

Lab Work 1 – ER Diagram Report

Introduction

The purpose of this lab work is to design and document an Entity-Relationship (ER) diagram for an Airport Management System. The system manages flights, bookings, passengers, baggage, airlines, and related operational details. The goal is to model the system in a way that ensures data integrity, supports normalization to 3rd Normal Form (3NF), and provides a clear structure for database implementation.

Entities and Attributes

The system consists of the following main entities, each with its attributes:

Airport: Stores airport details such as airport_id (PK), name, country, state, city, created_at, updated_at.

Airline: Stores airline details: airline_id (PK), airline_code, name, country, created_at, updated_at.

Flight: Represents flights with flight_id (PK), airline_id (FK), departure_airport_id (FK), arrival_airport_id (FK), scheduled/actual departure and arrival times, gates, created_at, updated_at.

Passenger: Passenger information with passenger_id (PK), first_name, last_name, gender, date_of_birth, passport_number, citizenship_country, residence_country, created_at, updated_at.

Booking: Bookings of flights: booking_id (PK), passenger_id (FK), flight_id (FK), status, booking_platform, ticket_price, created_at, updated_at.

BookingChange: Tracks modifications in bookings: change_id (PK), booking_id (FK), change_type, old_value, new_value, created_at, updated_at.

Baggage: Registered baggage: baggage_id (PK), booking_id (FK), weight_kg, created_at, updated_at.

BaggageCheck: Baggage inspection details: check_id (PK), baggage_id (FK), passenger_id (FK), result, created_at, updated_at.

BoardingPass: Boarding passes issued: boarding_pass_id (PK), booking_id (FK), seat, boarding_time, created_at, updated_at.

SecurityCheck: Passenger security checks: security_check_id (PK), passenger_id (FK), result, created_at, updated_at.

Relationships

The entities are connected through relationships that ensure consistency and reflect real-world operations. Key relationships include:

- One airline can operate many flights (1:N).
- One airport can serve as departure/arrival for many flights (1:N).
- One flight can have many bookings (1:N).
- Each booking belongs to exactly one passenger (N:1).
- Each booking may have multiple changes (1:N).
- Each booking can include multiple baggage items (1:N).
- Each baggage undergoes exactly one baggage check (1:1).
- Each booking generates one boarding pass (1:1).
- Each passenger undergoes one security check (1:1).

Normalization (3NF)

The database schema is designed in compliance with the Third Normal Form (3NF):

- 1NF: Each table has atomic values; no repeating groups or arrays.
- 2NF: All non-key attributes fully depend on the whole primary key (no partial dependencies).
- 3NF: No transitive dependencies; non-key attributes depend only on the primary key.

For example, in the Flight entity, all attributes (such as gates, departure/arrival times) depend only on flight_id. Similarly, in Booking, attributes like status and ticket_price depend only on booking_id, not indirectly on passenger_id or flight_id. This eliminates redundancy and ensures data integrity.

Conclusion

The ER diagram for the Airport Management System captures all necessary entities and relationships for managing flights, passengers, bookings, and related operations. The schema adheres to 3NF normalization rules, ensuring efficiency, data consistency, and minimal redundancy. This provides a strong foundation for implementation in a relational database system.