Database Design

Title: Online Examination System

By: Pratima Kshetry (pkshetry@umd.edu)

Table of Contents

Introduction

This project aims to create a database design for Online Examination System for any educational institutions. This system would allow instructors to conduct an online examination for their students for a given subject. The database for this system is designed in MySQL. It consists of 13 tables. The major entities of this system are:

ENTITIES	**ATTRIBUTES**
Role	RoleID, Role, Comment
User	UserID, PasswordHash, FirstName, LastName
UserRole	UserRoleID, UserID, RoleID
Subject	SubjectID, SubjectName, Comment
Teacher_Subject	TeacherSubjectID, TeacherID, SubjectID
QuestionType	QuestionTypeID, Type, Description
Questions	QuestionID, QuestionTypeID, SubjectID,Description,
QuestionOption	QuestionOptionID, QuestionID, IsCorrect,Text,
Exam	ExamID, TeacherID, Date, Duration
ExamQuestion	ExamQuestionID, ExamID, QuestionID,
ExamRegistration	RegistrationID,StudentID, ExamID, GPA
StudentExamAnswer	StudentExamAnswer, StudentID, ExamQuestionID, QuestionsAnswerID, AnswerText
StudentExamGrader	StudentExamGraderID, RegistrationID, GraderID

Description of Entities:

Role: This table defines the role information of a user. Role can be an admin (super user), Teaching Assistant, teacher, student as described and defined by the Role table.

User: This table contains user information and credentials. Each user can be assigned a role as defined in the Role table, and this is mapped in UserRole table. Thus, a user can have multiple roles, such as a user can be both teacher and admin.

Subject: This table will hold the subject/course name and other details about the subject.

Teacher_Subject: This table defines a user who is a teacher for a given subject. Thus, a user of the system can be a teacher of multiple subjects i.e. one teacher can teach many subjects.

QuestionType: This table will hold the information of whether the question is subjective or objective. Objective questions can be multiple choice.

Questions: This table contains questions for a given subject and also the question type.

QuestionOptions: This table will contain the multiple options for a given question, if the question is of objective type and will ask the user to type the answer in a text box if subjective. This table also contains an "IsCorrect" bit field that holds "1" value for a correct answer and "0" for an incorrect answer.

Exam: This table will contain the exam details of the given subject conducted by the given teacher.

ExamQuestion: This table will map the questions to be asked in a given exam with the Questions table.

ExamRegistration: This table will contain the registration details of a student for a given examination. Every student should register to give an examination. The final grades/GPA will also be stored in this table.

StudentExamAnswer: This table will map the answers given by the student to the questions asked in the given examination. Therefore this table contains the actual snapshot of the exam given by the student. It also has the "AnswerText" field (nullable), that would contain the answertext if the question is subjective.

StudentExamGrader: This table will contain the information of user (teacher) who has graded the given exam for a given student. Whenever there are subjective questions in exam, such questions should be graded independently by the grader. Grader can be a teacher or any authorized personnel such as Teaching Assistant. A subject can have more than one grader.

Logical Design:

Logical design is the process of constructing a model of information, which can then be mapped into storage objects supported by a Database Management System (Kamalnathan, October, 2009).

It comprises of following steps:

- Table Generation From an ER Model
- Normalization of Tables

An ER model is the graphical representation of logical relationships of tables before creating a database. It allows users to have a clear pictorial representation of the potential entities involved in the system and its respective relationship with other entities of the system.

The initial ERD of this system comprised of 14 tables. The only table that has been removed is the Questions_Answers table. This table was used to store the correct answer option for a given question. However, the same purpose of the Questions_Answers table was fulfilled by adding an "IsCorrect" bit field (indicated by 0 and 1) in the Question_Option table.

