Step 0: Unnormalized Form (UNF)

At this stage, all data is stored in a single "flat file" or table. This format contains **non-atomic values** (multiple pieces of information like Name, Email, and Role in a single cell) and **repeating groups** (multiple currency exchange rates in one field).

Booking_ID	User_Details (Name, Email, Role)	Hotel_Info (City, Capacity, Rates)	Room_Details	Dates	Total_Price	Currency_List
101	Riya; riya@email.com; User	London; 500; £200/£150	Deluxe (Max 4)	Jan 1-5	£800	USD: 1.2, EUR: 1.1

Step 1: First Normal Form (1NF)

Objective: Achieve Atomicity and eliminate repeating groups.
Implementation:

- Each cell now contains a single, discrete value.
- "User_Details" has been split into Username, Email, and Role.
- A Primary Key (Booking_ID) is identified to ensure each row is unique.

Booking_ID (PK)	Username	Email	Role	Hotel_City	Peak_Rate	Room_Type	Max_Occ	Check_In	Total_Price
101	Riya	riya@email.com	User	London	200	Deluxe	4	2026-01-01	800

Step 2: Second Normal Form (2NF)

Objective: Eliminate Partial Functional Dependencies.
Implementation:

- The system is decomposed into three distinct entities: **Users**, **Hotels**, and **Bookings**.
- Attributes now depend on the *entire* Primary Key of their respective tables. For example, a user's password_hash is now stored only once in the Users table, rather than being repeated for every booking they make.

Table: Users

user_id (PK)	username	email	password_hash	role

Table:Hotels

hotel_id (PK)	city	total_capacity	peak_rate	off_peak_rate

Table:Bookings

booking_id (PK)	user_id (FK)	hotel_id (FK)	room_type_name	max_occupancy	check_in_date	total_price
101	5	1	Standard	2	2026-02-01	150.00
102	8	1	Deluxe	4	2026-02-05	300.00
103	12	2	Deluxe	4	2026-02-10	280.00

Step 3: Third Normal Form (3NF)

Objective: Eliminate Transitive Dependencies.
Implementation:

- In the 2NF Bookings table, max_occupancy depended on the room_type_name, which in turn depended on the booking_id. This created a transitive dependency.
- To reach 3NF, the **Room_types** table was created. Now, max_occupancy depends solely on the type_id.
- The Bookings table now uses a Foreign Key (room_type_id) to reference these details.

Table: Users

user_id (PK)	username	email	password_hash	role

Table: Hotels

hotel_id (PK)	city	total_capacity	peak_rate	off_peak_rate

Table: Room_types

type_id (PK)	type_name	base_occupancy	max_occupancy
1	Standard	1	2
2	Deluxe	2	4
3	Suite	2	6

Table: Bookings

booking_id (PK)	user_id (FK)	hotel_id (FK)	room_type_id (FK)	check_in_date	total_price	status
101	5	1	1	2026-02-01	150.00	Confirmed
102	8	1	2	2026-02-05	300.00	Confirmed
103	12	2	2	2026-02-10	280.00	Paid