Project Report::Online Exam Management System—Sql Project Documentation

Submitted By: Jajam Prasanna

Roll No:22KQ1A5412

Department:AIDS-A

College:Pace Institute Of Technology&Science

Introduction

The Online Examination System is an SQL-based database project that enables conducting and managing online tests for students. This system allows administrators to create exams and questions, while students can register, take exams, and view their results. All records are stored and managed in a relational database using SQL operations and functions.

Objective

The objective of this project is to design a relational database that:

- Stores student details
- Manages exams and associated questions
- Records and evaluates results
- Automatically updates student exam status based on performance

Modules

1. Student Management:

This module allows the registration and management of student information such as name, email, and password.

2. Exam Management:

This module allows administrators to create exams with a specific name, duration, and total marks.

3. Question Bank Management:

This module stores all questions and multiple-choice options for each exam. It also stores the correct option to help in auto-evaluation.

4. Result Management:

This module records student scores and automatically sets pass/fail status using a trigger.

ER Diagram Explanation

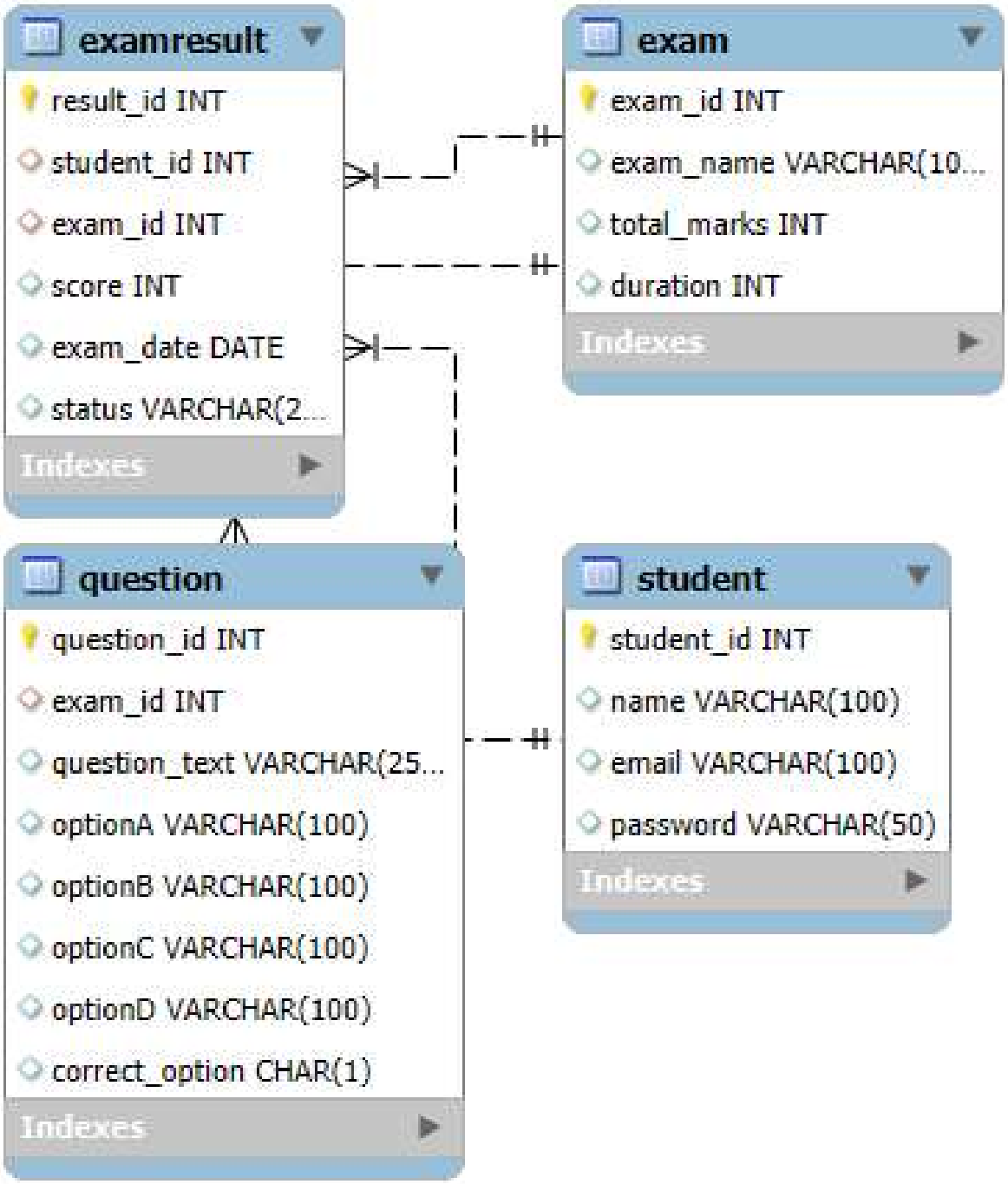
*The ER diagram represents the relationship between the main entities of the Online Examination System.