Building an ERD model for this system allowed us to identify the nature of relationships between the entities. These relationships were mostly many to many

A screenshot of the final ERD model developed for the system is attached below:

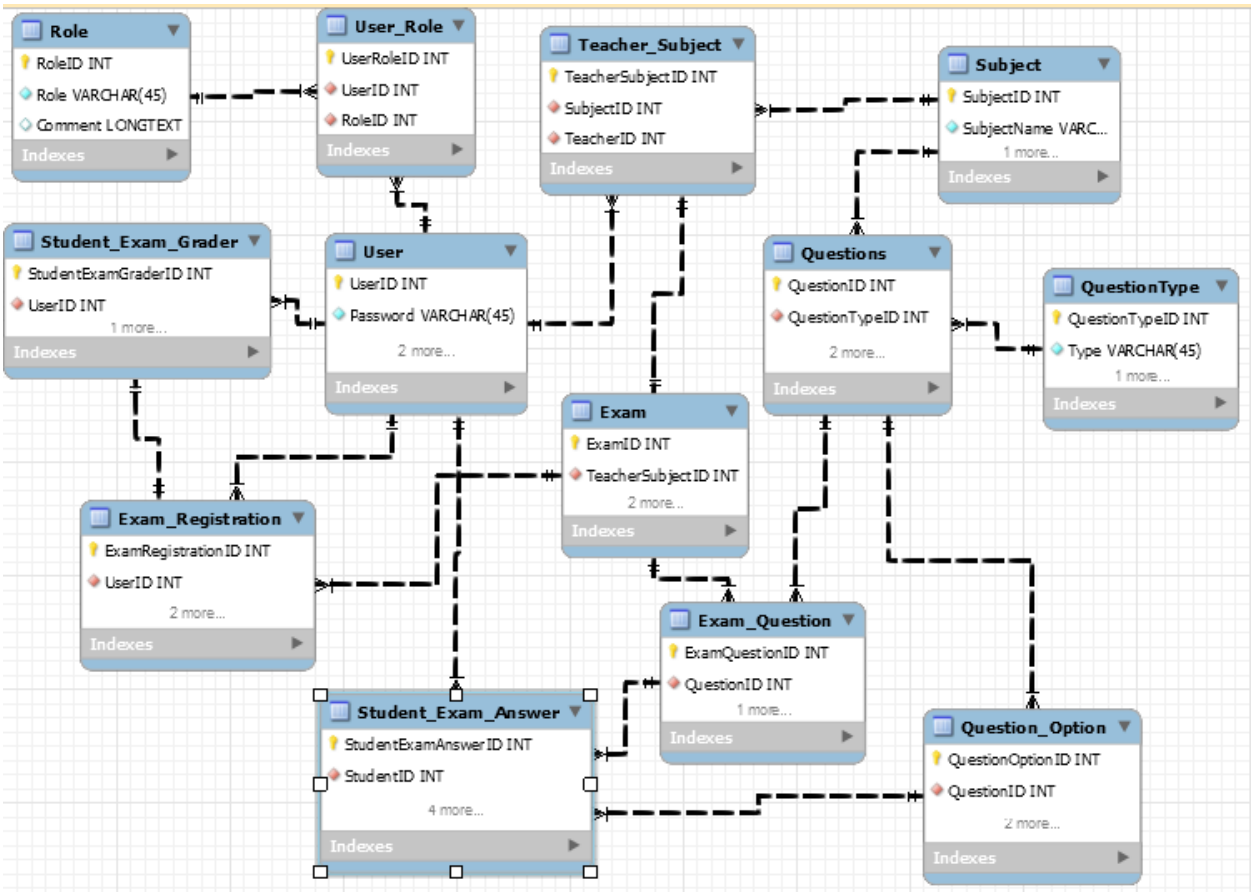

Fig. ERD Model

Physical Design:

Physical Design of the database was implemented using the Forward Engineering techniques. The "Forward Engineer" Option in MySQL workbench allows to create a database from an existing ERD model.

Theoretically, Forward engineering is the process of building from a high-level model or concept to build in complexities and lower-level details (Cory Janssen).

Since the deletion of Questions_Answers table was done before a "Forward Engineering" was performed to the final ERD, and henceforth no changes has been made, Reverse Engineering is not performed during this project.

After generating database from the existing ERD, I populated the database thus created, with appropriate data. The data inserted in the table are realistic and represent the real world data.

Issues Faced and Learnt:

Few issues encountered during this project were:

1. The forward engineering in my case did not work (at first) when I marked the PKs and CPKs as Unsigned I later learnt that this was similar to the case of data type mismatch. While I changed the PK and CPKs to Unsigned, but ignored to do the same to its corresponding FK. As a result, if the Singed/Unsigned characteristics of a PK and its corresponding FK do not match, then sets of values they can have would be different, which would be a violation of database normalization. As a solution to this problem, I changed the PKs and its corresponding FKs back to Signed characteristic.

