

Analysis of Database Schema

This document outlines the key constraints and normalization principles applied to the airline reservation system's database schema.

1. Key Constraints

Key constraints are rules that ensure data integrity within the database tables.

Primary Keys (PK)

A Primary Key is a unique identifier for each record in a table.

- **Passengers:** PassengerID
- **Airlines:** AirlineID
- **Airports:** AirportCode
- **Flights:** FlightID
- **Bookings:** BookingID
- **Seats:** SeatNumber

Foreign Keys (FK)

A Foreign Key is a key used to link two tables together. It is a field (or collection of fields) in one table that refers to the Primary Key in another table.

- **Flights Table:**
 - AirlineID references Airlines (AirlineID)
 - DepartureAirportCode references Airports (AirportCode)
 - ArrivalAirportCode references Airports (AirportCode)
- **Bookings Table:**
 - PassengerID references Passengers (PassengerID)
 - FlightID references Flights (FlightID)
- **Seats Table:**
 - FlightID references Flights (FlightID)
 - BookingID references Bookings (BookingID)

Other Constraints

- **NOT NULL:** Ensures that a column cannot have a NULL value.
 - **Passengers:** Name, Email
 - **Airlines:** Name
 - **Airports:** Name, City, Country
 - **Flights:** DepartureTime, ArrivalTime, Price
 - **Bookings:** BookingDate

2. Application of Normalization

Normalization is the process of organizing columns and tables in a relational database to minimize data redundancy.

First Normal Form (1NF)

All tables in the schema adhere to 1NF. – Each table has a primary key. – All column values are atomic (e.g., there are no lists or sets stored in a single cell). – There are no repeating groups of columns.

Second Normal Form (2NF)

The schema adheres to 2NF. This form is primarily concerned with tables that have composite primary keys. Since all tables have single-column primary keys, they automatically satisfy 2NF if they are in 1NF. All non-key attributes in each table are dependent on the table's primary key.

Third Normal Form (3NF)

The schema is designed in 3NF, which aims to eliminate transitive dependencies (where a non-key attribute depends on another non-key attribute).

Examples of 3NF application:

1. **Flights and Airlines:** Instead of storing the `AirlineName` in the `Flights` table, the `Flights` table contains an `AirlineID` (a foreign key). The `AirlineName` is stored in the `Airlines` table. This avoids redundancy and potential update anomalies. If an airline changed its name, you would only need to update it in one place (the `Airlines` table) instead of in every single flight record for that airline.
2. **Flights and Airports:** Similarly, details about the departure and arrival airports (like `Name`, `City`, `Country`) are not stored in the `Flights` table. Instead, the `Flights` table holds `DepartureAirportCode` and `ArrivalAirportCode` as foreign keys that reference the `Airports` table. This prevents storing the same airport information repeatedly for every flight that uses that airport.

By separating entities like `Airlines`, `Airports`, `Passengers`, and `Flights` into their own tables, we effectively minimize data redundancy and improve data integrity, which is the core goal of normalization.