Project report

Introduction

This project implements a test management system. The test management system is used by teachers and students of an educational institution to conduct timed tests. The teacher creates questions and stores them in the system along with their answers for the different courses. The system maintains this collection of questions as a question bank and allows the teacher to create tests from these questions. These tests only contain 10 multiple choice questions which are selected by the teacher. The students can take one test at a time and the system grades the test. The students and teachers can view the grades scored in the different tests.

The test management system is implemented as a java application along with an MS SQL server database. The java application provides the user interface for the teachers and students to interact with the test management system. The database stores information about the teachers, and students, their courses, questions, tests and grades. The database uses stored procedures providing the relevant information to the java application and trigger to grade a student test. We experimented with the query execution time by indexing the most frequently used table, users.

Working of the application

The test management system has two types of users-teachers and students. Both types of users need to first login to the system by providing their username, password and role type (teacher or student). The system verifies the user credentials using a stored procedure. On successful login, the user has to select the course of interest. The information about courses taught by a teacher and courses taken by a student is maintained in a different system like the course registration system and our test management system imports this data from it. Using the course information, the system displays the different courses available to the user. The user selects the course of interest.

For teachers, the system allows them to add or modify questions, create tests and view all student grades for each test. The teacher can add a new question to the question bank of the selected course. This question should have 4 answer choices and the correct choice must also be mentioned. The question is stored in the database and each question is assigned a unique identifier called questionId. To modify an existing question, the teacher must choose the question to be modified

Front –end, how to use the application, screen shots.

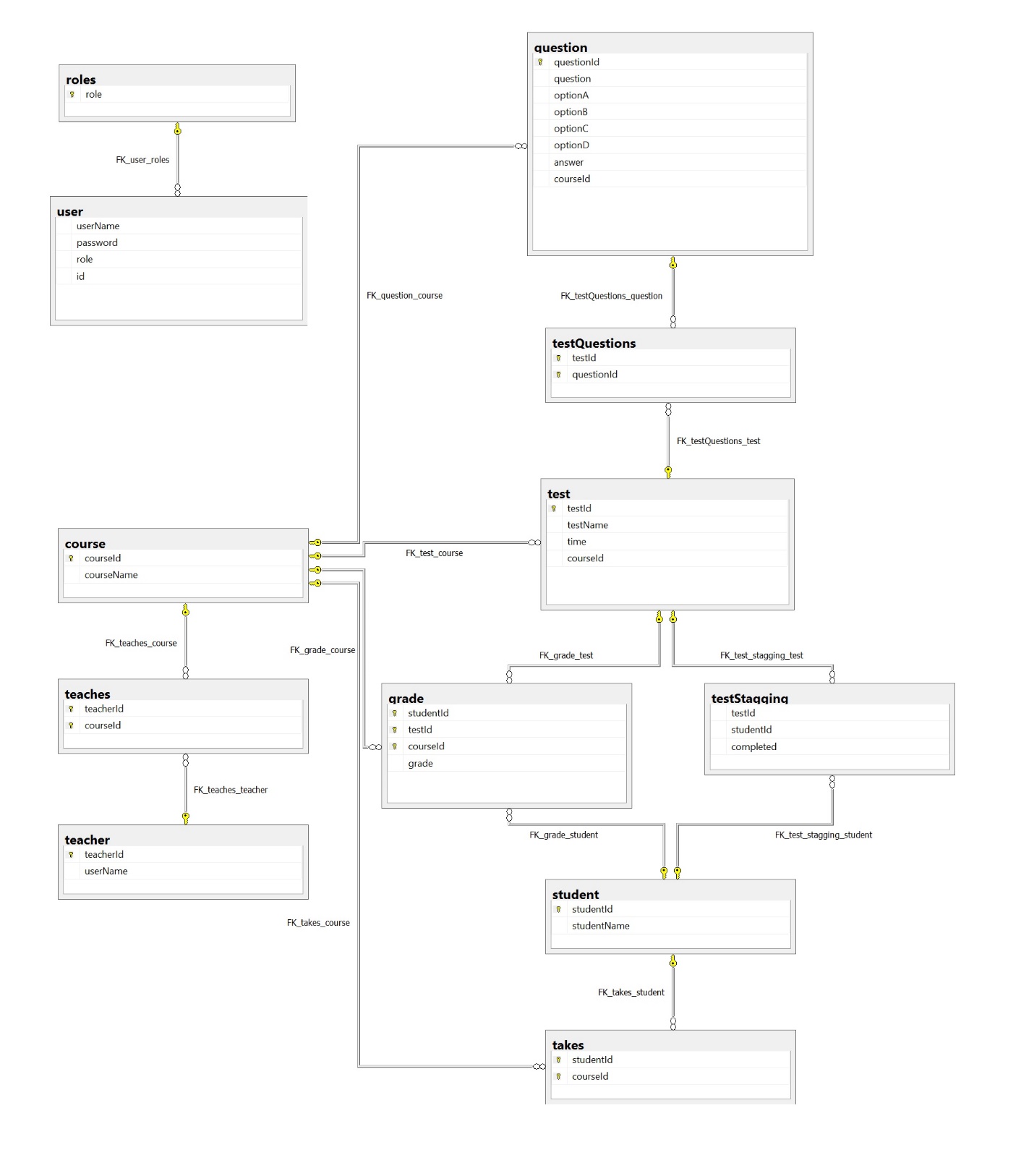
Login, choose course, set of actions depend on the role. Teacher can add /modify qtestions, create tests, view grades. Student can take test or view grades.