2. During the initial design phase of the project, I had initially created separate tables for students, teachers and admin. This was only taking space in the database while storing similar or related information. Later, during one of the assignments, I learnt that the same UserID can be used as a pointer to both teachers and students. Therefore, the ERD has been designed such that the same UserID points to all kinds of user of this system.

3. Initially I had created an additional table "Questions_Answer" that stored the correct answer for each given question. This only took up an additional space in the database. Later, this table was removed, and instead a "Is correct" bit field was added into the "Question_Option" field that stored "0" for an incorrect answer while a "1" for the correct answer.

4. I also had issues mapping the correct answer to a given question. At first, one option were mapped to several questions. To solve this issue, I wrote a select query as, *"Select QuestionOptionID, QuestionID, text, IsCorrect from Question_Option where questionID=1",* so that I can see which question has which option has set itself as Correct. Then, I ran the following query,*"Update Question_Option SET IsCorrect=1 where QuestionID=1 and QuestionOptionID=1",* multiple times (only changing the questionID and questionOption ID for

later queries) to set any one option of the given questionID as correct answer indicated by "1" changing.

5. I have been able to write join query for more than two tables. Before doing this project, I could write join queries for two to three tables. Since this project has strictly normalized tables, I was required to conduct a join query for at least three tables. This was a little intimidating at first, but was a great learning experience later.

6. I learnt about Views that could be used to join and simplify multiple tables into a single virtual table. Views made the complex queries look simpler.

7. I was also able to make a distinction between "Group By" and "Order By" query. The "Group By" query is used in conjunction with the aggregate functions to group the result-set by one or more columns while the "Order By" query is used to sort the output results.

Scope of the Project:

The system can be used in educational institutions as well as organizations conducting standardized tests or a regular examination where an evaluation of students' is required.

However, for this project, only database for the given system has been designed i.e. the front-end of the system has not be designed.

Future Enhancements:

1. A front end interface can be developed to make the data entry process simpler.

2. As of now, the project only displays the answers selected by student for a given question, but does not tell whether the answer selected by student is correct or incorrect. Although, Quertion_Option table has a bit field that has zero (0) for an incorrect answer and a "1" for correct answer, no further computation has been made that shows whether the answer selected by answer in correct or incorrect. Hence, we can add this feature to the project where the system prompts/notify the user of the incorrect answer.

3. This project can later also be integrated with college MIS, so that the system as a whole serves the purpose of college MIS and also allows conducting online examination in a college.

4. This project can also serve to provide Course Effectiveness analytics later when the large volume of data becomes available. Such analytics will help college administrators discover courses that has low performance of students, and also courses that students are most interested in.

CRUD operations:

Query1. UserName and their Respective Roles

```
/* Users Name and their respective Role */
CREATE view v_UsernameRole as
select concat(us.FirstName, us.lastname) as "UserName", rr.role as "Role" from user us, role rr, user_role ur
 WHERE us.UserID=ur.UserID and ur.RoleID=rr.RoleID and rr.Role="student" ORDER BY us.FirstName ASC LIMIT 10;

select username, role from v_userNameRole;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

username	role
AllenTuladhar	Student
AnnieWood	Student
DayaSubedi	Student
JohnChettri	Student
LilyYadav	Student
ManojKc	Student

Query2. Username and their GPA in respective Subjects

```
Create view v_UserNameSubjectGPA as
select concat(us.FirstName,' ', us.LastName) as "Name", subject.SubjectName as "Subject",
exam_registration.GPA as "GPA" from user us JOIN exam_registration ON
us.UserID=exam_registration.UserID
JOIN exam ON exam_registration.ExamID=exam.ExamID
JOIN teacher_subject ON exam.TeacherSubjectID=teacher_subject.TeacherSubjectID
JOIN subject on teacher_subject.SubjectID=subject.SubjectID  ;

Select name, subject, gpa from v_usernameSubjectGPA order by name asc  ;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

Name	Subject	GPA
Allen Tuladhar	Statistics	3.5
Allen Tuladhar	Users and Use Context	3.6
Annie Wood	Statistics	3.1
Christina Dawn	Data Analytics	3.5
Christina Dawn	Users and Use Context	3.5
John Chettri	Statistics	3.56

