

Normalization Justification Document

Project: Appointment Scheduling System
Student: Shema Christian
Course: INSY 8311 – Database Development with PL/SQL
Date: 16 May 2025

1. Introduction

Relational database normalization is a systematic process to organize tables and their relationships to minimize redundancy and avoid undesirable characteristics like update anomalies. For this project’s logical model, we ensure our schema meets Third Normal Form (3NF), thereby improving data integrity, consistency, and maintenance.

2. Tables & Attributes

Table	Primary Key	Attributes
USER	user_id	name, email, role
SERVICE	service_id	service_name, duration_minutes, price
APPOINTMENT	appointment_id	appointment_date, appointment_time, status, patient_id (FK), provider_id (FK), service_id (FK)
REMINDER	reminder_id	reminder_date, appointment_id (FK)
AVAILABILITY	avail_id	provider_id (FK), day_of_week, start_time, end_time

3. First Normal Form (1NF)

Rule:

- Tables must have a primary key and atomic (indivisible) attribute values.

Validation:

- Every table defines a single-column primary key.
- All columns store a single value per row (e.g., `USER.email` is one email address, not a list).

4. Second Normal Form (2NF)

Rule:

- Non-key attributes must depend on the entire primary key (no partial dependencies).

Validation:

- In each table, every non-key attribute is fully functionally dependent on its table's primary key:
 - In **APPOINTMENT**, attributes like `appointment_date`, `status`, `patient_id`, etc., all describe that specific appointment (identified by `appointment_id`).
 - There are no composite primary keys, so partial dependency is inherently avoided.

5. Third Normal Form (3NF)

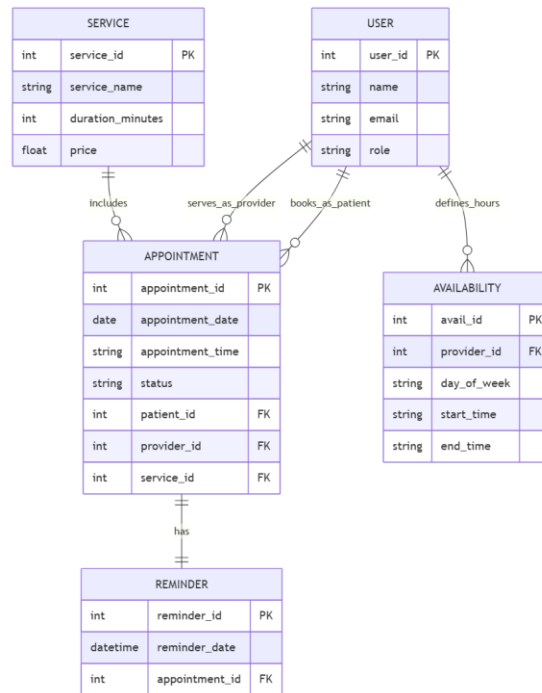
Rule:

- No transitive dependencies: non-key attributes must depend only on the primary key, not on other non-key attributes.

Validation:

- Service details (`service_name`, `price`, `duration_minutes`) reside in **SERVICE**, not in **APPOINTMENT**, preventing duplication of service information across appointment records.
 - **REMINDER** only stores `reminder_date` and the related `appointment_id`, eliminating redundant date data in **APPOINTMENT**.
 - **AVAILABILITY** isolates provider schedule info, so provider working hours aren't repeated in each appointment record.
-

6. ER Diagram



7. Conclusion

By adhering to 3NF, our database design:

- **Eliminates Redundancy:** Each fact is stored once.
- **Ensures Data Integrity:** Updates happen in a single table, reducing anomalies.
- **Enhances Scalability:** New services or users can be added without restructuring existing tables.

This normalized schema lays a robust foundation for implementing reliable PL/SQL procedures, triggers, and audit mechanisms in later phases.