Teachers & students

Teachers add/modify questions & create tests.

Students take tests and view grades

Database Schema



The database schema of the testing application consists of the following tables:

1. course: This table stores information about the course like course id and course name. The course id is a unique identifier assigned to each course and hence is chosen as the primary key of the table.

2. grade: This table stores information about the grade scored by each student in each test for each course. The columns testId, courseId and studentId are foreign keys relating the grade table to test, course and student tables respectively. The combination of all the three foreign keys is used as the primary key of this table. The grade column can have the values ‘A’, ‘B’, ‘C’, ‘D’, ‘E’ or ‘F’.

3. question: This table stores information about the questions and serves as a question bank for all the courses. Each question is related to one course and hence the courseId is added as a foreign key to the table. Each question is assigned a unique identifier called questionId and s used as the primary key of the table. Each question has four answer options and one correct answer.

4. roles: This table stores the different user roles present in the testing system. There are two user roles-student and teacher.

5. student: This table stores information about students like student name and unique identifier called studentId which is used a primary key of the table.

6. takes: This table lists the course ids of the courses taken by each student. This table is assumed to be populated by an external system like student registration system which knows the courses taken by all students.

7. teacher: This table stores information about teachers like teacher name and unique identifier called teahcerId which is used a primary key of the table.

8. teaches: This table lists the course ids of the courses taught by each teacher. This table is assumed to be populated by an external system which knows the courses taught by all teachers.

9. test: This table stores information about every test created for all courses. Each test is assigned a unique identifier called testId and hence is used the primary key of the table. Every test has a name and a time duration. Every test is created for a course and hence the courseId is added as a foreign key to represent the course for which the test is created.

10. testQuestions: For all tests, this table lists the questionIds of all the questions that are assigned to the test. Every test has ten questions and hence this table will have ten records, one for each question in the test.

11. testStagging: This table is used to temporarily store information about a test that is being taken a student. It stores the studentId of the student taking the test, the testId of the test being taken and the status of the test. The status indicates if the test is being taken or completed.

12. user: This table stores user information like username and password, which is used for verifying the user credentials while logging the user into the testing system. It also stores the user role. Each user is assigned a unique identifier called id, which is used as the primary key of the table.

It can be noted that all the columns of the tables have atomic values. Hence, the database schema is in first normal form. In each table, all the non-prime columns are functionally determined by the primary key only. There are no transitive dependencies. Hence, the database schema is also in second and third normal form.

Topics implemented

Triggers: 1

Stored procedures: 11

Index:1

Conclusion