Normalization and Functional Dependencies

Competitive Exam Preparation Progress Management System

Preet Dave

Introduction

In database systems, **Normalization** is a technique for designing data structures to minimize redundancy and dependency by dividing large tables into smaller, well-structured tables. **Functional Dependencies (FDs)** help in identifying relationships among attributes for normalization.

This document details the FDs in the *Exam Management System* and explains how normalization was applied to achieve a robust, well-structured relational schema.

Functional Dependencies (FDs)

The following functional dependencies were identified:

Table	**Functional Dependencies (FDs)**
Student	student_id → name, email, phone
Exam	exam_id → exam_name, exam_type, exam_date, application_deadline, syllabus, total_marks, duration_minutes
GovtExam	exam_id → conducting_body, vacancy_count
PrivateExam	exam_id → company_name, job_profile
Registration	registration_id → status
Registers	(student_id, exam_id) → registration_id
MockTest	test_id → exam_id, total_marks, test_date
Attempts_MockTest	(student_id, test_id) → score, percentage
StudentPreferred	(student_id, exam_type) *(multi-valued)*
PracticeSession	session_id → topic_name, attempt_date, num_questions, correct_answers
Attempts_PracticeSession	(student_id, session_id) *(composite key)*
Notification	notification_id → message, notify_date
Receives	(student_id, notification_id) *(composite key)*

Table 1: Identified Functional Dependencies

Normalization Process

Unnormalized Form (UNF)

Initially, data would exist in a single table with repeating groups:

- Multiple exams a student can register for
- Multiple preferences per student
- Multiple practice sessions attempted

First Normal Form (1NF)

Action: - Removed repeating groups. - Ensured atomic values in each attribute. - Separated multi-valued attributes into new tables:

- Created **StudentPreferred** from student's preferred exam types.
- Created **Attempts_PracticeSession** for practice session attempts.
- Created **Registers** for student exam registrations.

Second Normal Form (2NF)

Action: - Removed partial dependencies. - For many-to-many relationships, used composite primary keys:

- **Registers (student_id, exam_id)**
- **Attempts_MockTest (student_id, test_id)**
- **Receives (student_id, notification_id)**

- Decomposed tables where non-key attributes depended on part of composite key.

Third Normal Form (3NF)

Action: - Removed transitive dependencies:

- Status moved to separate **Registration** table linked via registration_id.
- Exam-type specific attributes stored in **GovtExam** and **PrivateExam**, linked via exam_id.

Boyce-Codd Normal Form (BCNF)

Action: - Verified that every functional dependency has a super key as determinant. - All relations already satisfied BCNF after 3NF adjustments.

Summary of Tables After Normalization

Normalized Table	Derived From
Student	Original UNF table
Exam	Original UNF table
GovtExam	Specialization of Exam
PrivateExam	Specialization of Exam
Registration	Separated from Registers for status dependency
Registers	Derived from student-exam relationship
MockTest	Separated from Exam for multiple mocks per exam
Attempts_MockTest	Separated from student-mock attempt relationship
StudentPreferred	Converted from multi-valued attribute in UNF
PracticeSession	Separated atomic session details
Attempts_PracticeSession	Separated many-to-many relationship
Notification	From original UNF notification data
Receives	For student-notification relationship

Table 2: Tables Created Through Normalization

Conclusion

This document systematically identified functional dependencies and normalized the Exam Management System database up to BCNF. All redundant, multi-valued, and transitive dependencies were eliminated. The resulting design ensures high data integrity, consistency, and improved query performance.