International Airport Database System

Entities:

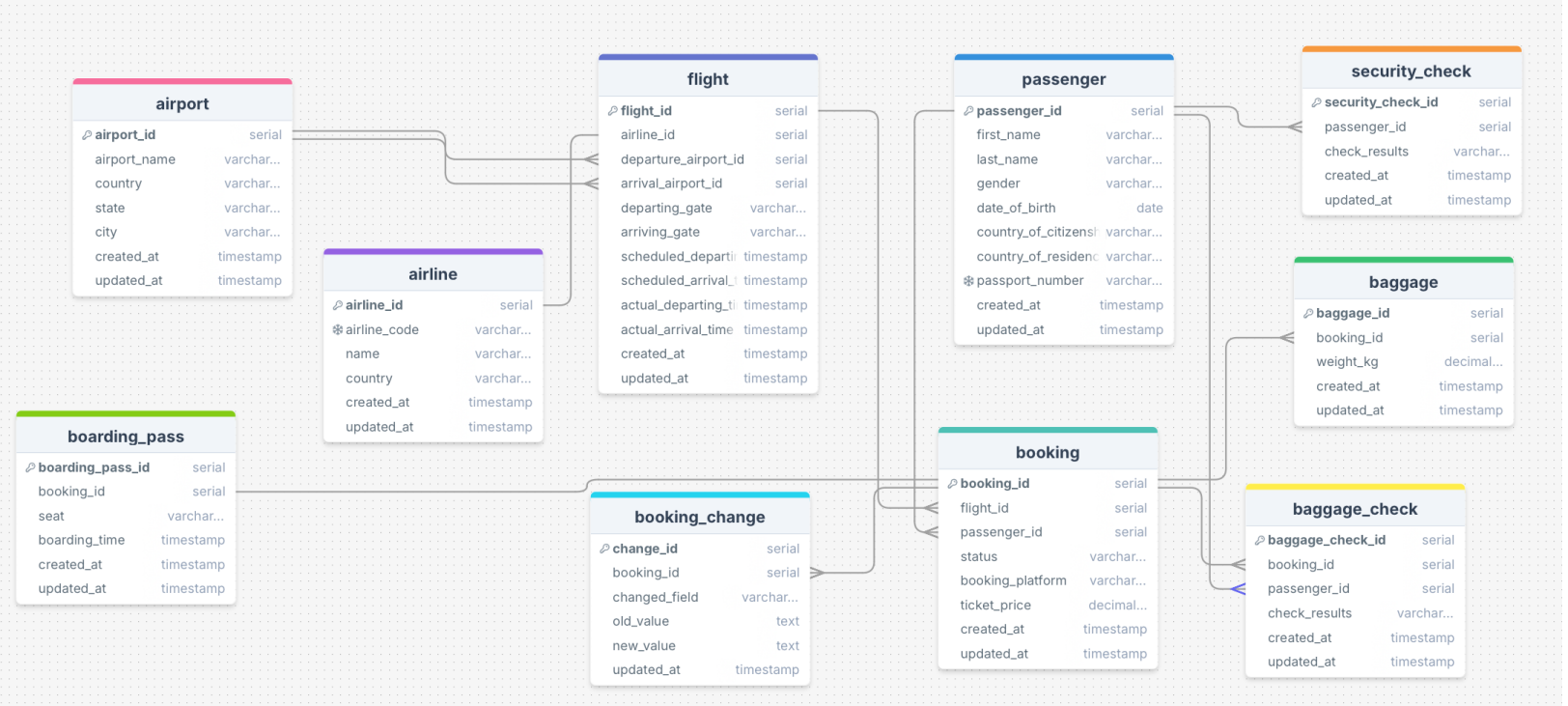
1. **airport** - Represents a physical airport location.
    I. airport_id (PK): unique ID for the airport.
    II. airport_name, country, state, city: Geographic details of the airport.
    III. created_at, updated_at: Timestamps for record tracking.
2. **airline** - Represents an airline company.
    I. airline_id (PK): Unique ID for the airline.
    II. airline_code: IATA code (e.g., AA, DL).
    III. name, country: Details of the airline.
    IV. created_at, updated_at: Timestamps.
3. **flight** - Represent a specific flight journey between two airports.
    I. flight_id (PK): Unique flight number.
    II. departing_gate, arriving_gate: Gate information.
    III. scheduled_*_time, scheduled_*_time: Planned and real timings.
    IV. *_airport_id (FK): Links to origin and destination airports.
    V. airline_id (FK): Links to the operating airline.
    VI. created_at, updated_at: Timestamps.
4. **passenger** - Represents a person who can make bookings and travel.
    I. passenger_id (PK): unique ID for the passenger.
    II. first_name, last_name, gender, date_of_birth: Personal details.
    III. country_of_citizenship, country_of_residence: Nationality and legal residence.
    IV. created_at, updated_at: Timestamps.
5. **booking** - Represents a passenger's reservation for a seat on a flight.
    I. booking_id (PK): Unique booking reference.
    II. status: Current state (e.g., Confirmed, Cancelled).
    III. ticket_price: The cost of the flight ticket.
    IV. flight_id (FK), passenger_id (FK): Links to the specific flight and passenger.
6. **boarding_pass** - Represents the issued pass that allows a passenger to board.
    I. boarding_pass_id (PK): unique ID for the pass.
    II. seat: The assigned seat number.
    III. boarding_time: The time boarding is scheduled.
    IV. booking_id (FK): Links to the one booking it fulfills.
    V. created_at, updated_at: Timestamps.

