Laboratory Work 1. ERD Diagram

Course: Databases
Topic: International Airport Management System

System Description

This database is designed to **manage key processes of an international airport**, including:

- flight bookings,
- flight management,
- airline information,
- baggage registration and checks,
- passenger security checks,
- and boarding pass management.

The database consists of **interconnected tables** that ensure **data integrity** and **minimize redundancy** by using **primary keys**, **foreign keys**, and normalization up to the **Third Normal Form (3NF)**.

Entity Descriptions

1. Airport

Stores information about airports:

- airport_id *(PK)* — unique airport identifier
- airport_name — name of the airport
- country — country
- state — state/region
- city — city
- created_at, updated_at — timestamps for record creation and update

Relationship:

- One airport can serve as a **departure or arrival airport** for many flights *(One-to-Many)*.

---

2. Airline

Stores details about airlines:

- airline_id *(PK)* — unique airline identifier
- airline_code *(UNI)* — unique airline code
- name — airline name
- country — country where the airline is registered
- created_at, updated_at — timestamps for record creation and update

Relationship:

- One airline operates **many flights** *(One-to-Many)*.

---

3. Flight

Represents planned and actual flight information:

- flight_id *(PK)* — unique flight identifier
- departure_airport_id, arrival_airport_id *(FK)* — departure and arrival airports
- airline_id *(FK)* — airline operating the flight
- departing_gate, arriving_gate — gates for departure and arrival
- scheduled_departure_time, scheduled_arrival_time — scheduled times
- actual_departure_time, actual_arrival_time — actual times
- created_at, updated_at — timestamps for record creation and update

Relationship:

- A flight belongs to **one airline** *(Many-to-One)*.
- A flight can have **many bookings** *(One-to-Many)*.

---

4. Passenger

Stores passenger personal data:

- passenger_id *(PK)* — unique passenger identifier
- first_name, last_name — first and last names
- gender — gender
- date_of_birth — date of birth
- country_of_citizenship — country of citizenship

- country_of_residence — country of residence
- passport_number *(UNI)* — unique passport number
- created_at, updated_at — timestamps for record creation and update

Relationship:

- One passenger can have **many bookings** *(One-to-Many)*.
- One passenger can undergo **many security checks** *(One-to-Many)*.

---

5. Booking

Contains information about flight bookings:

- booking_id *(PK)* — unique booking identifier
- flight_id *(FK)* — associated flight
- passenger_id *(FK)* — passenger who made the booking
- status — booking status
- booking_platform — platform used for booking (e.g., website, mobile app)
- ticket_price — price of the ticket
- created_at, updated_at — timestamps for record creation and update

Relationship:

- A booking is linked to **one flight** *(Many-to-One)*.
- A booking can have **many boarding passes and baggage records** *(One-to-Many)*.

---

6. BoardingPass

Stores information about issued boarding passes:

- boarding_pass_id *(PK)* — unique boarding pass identifier
- booking_id *(FK)* — associated booking
- seat — passenger seat number
- boarding_time — time of boarding
- created_at, updated_at — timestamps for record creation and update

Relationship:

- A booking can have **multiple boarding passes** *(One-to-Many)*.

- Each boarding pass belongs to **one booking** *(Many-to-One)*.

---

7. Baggage

Stores data about registered baggage:

- baggage_id *(PK)* — unique baggage identifier
- booking_id *(FK)* — booking associated with the baggage
- weight_kg — weight of baggage in kilograms
- created_at, updated_at — timestamps for record creation and update

Relationship:

- A booking can include **multiple pieces of baggage** *(One-to-Many)*.

---

8. Baggage Checking

Stores information about baggage inspections:

- baggage_checking_id *(PK)* — unique baggage check identifier
- baggage_id *(FK)* — baggage being checked
- passenger_id *(FK)* — passenger who owns the baggage
- check_results — results of the baggage inspection
- created_at, updated_at — timestamps for record creation and update

Relationship:

- One piece of baggage can undergo **multiple inspections** *(One-to-Many)*.

