

Normalization Document for the Online Examination Management System

1. Introduction

Database for Online Examination Management System has been designed in a way that can handle students, faculty, exams, questions, answers, and respective entities in an efficient way.

Normalization is a method of trying to reduce redundancy, maintain data integrity, and enhance database performance. The current paper addresses the normalization process performed in order to achieve 1NF, 2NF, 3NF, and BCNF utilizing the reduced schema.

2. Unnormalized Form (UNF)

Entities and Functional Dependencies:

1. **Student:**

- Student_ID → {Reg_No, Name, Email, Password, Contact, College_ID}

2. **Faculty:**

- Faculty_ID → {Name, Email, Password, Contact, College_ID}

3. **College:**

- College_ID → {Name, Address}

4. **Venue:**

- Venue_ID → {Name, Capacity}

5. **Exam:**

- Exam_ID → {Title, Duration, Subject, Description, Total_marks, Start_time, End_time, Created_by (Faculty_ID), Venue_ID}

6. **Question:**

- Question_ID → {Question_text, Question_type, Correct_answer, Option_A, Option_B, Option_C, Option_D, Marks, Exam_ID}

7. **Answer:**

- Answer_ID → {Question_ID, Attempt_ID, Answer_text, Is_correct, Obtained_marks}

8. **Exam_Attempt:**

- Attempt_ID → {Student_ID, Exam_ID, Start_time, End_time, Status, Final_score}

9. **Exam_Availability:**

- Availability_ID → {Exam_ID, Available_from, Available_until, Student_Group}

10. **System_Logs:**

- Log_ID → {User_ID, User_Type, Activity, IP_Address, Created_At}

11. **Admin:**

- Admin_ID → {Name, Email, Password, Contact}

3. First Normal Form (1NF)

Transformation Actions:

- Ensure atomicity by decomposing repeating groups into individual tables.
- Assign primary keys to uniquely distinguish records.

Normalized Tables:

1. **Student** (Student_ID, Reg_No, Name, Email, Password, Contact, College_ID)
2. **Faculty** (Faculty_ID, Name, Email, Password, Contact, College_ID)
3. **College** (College_ID, Name, Address)
4. **Venue** (Venue_ID, Name, Capacity)
5. **Exam** (Exam_ID, Title, Duration, Subject, Description, Total_marks, Start_time, End_time, Created_by, Venue_ID)
6. **Question** (Question_ID, Question_text, Question_type, Correct_answer, Option_A, Option_B, Option_C, Option_D, Marks, Exam_ID)
7. **Answer** (Answer_ID, Question_ID, Attempt_ID, Answer_text, Is_correct, Obtained_marks)
8. **Exam_Attempt** (Attempt_ID, Student_ID, Exam_ID, Start_time, End_time, Status, Final_score)
9. **Exam_Availability** (Availability_ID, Exam_ID, Available_from, Available_until, Student_Group)
10. **System_Logs** (Log_ID, User_ID, User_Type, Activity, IP_Address, Created_At)
11. **Admin** (Admin_ID, Name, Email, Password, Contact)

4. Second Normal Form (2NF)

Transformation Actions:

- Remove partial dependencies by making all non-key attributes dependent on the entire primary key.

Changes:

- As **Answer** has a composite key (Answer_ID, Question_ID, Attempt_ID), relocate dependent attributes to a new table.
- Ensure relationships between **Exam** and **Venue** are properly defined.

Updated Tables:

1. **Answer** (Answer_ID, Attempt_ID, Answer_text, Is_correct, Obtained_marks)
2. **Question_Answer** (Question_ID, Answer_ID) (New table to establish relationship Questions and Answers)

5. Third Normal Form (3NF)

Transformation Actions:

- Eliminate transitive dependencies by having non-key attributes dependent on the primary key alone.

Changes:

- Separate email from Student, Faculty, and Admin to prevent transitive dependency.

Updated Tables:

1. **Student_Email** (Student_ID, Email)
2. **Faculty_Email** (Faculty_ID, Email)
3. **Admin_Email** (Admin_ID, Email)

6. Boyce-Codd Normal Form (BCNF)

Transformation Actions:

- Ensure each determinant is a candidate key.

Changes:

- If **Reg_No** is a unique identifier for students, we have a distinct table for registration numbers.

Updated Tables:

1. **Student_Reg** (Reg_No, Student_ID)

7. Conclusion

The updated normalization process guarantees that:

1. The database schema is optimized without redundancy.
2. Functional dependencies are properly translated to the schema.
3. Data anomalies (insertion, update, deletion) are minimized.

This schema offers efficient storage and ensures data integrity while permitting easy retrieval and updates.