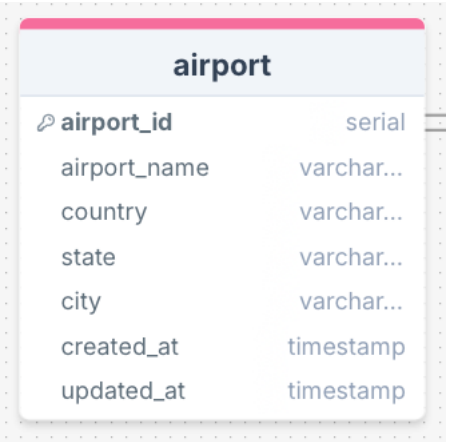
7. **baggage** - Represents a piece of luggage checked in for a booking.
    I. baggage_id (PK): Unique ID for the baggage.
    II. weight_kg: The weight of the bag.
    III. booking_id (FK): Links to the booking it belongs to.
    IV. created_at, updated_at: Timestamps.
8. **baggage_check** - Represents the security screening result for a passenger's baggage.
    I. baggage_check_id (PK): Unique ID for the baggage check event.
    II. check_results: Outcome of the check.
    III. booking_id (FK), passenger_id (FK): Links to the specific booking and passenger being screened.
    IV. created_at, updated_at: Timestamps.
9. **security_check** - Represents the security screening result for a passenger.
    I. security_check_id (PK): Unique ID for the check event.
    II. check_results: Outcome.
    III. passenger_id (FK): Links to the passenger screened.
    IV. created_at, updated_at: Timestamps.
10. **booking_change** – A booking change table that tracks all changes made to bookings.
    I. change_id (PK): Unique ID for the log entry.
    II. booking_id (FK): The booking that was changed.
    III. changed_field: the name of the attribute that was modified.
    IV. old_value, new_value: The data before and after the change.
    V. updated_at: When the change occurred.

Relationship	Cardinality	Description
Airline operates flights	One-to-many (1:M)	One airline can operate many flights. Each flight is operated by exactly one airline. A flight cannot exist without an airline.
Airport originates flights	One-to-many (1:M)	One airport can be origin for many flights. Each flight has exactly one origin airport. A flight must have a departure airport.
Airport receives flights	One-to-many (1:M)	One airport can be the destination for many flights. Each flight has

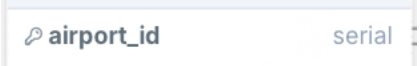
		exactly one destination airport. A flight must have an arrival airport.
Flight has bookings	One-to-many (1:M)	One flight can have many bookings. Each booking is for exactly one flight. A booking cannot exist without a flight.
Passenger makes bookings	One-to-many (1:M)	One passenger can make many bookings over time. Each booking is made by exactly one passenger. A booking must be linked to a passenger.
Booking generates boarding passes	One-to-one (1:1)	One booking generates one boarding_pass. Each boarding_pass is generated for exactly one booking. The relationship is optional because a boarding_pass is only created after check-in.
Booking has baggage	One-to-many (1:M)	One booking can have many pieces of baggage. Each piece of baggage is linked to exactly one booking. A bag cannot exist without a booking.
Booking undergoes baggage check	One-to-many (1:M)	One booking can undergo many baggage checks. Each baggage check is performed on the bags from one booking.
Passenger undergoes security check	One-to-many (1:M)	One passenger can undergo many security checks over time. Each security check is performed on one passenger.
Booking is altered in booking_change	One-to-many	One booking can have many entries in the booking_change log, one for each change made to

		it. Each booking_change entry documents a change to one booking.
Baggage_check performed on passenger	Many-to-one (M:1)	This is the other side of the “undergoes” relationship. It emphasizes that a baggage_check is also directly linked to the passenger who owns the baggage, in addition to the booking.



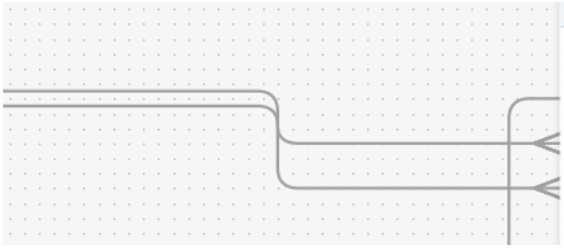


- Represents a table in the database.



- The key sign indicates the primary key (PK) – a unique identifier for each record in the table.

- These lines indicate foreign key (FK) – an attribute that creates a link to the primary key in another table.



- These lines indicate represent the cardinality between tables.