Flight
🔑 flight_id	int
departure_airport_id	int
airline_id	int
arrival_airport_id	int
departing_gate	timesta...
arriving gate	timesta...
scheduled_departu	timesta...
scheduled_arrival_t	timesta...
actual_arrival_time	timesta...
actual_departure_ti	timesta...
updated	timesta...
created	timesta...

Airport
🔑 airport_id	int
airport_name	varchar
country	varchar
state	varchar
city	varchar
created	timestamp
updated	timestamp

Baggage Check
🔑 baggage_checking__id	int
baggage_id	int
check_results	varchar
created	timesta...
updated	timesta...

Baggage
🔑 baggage_id	int
booking_id	int
weight	int
created	timesta...
updated	timesta...

Airlines
🔑 airline_id	int
airline_cod	int
name	varchar
country	varchar
created	timesta...
updated	timesta...

Boarding Passes
🔑 boarding_pass_id	int
booking_id	int
❄ seat	int
boarding_time	timesta...
created	timesta...
updated	timesta...

Booking
🔑 booking_id	int
flight_id	int
passenger_id	int
status	varchar
ticket_price	int
booking_platform	varchar
created	timesta...
updated	timesta...

SecurityCheck
🔑 security_check_id	int
passenger_id	int
check_results	varchar
created	timesta...
updated	timesta...

Passengers
🔑 passenger_id	int
first_name	varchar
last_name	varchar
gender	varchar
date_of_birth	date
country_of_citizenshi	varchar
country_of_residence	varchar
❄ passport_number	int
created	timesta...
updated	timesta...

Booking Changes
🔑 booking_changes_id	int
booking_id	int
changed_time	timesta...
changed_information	varchar

Documentation – International Airport Database

I created an ER-diagram for the International Airport system.
First, I identified the main entities: Airport, Airlines, Flight, Passenger, Booking, Booking Changes, Boarding Pass, Baggage, Baggage Check, Security Check.

For each entity I added attributes, including primary keys (PK), foreign keys (FK), unique fields (UQ). For example, passport number and airline code are unique.

I normalized the structure to Third Normal Form (3NF) to avoid redundancy.

I connected entities with relationships:

One airline has many flights.
One airport is linked to many flights (departure/arrival).
One flight has many bookings.
One passenger can have many bookings.
One booking can include many boarding passes and baggage.

Each baggage is linked to baggage checks.
Each passenger is linked to security checks.

Finally, I added system fields (created/updated) to track changes.
This ensures the database is clear, normalized, and supports airport operations.