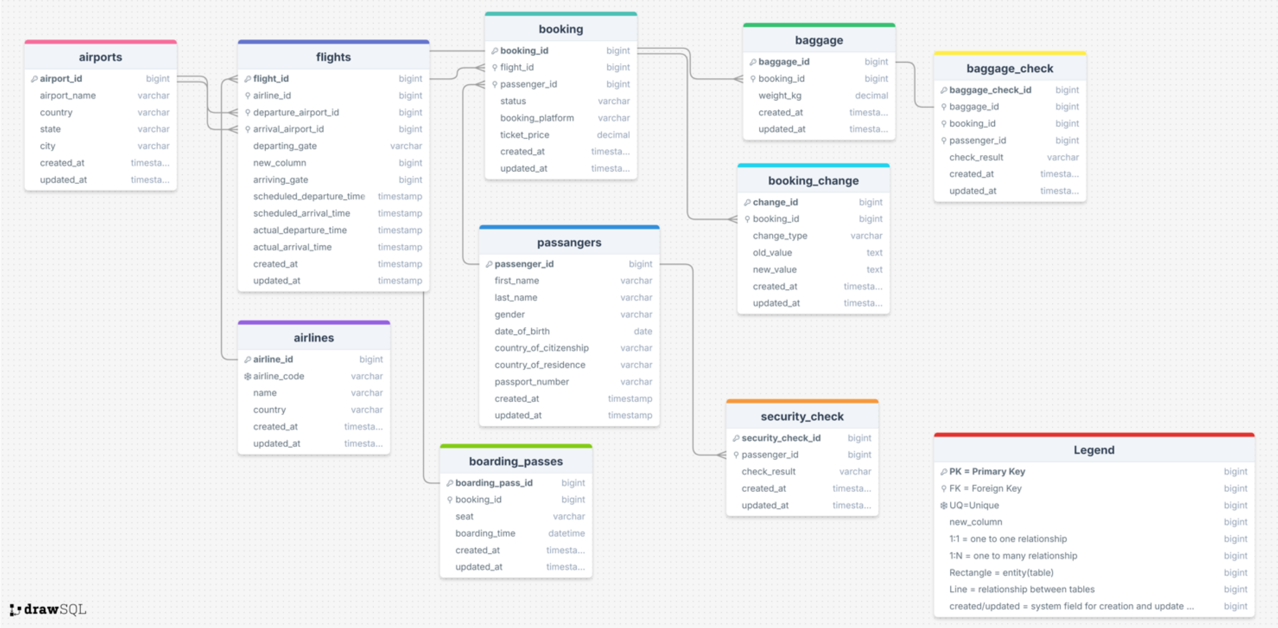

airports	
airport_id	bigint
airport_name	varchar
country	varchar
state	varchar
city	varchar
created_at	timesta...
updated_at	timesta...

flights	
flight_id	bigint
airline_id	bigint
departure_airport_id	bigint
arrival_airport_id	bigint
departing_gate	varchar
new_column	bigint
arriving_gate	bigint
scheduled_departure_time	timestamp
scheduled_arrival_time	timestamp
actual_departure_time	timestamp
actual_arrival_time	timestamp
created_at	timestamp
updated_at	timestamp

booking	
booking_id	bigint
flight_id	bigint
passenger_id	bigint
status	varchar
booking_platform	varchar
ticket_price	decimal
created_at	timesta...
updated_at	timesta...

baggage	
baggage_id	bigint
booking_id	bigint
weight_kg	decimal
created_at	timesta...
updated_at	timesta...

baggage_check	
baggage_check_id	bigint
baggage_id	bigint
booking_id	bigint
passenger_id	bigint
check_result	varchar
created_at	timesta...
updated_at	timesta...

booking_change	
change_id	bigint
booking_id	bigint
change_type	varchar
old_value	text
new_value	text
created_at	timesta...
updated_at	timesta...

passangers	
passenger_id	bigint
first_name	varchar
last_name	varchar
gender	varchar
date_of_birth	date
country_of_citizenship	varchar
country_of_residence	varchar
passport_number	varchar
created_at	timestamp
updated_at	timestamp

airlines	
airline_id	bigint
airline_code	varchar
name	varchar
country	varchar
created_at	timesta...
updated_at	timesta...

boarding_passes	
boarding_pass_id	bigint
booking_id	bigint
seat	varchar
boarding_time	datetime
created_at	timesta...
updated_at	timesta...

security_check	
security_check_id	bigint
passenger_id	bigint
check_result	varchar
created_at	timesta...
updated_at	timesta...

Legend	
PK = Primary Key	bigint
FK = Foreign Key	bigint
UQ=Unique	bigint
new_column	bigint
1:1 = one to one relationship	bigint
1:N = one to many relationship	bigint
Rectangle = entity(table)	bigint
Line = relationship between tables	bigint
created/updated = system field for creation and update ...	bigint

Database Laboratory work #1

Report: ERD for International Airport

Complited by Mussin Aliaskar 24B031890 Wed(13:00-14:00)

1. Introduction

The purpose of this work is to design an **Entity–Relationship Diagram (ERD)** for an International Airport Database. The system is intended to manage airport operations including flights, airlines, passenger bookings, boarding passes, baggage, and security checks. The ERD helps to visualize the structure of the database and the relationships between different entities, ensuring that the design is normalized and supports efficient data storage and retrieval.

2. Tasks Performed

Step 1. Identify Entities

From the system description, the following entities were identified:

- Airport
- Airline
- Flight
- Passenger
- Booking

- BookingChange

- BoardingPass

- Baggage

- Baggage Check

- SecurityCheck

Step 2. Determine Attributes

Each entity contains attributes. For example:

- *Passenger*: passenger_id, first_name, last_name, gender, date_of_birth, passport_number, country_of_citizenship, country_of_residence, created_at, updated_at.

- *Flight*: flight_id, airline_id, departure_airport_id, arrival_airport_id, scheduled_departure_time, scheduled_arrival_time, actual_departure_time, actual_arrival_time.

(Full list given in schema section).

Step 3. Define Keys

- Primary keys (PK) uniquely identify rows in each table.

- Foreign keys (FK) maintain relationships, e.g. booking.passenger_id → passenger.passenger_id.

- Unique constraints were defined where needed (e.g. airline_code).

Step 4. Normalization

The schema was checked up to **Third Normal Form (3NF)**:

- No repeating groups (e.g. baggage stored separately from bookings).

- No partial dependencies (composite keys avoided).

- No transitive dependencies (e.g. baggage checks depend only on baggage_id, not indirectly on booking).

Step 5. Identify Relationships

- Airline — Flight (1:M)

- Airport — Flight (1:M, for both departure and arrival)

- Flight — Booking (1:M)

- Passenger — Booking (1:M)

- Booking — BoardingPass (1:1)

- Booking — Baggage (1:M)

- Baggage — BaggageCheck (1:1)

- Passenger — SecurityCheck (1:M)

- Booking — BookingChange (1:M)

Step 6. Create ER Diagram

The diagram was implemented in drawSQL using SQL schema definitions. Boxes represent entities, attributes are listed inside, and lines show relationships with cardinality.

4. Results

As a result of the work:

- A structured set of entities and attributes for the Airport Database was created.

- Relationships and cardinalities were identified.

- The schema was normalized to 3NF.

- An ER-diagram was drawn in drawSQL to visualize the database design.

5. Conclusion

The completed ER-diagram provides a clear conceptual model of the International Airport Database. The design ensures data integrity, avoids redundancy, and reflects real-world airport operations. This model can be further used for creating a physical database in SQL and for implementing an airport management system.