

Practical No.: -3

Aim:-Implement Schema and Tables with Naming Standards and Normalization

High:

Implement a normalized schema for a hospital management system with tables for patients, doctors, appointments, prescriptions, and billing. Follow strict naming conventions and include appropriate indexes.

Moderate:

Implement a database schema for a university with tables for students, faculty, and classes. Apply at least 2NF normalization and use consistent naming conventions.

Poor:

Create tables for a small bookstore with basic fields for books and authors. Focus on correct naming conventions and primary keys.

Process:

Design a normalized schema with proper naming conventions and indexing for a hospital.

1. Understand the Scenario:

- Analyze a hospital's operations: patient records, doctor schedules, appointments, prescriptions, and billing.

2. Identify Key Entities:

- Patients, Doctors, Appointments, Prescriptions, Bills

3. Define Attributes for Each Entity:

- Example: Patient (ID, Name, DOB, Gender), Doctor (ID, Name, Specialization)

4. Apply Naming Standards:

- Use prefixes like tbl_ for tables and col_ for columns. Example: tbl_Patient, col_PatientName

5. Establish Relationships:

- Link Patients ↔ Appointments ↔ Doctors ↔ Prescriptions ↔ Bills using foreign keys.

6. Normalize Tables:

- Apply 1NF, 2NF, and 3NF:
 - 1NF: Atomic values

- 2NF: No partial dependency (for composite keys)
- 3NF: No transitive dependency

7. Draw ER Diagram:

- Use tools like dbdiagram.io or paper to visualize relationships.

8. Design Indexes:

- Create indexes on frequently searched fields (e.g., col_DoctorSpecialization, col_AppointmentDate).

9. Prepare SQL Implementation Plan:

- List the sequence of table creation considering dependencies.

10. Verify Schema:

- Check normalization, keys, and naming compliance.

Moderate Level Task: University Database

Create a well-structured schema for students, faculty, and classes, normalized up to 2NF.

1. Understand the Use Case:

- Basic academic database for managing student enrollments and faculty assignments.

2. List Entities:

- Students, Faculty, Classes

3. List Basic Fields:

- Student (ID, Name, Branch), Faculty (ID, Name, Department), Class (ClassID, Subject, FacultyID)

4. Use Naming Conventions:

- Follow lowercase or underscore format (e.g., student_id, faculty_name)

5. Normalize Tables (Up to 2NF):

- 1NF: Ensure single values per field.
- 2NF: Remove partial dependencies (e.g., separate class from student if many students in one class).

6. Identify Primary and Foreign Keys:

- Example: class.faculty_id → faculty.id

7. Document Relationships:

- Use a basic ER diagram to show foreign key links.

8. Plan Table Creation:

- Create faculty first, then class, then student.

9. Ensure Constraints:

- Apply NOT NULL, UNIQUE where required.

10. Final Validation:

- Review table structure and relationships.

Poor Level Task: Bookstore Database

Goal: Simple schema with books and authors, with focus on correct naming and primary keys.

Step-by-Step Guide:

1. Define Two Entities:

- Books and Authors

2. List Basic Fields:

- Books (BookID, Title, Price, AuthorID)
- Authors (AuthorID, Name)

3. Use Clean Naming Standards:

- Example: book_id, author_name

4. Set Primary Keys:

- Make book_id and author_id primary in their respective tables.

5. Add Basic Foreign Key:

- books.author_id references authors.author_id

6. Keep Data Atomic:

- Don't combine multiple values in one field (e.g., no "Author1 & Author2" in one cell).

7. **Avoid Redundancy:**

- Store author info only once in the authors table.

8. **Design ERD (Optional):**

- Just two boxes with a line from Books to Authors.

9. **Plan Execution Order:**

- Create authors first, then books.

10. **Check Output Structure:**

- Ensure both tables exist and are linked by foreign key.

Level	Scenario	Expected Output
High	Hospital Management System	Minimum 5 normalized tables: → tbl_Patient, tbl_Doctor, tbl_Appointment, tbl_Prescription, tbl_Billing - Each table in 3NF - Clear primary keys and foreign keys - Consistent naming standards (tbl_ and col_) - Indexes on key searchable fields (e.g., col_AppointmentDate) - ER Diagram showing all relationships and cardinalities
Moderate	University Database	- 3 main tables: student, faculty, class - Tables normalized up to 2NF - Primary and foreign keys properly defined - Simple ER diagram linking student ↔ class ↔ faculty - Consistent column names (e.g., student_id, class_id) - Constraints such as NOT NULL , UNIQUE where needed
Poor	Bookstore Database	- 2 basic tables: books, authors - Primary keys: book_id, author_id - Foreign key: books.author_id → authors.author_id - 1NF compliance (atomic fields) - Correct naming and basic field types - Simple reference integrity between books and authors

Viva Questions:

- What are common naming conventions?
- What is a surrogate key?
- What happens when you skip normalization?