN KEY(ProfessorID) References Professor (ProfessorID)

Course

CourseID	Coursename	No.OfExams	No.OfAssessments	No.OfModules	ModuleCode

CONSTARINT fk30 FOREIGN KEY(ModuleCode) References Module(ModuleCode)

GPA

StudentID	ModuleCode	Grade	GPA	CumulativeGPA

CONSTRAINT fk20 FOREIGN KEY(StudentID) References Student(StudentID)
CONSTARINT fk30 FOREIGN KEY(ModuleCode) References Module(ModuleCode)

Sit

StudentID	AcademicNo

CONSTRAINT fk3 FOREIGN KEY(AcademicNo) References Exam(AcademicNo),
CONSTRAINT fk4 FOREIGN KEY(StudentID) References Student(StudentID)

RepeatSit

StudentID	RepeatingExamNO

CONSTRAINT fk5 FOREIGN KEY(RepeatingExamNo) References Repeating Exam(RepeatingExamNo),
CONSTRAINT fk6 FOREIGN KEY(StudentID) References Student(StudentID)

Follow

StudentID	CourseID	ModuleCode

CONSTRAINT fk7 FOREIGN KEY(CourseID) References Course(CourseID),
CONSTRAINT fk8 FOREIGN KEY(StudentID) References Student(StudentID),
CONSTARINT fk30 FOREIGN KEY(ModuleCode) References Module(ModuleCode)

Fails

RepeatExamNo	AcademicNo

CONSTRAINT fk15 FOREIGN KEY(AcademicNo) References Exam(AcademicNo),
CONSTRAINT fk16 FOREIGN KEY(RepeatingExamNo) References RepeatingExam(RepeatingExamNo)

AcademicActivities

AcademicNo

Exam

AcademicNo	StudentID	StudentName	ModuleCode	ModuleName	Marks	ProfessorID

CONSTARINT fk16 FOREIGN KEY(ModuleCode) References Module(ModuleCode),
CONSTRAINT fk12 FOREIGN KEY(AcademicNo) References AcademicActivities (AcademicNo),
CONSTRAINT fk10 FOREIGN KEY(ProfessorID) References Professor (ProfessorID)

Assessments

AcademicNo	StudentID	StudentName	ModuleCode	ProfessorID	Assesment Name	Assessment Marks

CONSTARINT fk15 FOREIGN KEY(ModuleCode) References Module(ModuleCode),
CONSTRAINT fk11 FOREIGN KEY(AcademicNo) References AcademicActivities (AcademicNo),
CONSTRAINT fk9 FOREIGN KEY(ProfessorID) References Professor (ProfessorID)

AcademicActivities

AcademicNo

- Option 1 is used to map the ISA relationship, because option 3 & option 4 cannot be used as the sub classes have relationships and option 2 cannot be used because exam and assessments do not cover all the academic activities in the university.

Query 1

```
SELECT s.StudentID,m.ModuleCode, p.ProfessorID, SUM(Marks)
FROM Student s, Professor p, Exam e, Module m
WHERE s.StudentID=e.StudentID AND p.ProfessorID=e. ProfessorID AND m.ModuleCode=e.ModuleCode
GROUP BY s. StudentID,m.MouleCode,p.ProfessorID
HAVING SUM(Marks)>200
```

Query 2

```
SELECT s.StudentName
FROM m.Module,Exam e,g.GPA
WHERE s.StudentID=e.StudentID AND m.ModuleCode=e.ModuleCode AND
        s.StudentID=g.StudentID AND g.GPA =(SELECT MAX(GPA)
                                            FROM GPA)
```

Procedure

To get count of StudentIDs and update marks of given student and academic number.

```
CREATE PROCEDURE studentupdate (@sid VARCHAR(20),@acn VARCHAR(10)),@mn VARCHAR(40)
                        output
AS
BEGIN
        SELECT @mn=COUNT(StudentID)
        FROM Exam e
        UPDATE EXAM
        SET Marks=Marks+10
        WHERE  StudentID=@sid AND AcademicNo=@acn
END;
DECLARE @m
EXEC studentupdate 'IT20219598','IT1001', @m output
PRINT @m
```

Trigger

Ensure that student doesn't sit for same exam for more than 3 times

```
CREATE TRIGGER chechrepeatedtimes
ON RepeatSit
FOR INSERT, UPDATE
AS
BEGIN
DECLARE @rn int
DECLARE @sid
DECLARE @total int
SELECT @rn=RepeatExamNo FROM INSERTED
SELECT @total=COUNT(*)
FROM RepeatSit
WHERE RepeatExamNo=@rn AND @sid=StudentID
IF @total>3
        BEGIN
                PRINT 'Repeating times has exceeded'
                ROLLBACK TRANSACTION
        END
END
```