* The **Student** entity contains personal details of users who take the exams.

* Each student can have multiple examination results, which is why there is a one-to-many relationship between **Student** and **ExamResult**.

* The **Exam** entity stores information about each test, and it is also linked to the **Question** entity in a one-to-many relationship because one exam contains multiple questions.

*The **ExamResult** table connects the **Student** and **Exam** entities and stores the score, date, and status of each exam attempt. This structure allows the system to manage students, schedule exams, store questions, and track results in a normalized way.


examresult
result_id INT
student_id INT
exam_id INT
score INT
exam_date DATE
status VARCHAR(2...
Indexes
exam
exam_id INT
exam_name VARCHAR(10...
total_marks INT
duration INT
Indexes
question
question_id INT
exam_id INT
question_text VARCHAR(25...
optionA VARCHAR(100)
optionB VARCHAR(100)
optionC VARCHAR(100)
optionD VARCHAR(100)
correct_option CHAR(1)
Indexes
student
student_id INT
name VARCHAR(100)
email VARCHAR(100)
password VARCHAR(50)
Indexes

SQL Code

→CREATE DATABASE / CREATE TABLE

```
CREATE DATABASE OnlineExamDB;

status VARCHAR(20),

FOREIGN KEY(student_id) REFERENCES Student(student_id),

FOREIGN KEY(exam_id) REFERENCES Exam(exam_id)

);
```

→INSERT INTO

```
INSERT INTO Student(name,email,password) VALUES

('Priya','priya@gmail.com','12345'),

('Amit','amit@gmail.com','12345');


INSERT INTO Exam(exam_name,total_marks,duration) VALUES

('SQL Basic Test',50,30),

('Python Basic Test',50,30);


INSERT INTO Question(exam_id,question_text,optionA,optionB,optionC,optionD,correct_option)
VALUES

(1,'SQL keyword to retrieve data?','SELECT','CREATE','INSERT','DELETE','A'),

(1,'Which SQL keyword is used to remove table?','DROP','ALTER','UPDATE','TRUNCATE','A'),

(2,'Which keyword defines a function in Python?','for','if','def','func','C');
```

→UPDATE

```
UPDATE Question SET optionD = 'TRUNCATE' WHERE question_id = 2;
```

→DELETE

```
DELETE FROM Student WHERE student_id = 2;
```

→WHERE / LIKE

```
SELECT * FROM Question WHERE question_text LIKE '%SQL%';
```

→GROUP BY + Aggregate functions

```
SELECT exam_id, COUNT(result_id) AS total_attempts, AVG(score) AS avg_score
FROM ExamResult
GROUP BY exam_id;


SELECT name FROM Student
WHERE student_id IN (
SELECT student_id FROM ExamResult WHERE status='Passed'
);
```

→Stored Procedure

```
DELIMITER //
CREATE PROCEDURE AddStudent(
IN sname VARCHAR(100),
IN semail VARCHAR(100),
IN spass VARCHAR(50)
)
BEGIN
INSERT INTO Student(name,email,password)
VALUES (sname,semail,spass);
END //
DELIMITER ;
```

→Trigger

```
DELIMITER //
CREATE TRIGGER set_status_after_insert
BEFORE INSERT ON ExamResult
FOR EACH ROW
BEGIN
```

```
IF NEW.score >= 25 THEN
SET NEW.status = 'Passed';
ELSE
SET NEW.status = 'Failed';
END IF;
END //
DELIMITER ;

INSERT INTO ExamResult(student_id,exam_id,score,exam_date)
VALUES(1,1,38,CURDATE());
```

Conclusion

*This project demonstrates how an Online Examination System can be implemented using SQL.

* It manages students, exams, questions, and results in a structured way.

* The use of procedures and triggers helps in automating routine tasks like student registration and result status updates.

*The design can be easily extended further to support multiple subjects and detailed analytics.