

Database Laboratory Work 1: ERD Diagram

Introduction

In this work, I designed an ER diagram for the airport management system. First, I read the description and identified the main entities: Airport, Airline, Flight, Passenger, Booking, BoardingPass, Baggage, BaggageCheck, SecurityCheck, and BookingChange. After that, I listed the attributes for each entity and marked primary keys, foreign keys, and unique fields. Then I looked at the relationships between the entities. For example, one airline can have many flights, and one passenger can have many bookings. The many-to-many relationship between flights and passengers was solved with the Booking table. I also normalized the database to the Third Normal Form (3NF) to remove redundancy and make the design clear. Finally, I created the ER diagram in draw.io, connected the entities, and added cardinalities.

Entities and Attributes

Airport: airport_id (PK), airport_name, country, state, city, created, updated

Airline: airline_id (PK), airline_code (UQ), name, country, created, updated

Flight: flight_id (PK), airline_id (FK), departure_airport_id (FK), arrival_airport_id (FK), departing_gate, arriving_gate, scheduled_departure_time, scheduled_arrival_time, actual_departure_time, actual_arrival_time, created, updated

Passenger: passenger_id (PK), first_name, last_name, gender, date_of_birth, citizenship_country, residence_country, passport_number (UQ), created, updated

Booking: booking_id (PK), flight_id (FK), passenger_id (FK), status, booking_platform, ticket_price, created, updated

BookingChange: change_id (PK), booking_id (FK), change_details, created, updated

BoardingPass: boarding_pass_id (PK), booking_id (FK, UQ), seat, boarding_time, created, updated

Baggage: baggage_id (PK), booking_id (FK), weight_kg, created, updated

BaggageCheck: baggage_check_id (PK), baggage_id (FK), check_results, created, updated

SecurityCheck: security_check_id (PK), passenger_id (FK), check_results, created, updated

Conclusion

The ER diagram for the airport management system has been successfully created. All entities, attributes, and relationships were identified based on the system description. The design was normalized to the Third Normal Form (3NF), which removed redundancy and ensured data consistency. Relationships were represented with correct cardinalities, and primary and foreign keys were assigned to maintain referential integrity. The final diagram provides a clear structure for implementing the database in a relational database management system.