Query3. Instructor Name and subjects taught by them

```
/* Instructor name and subjects sorted by Subject name in Descendig Order*/
CREATE view v_InstructorNameSubjectName as
select concat(us.FirstName,"  ", us.lastname) as "InstructorName", sub.SubjectName from user us,
subject sub, teacher_subject teachSub WHERE us.UserID=teachSub.TeacherID
AND teachSub.SubjectID=sub.SubjectID ;

Select InstructorName, subjectName from v_InstructorNameSubjectName order BY subjectName;
```

sult Grid | Filter Rows: | Export: | Wrap Cell Content:

InstructorName	SubjectName
Pratima Karki	Aeronautics
Dipesh Karki	Biology
Sudeep Karki	Business Intelligence
Andrea Wiggins	Data Analytics
Natalie Brees	Data Analytics
Natalie Brees	Data Analytics
Vedat Diker	Database
Shree Karkee	Digging Into Data
Pratima Karki	Generall Science
Sudeep Karki	GIS
Dipesh Karki	Human Biology
Natalie Brees	InterNetworking

Query4. Students Registered for Statistics Examination

```
/* Which students are registered for which exams*/
CREATE view v_StatisticsExam as
select concat (us.FirstName," ",us.LastName) as "Name", subject.SubjectName, exam_registration.ExamRegistrationID
from user us join exam_registration On us.UserID=exam_registration.UserID
JOIN exam on exam_registration.ExamID=exam.ExamID
JOIN teacher_subject ON exam.TeacherSubjectID=teacher_Subject.teacherSubjectID
JOIN subject ON teacher_subject.SubjectID=subject.SubjectID where subject.subjectName="Statistics";

select name,subjectName,ExamregistrationID from v_StatisticsExam order By name;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

Name	SubjectName	ExamRegistrationID
Allen Tuladhar	Statistics	10
Alok Katwal	Statistics	55
Andy Adams	Statistics	62
Annie Wood	Statistics	9

Query 5. Grader of a Subject

```
CREATE view v_grader as
SELECT concat(us.firstname," ", us.lastname) as "GraderName", examReg.examRegistrationID, sub.subjectName
  from user us join student_exam_grader examReg ON
       us.userID=examReg.UserID
    INNER JOIN teacher_subject ON
       examReg.UserID=teacher_subject.TeacherID
      INNER JOIN subject sub ON
       teacher_subject.SubjectID=sub.SubjectID ;

select GraderName, subjectname from v_grader where v_grader.subjectName="database";
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

GraderName	subjectName
Vedat Diker	Database

Query 6. Student's selected Answer for given subject

```
CREATE view v_StudentAnswer as
SELECT concat(us.firstname," ", us.lastname) as "UserName", sub.subjectname as "Subject",
ques.description as "Question", question_option.Text as "SelectedAnswer" from user us
JOIN exam_registration ex_reg ON us.UserID=ex_reg.UserID
JOIN  exam ex ON ex_reg.ExamID=ex.ExamID
JOIN teacher_subject ON exam.TeacherSubjectID=teacher_subject.SubjectID
JOIN subject sub ON teacher_subject.subjectID= sub.SubjectID
JOIN questions ques ON sub.subjectID=ques.SubjectID
JOIN exam_question ON ques.QuestionID=exam_question.QuestionID
JOIN student_exam_answer ON exam_question.ExamQuestionID=student_exam_answer.ExamQuestionID
JOIN question_Option ON student_exam_answer.QuestionOptionID=question_Option.QuestionOptionID
WHERE sub.SubjectName="Database" and us.userID=3;

SELECT username, subject, question, SelectedAnswer from v_StudentAnswer;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

UserName	Subject	Question	SelectedAnswer
Christina Dawn	Database	1. The DBMS acts as an interface between wha…	A.Database application and the database
Christina Dawn	Database	2. Which of the following products was an early…	D. R:base

Query 7. Highest GPA for each given Subject