---

9. Security Check

Stores data about passenger security checks:

- security_check_id *(PK)* — unique security check identifier
- passenger_id *(FK)* — passenger being checked
- check_results — results of the security check
- created_at, updated_at — timestamps for record creation and update

Relationship:

- A passenger can undergo **multiple security checks** *(One-to-Many)*.

Relationships Between Tables

- **Airport — Flight:**
  One airport can be the departure or arrival point for many flights *(One-to-Many)*.
- **Airline — Flight:**
  One airline can operate many flights *(One-to-Many)*.
- **Flight — Booking:**
  One flight can have many bookings *(One-to-Many)*.
- **Passenger — Booking:**
  A passenger can have many bookings *(One-to-Many)*.
- **Booking — BoardingPass:**
  A booking can have multiple boarding passes *(One-to-Many)*.
- **Booking — Baggage:**
  A booking can include multiple pieces of baggage *(One-to-Many)*.
- **Baggage — Baggage Checking:**
  One piece of baggage can undergo multiple inspections *(One-to-Many)*.
- **Passenger — Security Check:**
  A passenger can have multiple security checks *(One-to-Many)*.

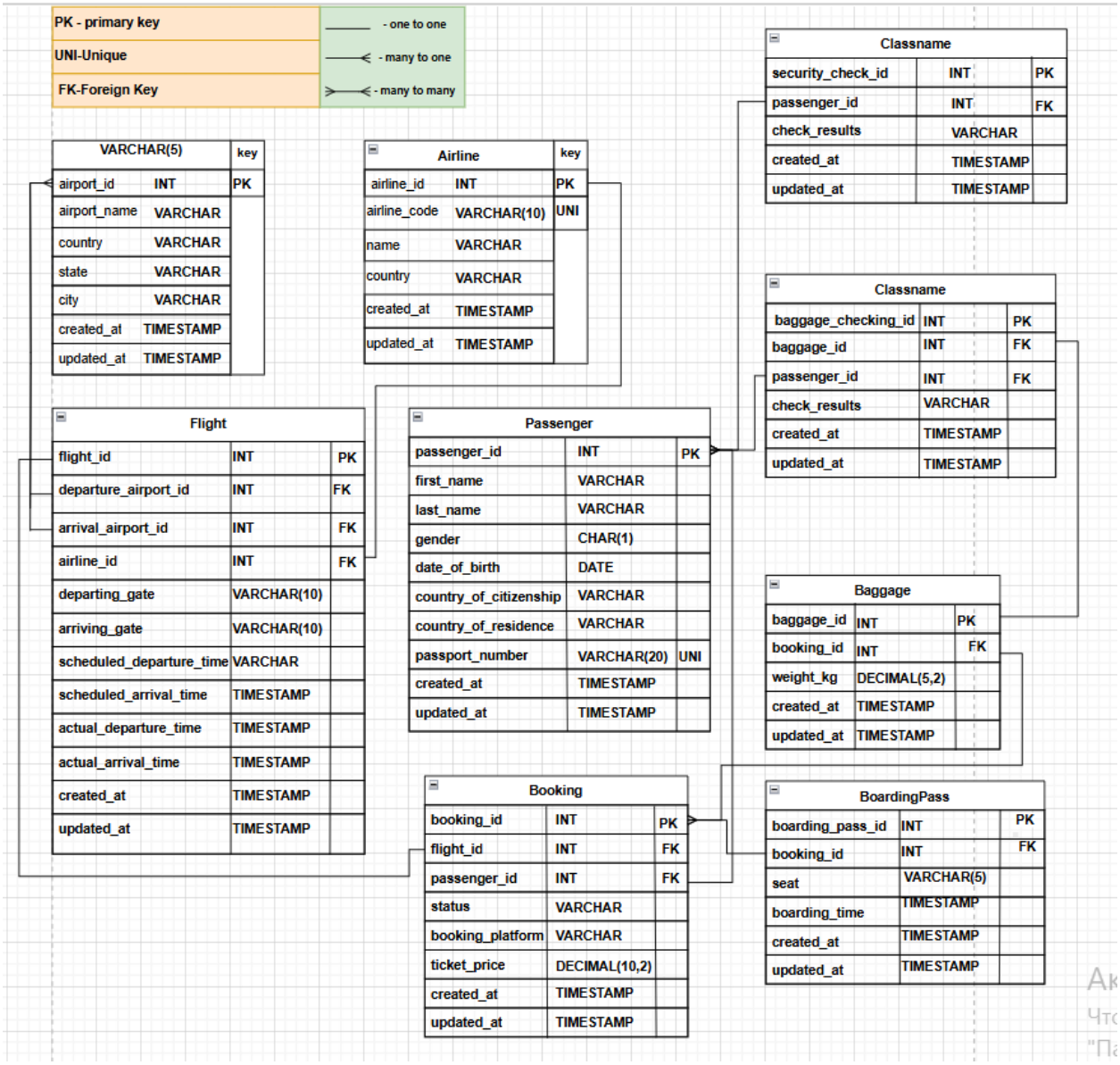
Conclusion

The proposed ERD design ensures:

- **Data integrity** through primary and foreign keys,
- **Minimal redundancy** thanks to normalization up to **3NF**,
- **Scalability**, allowing the system to handle flight bookings, baggage management, inspections, and security processes efficiently.

This structure provides a clear and organized way to manage all airport-related operations while keeping data accurate and consistent.

