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.

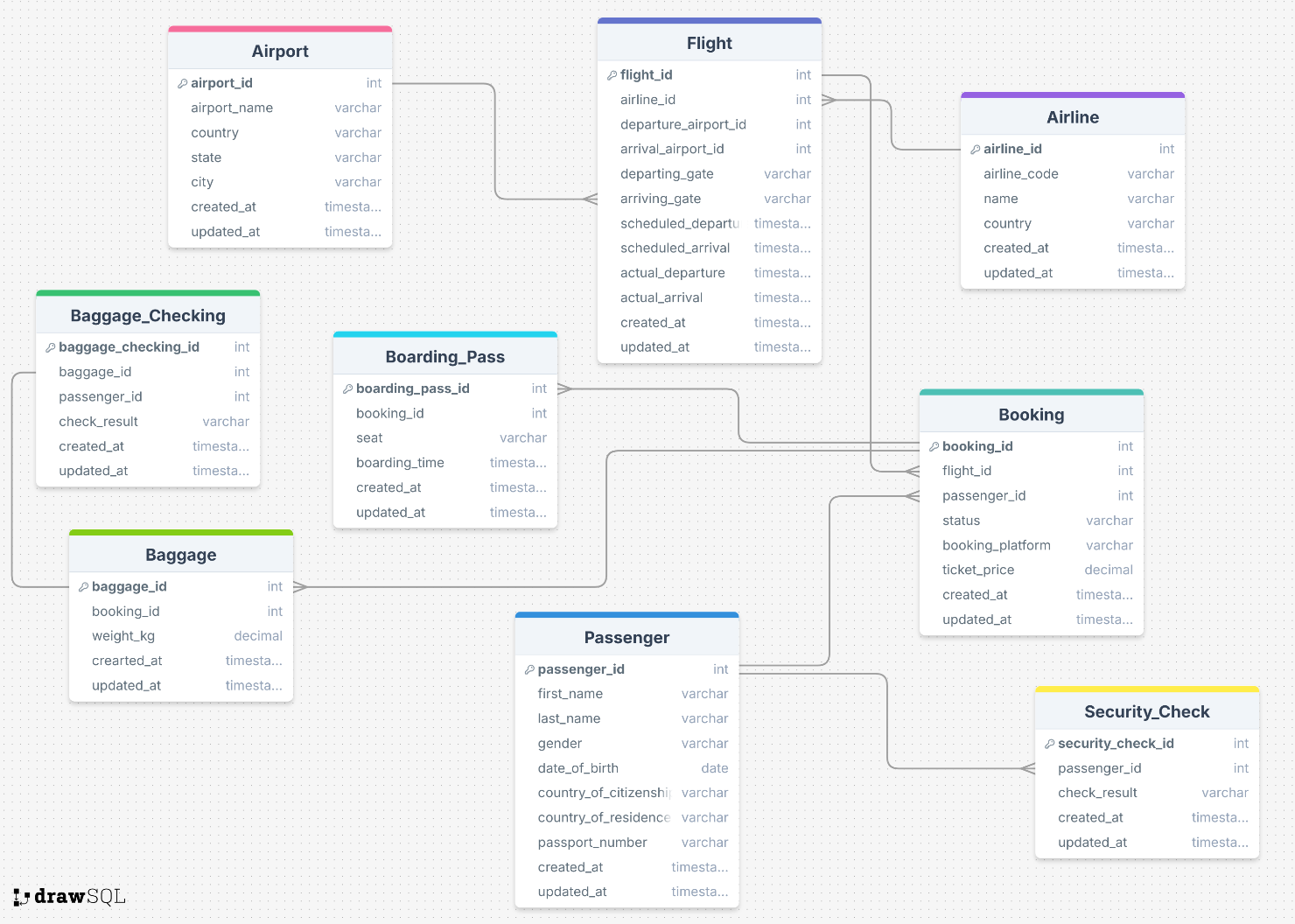
**Passenger 1 — N Booking**: Passenger may have many bookings.

**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.