1. **Identified Entities:**
   Airport,Airline,Flight,Passenger,Booking,Boarding_Pass,Baggage,Baggage_Check, Security_Check
2. **Attributes of entities:**
   Airport(airport_id (PK), airport_name, country, state, city, created_at, updated_at**)**
   Airline(airline_id (PK), airline_code (Unique), name, country, created_at, updated_at**)**
   Flight(flight_id (PK), airline_id (FK), departure_airport_id (FK), arrival_airport_id (FK), departing_gate, arriving_gate, scheduled_departure, scheduled_arrival, actual_departure, actual_arrival, created_at, updated_at**)**
   Passenger(passenger_id (PK), first_name, last_name, gender, date_of_birth, citizenship_country, residence_country, passport_number (Unique), created_at, updated_at**)**
   Booking(booking_id (PK), flight_id (FK), passenger_id (FK), status, booking_platform, ticket_price, created_at, updated_at)
   Boarding_Pass(boarding_pass_id (PK), booking_id (FK), seat, boarding_time, created_at, updated_at**)**
   Baggage(baggage_id (PK), booking_id (FK), weight_kg, created_at, updated_at)
   Baggage_Checking(checking_id (PK), baggage_id (FK), check_result, created_at, updated_at**)**
   Security_Check(security_check_id (PK), passenger_id (FK), check_result, created_at, updated_at**)**

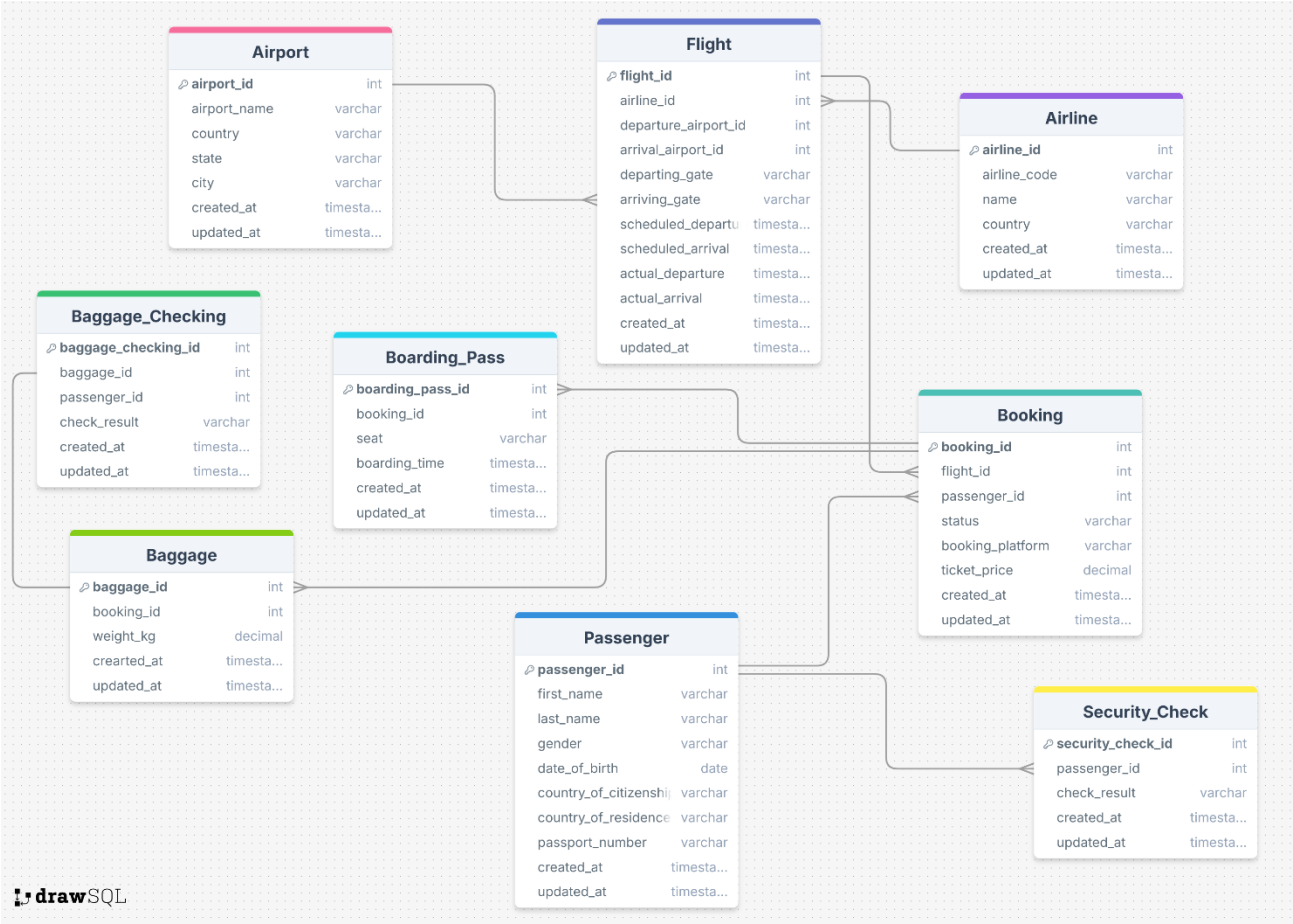
3. Normalization (3NF):

- Each table has a primary key.
- All non-key attributes depend solely on the PK.
- No transitive dependencies — separate BookingChange table to store changes.
- No repeating groups — one record per flight, booking, baggage, etc.

4. Relations:

Airport 1 — N Flight (departure/arrival): Flight must reference exactly one departure and arrival airport.
Airline 1 — N Flight: Each flight belongs to one airline.
Flight 1 — N Booking: Bookings cannot exist without a flight.
Passenger 1 — N Booking: Passenger may have many bookings.
Booking 1 — N Boarding_Pass: One booking can generate many boarding passes.
Booking 1 — N Baggage: One booking can have many baggage items.
Baggage 1 — 0..1 Baggage_Checking: Each baggage may have zero or one checking record.

Passenger 1 — N Security_Check: One passenger can have many security checks.



6. Legend

- Entity – shown as a box with the entity name
- PK – Primary Key (unique identifier for each record)
- FK – Foreign Key (connects to primary key of another table)
- Attributes – Characteristics or properties that describe an entity (e.g., name, date, price)
- Relationships – lines connecting entities, labeled with
    - 1:1 – one-to-one relationship
    - 1:N – one-to-many relationship
    - N:M – many-to-many relationship (using a linking table)

Brief Description:

- **Airport**: Stores details of all airports.
- **Airline**: Stores airline company data.

- **Flight**: Connects Airline and Airports, contains schedule/actual times.
- **Passenger**: Contains passenger profile info.
- **Booking**: Represents a passenger’s flight reservation.
- **Boarding_Pass**: Generated for each booking, includes seat info.
- **Baggage**: Stores each baggage registered under a booking.
- **Baggage_Checking**: Logs baggage inspection results.
- **Security_Check**: Stores security screening results per passenger.