Downlowad LINK: 
practice 2
database.drawio


PK - primary key
UNI-Unique
FK-Foreign Key
- one to one
- many to one
- many to many
VARCHAR(5)
key
airport_id INT PK
airport_name VARCHAR
country VARCHAR
state VARCHAR
city VARCHAR
created_at TIMESTAMP
updated_at TIMESTAMP
Airline
key
airline_id INT PK
airline_code VARCHAR(10) UNI
name VARCHAR
country VARCHAR
created_at TIMESTAMP
updated_at TIMESTAMP
Classname
security_check_id INT PK
passenger_id INT FK
check_results VARCHAR
created_at TIMESTAMP
updated_at TIMESTAMP
Classname
baggage_checking_id INT PK
baggage_id INT FK
passenger_id INT FK
check_results VARCHAR
created_at TIMESTAMP
updated_at TIMESTAMP
Flight
flight_id INT PK
departure_airport_id INT FK
arrival_airport_id INT FK
airline_id INT FK
departing_gate VARCHAR(10)
arriving_gate VARCHAR(10)
scheduled_departure_time VARCHAR
scheduled_arrival_time TIMESTAMP
actual_departure_time TIMESTAMP
actual_arrival_time TIMESTAMP
created_at TIMESTAMP
updated_at TIMESTAMP
Passenger
passenger_id INT PK
first_name VARCHAR
last_name VARCHAR
gender CHAR(1)
date_of_birth DATE
country_of_citizenship VARCHAR
country_of_residence VARCHAR
passport_number VARCHAR(20) UNI
created_at TIMESTAMP
updated_at TIMESTAMP
Baggage
baggage_id INT PK
booking_id INT FK
weight_kg DECIMAL(5,2)
created_at TIMESTAMP
updated_at TIMESTAMP
Booking
booking_id INT PK
flight_id INT FK
passenger_id INT FK
status VARCHAR
booking_platform VARCHAR
ticket_price DECIMAL(10,2)
created_at TIMESTAMP
updated_at TIMESTAMP
BoardingPass
boarding_pass_id INT PK
booking_id INT FK
seat VARCHAR(5)
boarding_time TIMESTAMP
created_at TIMESTAMP
updated_at TIMESTAMP