```
Create View V_GPATABLE as select concat(us.FirstName,' ', us.LastName) as "Name", exam_registration.ExamID, subject.SubjectName,
exam_registration.GPA from user us JOIN exam_registration ON us.UserID=exam_registration.UserID
JOIN exam ON exam_registration.ExamID=exam.ExamID
JOIN teacher_subject ON  exam.TeacherSubjectID=teacher_subject.TeacherSubjectID
JOIN subject on teacher_subject.SubjectID=subject.SubjectID;

   Select Name,SubjectName,ExamID,GPA from V_GPATABLE v1, (select subjectName as Sn,max(GPA) as mxgpa from V_GPATABLE
   group by subjectName) v2 where v1.subjectName=v2.Sn and v1.GPA=v2.mxGPA order by SubjectName
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

Name	SubjectName	ExamID	GPA
Shankar Devkota	Data Analytics	5	3.95
Manoj Kc	Database	3	3.25
Ruby Yadav	Digging Into Data	12	3.83
Samira Regmi	Machine Learning	15	3.95
Suman Gautam	Marketing Analytics	16	3.83
John Chettri	Maths	10	3.55

Query 8. Average GPA for each subject of the system

```
Create View V_GPATABLE as select concat(us.FirstName,' ', us.LastName) as "Name", exam_registration.ExamID,
subject.SubjectName, exam_registration.GPA from user us
JOIN exam_registration ON us.UserID=exam_registration.UserID
JOIN exam ON exam_registration.ExamID=exam.ExamID
JOIN teacher_subject ON  exam.TeacherSubjectID=teacher_subject.TeacherSubjectID
JOIN subject on teacher_subject.SubjectID=subject.SubjectID;

/** Average GPA*/
select subjectName, round(avg(GPA),2) as "Average GPA of Students" from v_gpatable
group by subjectName order by subjectName;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

SubjectName	Average GPA of Students
Data Analytics	3.64
Database	2.98
Digging Into Data	3.58
Machine Learning	3.85
Marketing Analytics	3.82
Maths	3.50

Query 9. Total Users Registered as Students in the system

```
/* Total users registered as Students*/
select  count(concat (FirstName,LastName)) as "Total Users" from user JOIN user_role on
user.UserID=user_role.UserID JOIN role ON user_role.RoleID=role.RoleID where role.Role="student" ;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

Total Users
16

Query 10. Question Bank for a given Subject

```
   /* question Bank for each subject*/
   CREATE view v_QuestionBank as
select ex.examID as "ExamID", sub.subjectname AS "SubjectName", ques.description as "Question" from exam ex
  JOIN teacher_subject ts ON
  ex.TeacherSubjectID=ts.TeacherSubjectID
  JOIN  subject sub ON
  ts.SubjectID=sub.SubjectID
  JOIN questions ques ON
  sub.SubjectID=ques.SubjectID where sub.subjectName="Database";

  SELECT ExamID, subjectName, Question from v_questionBank;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

ExamID	SubjectName	Question
3	Database	1. The DBMS acts as an interface between what two components of an enterprise-class databa...
3	Database	2. Which of the following products was an early implementation of the relational model develop...

Query 11. List of Subjective Questions

```
  /*List of Subjective Questions */
select sub.subjectname, Type as "Question Type", ques.description AS "Question" from subject sub
  JOIN questions ques on
  sub.SubjectID=ques.SubjectID
  join questiontype qt on
  ques.QuestionTypeID=qt.QuestionTypeID
  WHERE qt.Type="subjective"  GROUP  BY sub.SubjectName;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

subjectname	Question Type	Question
ASP.Net Programming	Subjective	What is a view state
Computer Science	Subjective	2.What is virtual memory? Differentiate between logical & physical memory.
Sociology	Subjective	1. Describe the scope of sociology
Statistics	Subjective	Explain Central Limit Theorem

Requirement Table:

Query Name	Req. A	Req. B	Req. C	Req. D	Req. E
Query 1	X	X		X	
Query 2	X				
Query 3	X	X		X	
Query 4	X	X			
Query 5	X				
Query 6	X	X			
Query 7	X	X	X		X
Query 8	X		X		
Query 9			X		
Query 10	X				
Query 11			X	X	

References :

1. Kamalnathan, October, 01, 2009. Available at: http://www.prowareness.com/blog/database-design-conceptual-design-logical-design-physical-design/

2. Cory Janssen. Available at: http://www.techopedia.com/definition/19445/forward-engineering