Lab 1

1. The database is called Airport and it consists of ==9 entities==: International Airport, Airlines, Flights, Bookings, Boarding, Baggage, Checking, Passengers, Security.
2. ==Attributes of entities==:
   - **International Airport:** airport_id (serial, primary key) not null, airport_name (varchar)not null, country(varchar) not null, state(varchar) nullable, city(varchar) not null, created(timestamp), updated(timestamp).
   - **Airlines:** airline_id(serial, primary key)not null, airline_code(varchar)not null, name(varchar)not null, country(varchar)not null, created(timestamp), updated(timestamp).
   - **Flights:** flight_id(serial, primary key) not null, departing_gate(varchar) nullable, arriving_gate(varchar)nullable, created_at(timestamp), updated_at(timestamp), airline_id(int, foreign key) not null, departure_airport_id (int, foreign key) not null, arrival_airport_id (int, foreign key) not null, scheduled_departure_time (timestamp) not null, scheduled_arrival_time (timestamp) not null, actual_departure_time (timestamp) nullable, actual_arrival_time (timestamp) nullable.
   - **Bookings:** booking_id(serial, primary key)not null, flight_id(int, foreign key)not null, passenger_id(int, foreign key)not null, status(varchar)not null, booking_platform(varchar)nullable, booking_created(timestamp), booking_updated(timestamp), ticket_price(decimal)not null.
   - **Boarding:** pass_id(serial, primary key)not null, booking_id(int, foreign key) not null, seat(char) not null, boarding_time(timestamp) not null, pass_created(timestamp), pass_updated(timestamp)

- **Baggage:** baggage_id(serial, primary key) not null, booking_id(int, foreign key)not null, weight(decimal) not null, created_at(timestamp), updated_at(timestamp).
- **Checking:** checking_id(serial, primary key)not null, booking_id(int, foreign key) not null, passenger_id(int, foreign key)not null, check_results(varchar)not null, created_at(timestamp), updated_at(timestamp).
- **Passenger:** passenger_id(serial, primary key)not null, first_name(varchar)not null, last_name(varchar) not null, gender(varchar) not null, date_of_birth(date) not null, citizenship_country(varchar) not null, residence_country(varchar) nullable, passport_number(varchar) not null, profile_created(timestamp), profile_updated(timestamp).
- **Security:** security_check_id(serial, primary key)not null, passenger_id(int, foreign key) not null, check_results(varchar) not null, created_at(timestamp), updated_at(timestamp).

3. Normalization:
The conditions of 1NF are already satisfied: relation is atomic and single-valued, there is a primary key and non-repeated values. Conditions of 2NF are already satisfied, each non-key value depends on primary key. In 3NF the column check_results in security and checking were added together to another table Check_results .

4. Relations: Bookings -> Boarding (one to many), Passengers ->Bookings(one to many), Passengers -> security(one to many), Passengers -> Checking(one to many), Airlines -> Flights(one to many), Flights -> Bookings(one to many), Bookings -> Baggage(one to many), Flights -> Bookings -> Passengers(many to many)

5.

6. Attributes are added

7. Relations are showed

8.
- Airlines – Flights: one airline can operate many flights
- Flights – Bookings: a flight can have many bookings

- Passengers – Bookings: one passenger can make many bookings
- Flights – Passengers: many-to-many, one passenger can make many bookings so they may have several flights
- Bookings – Boarding: one booking can have multiple boardings
- Booking – Baggage: one booking can consist of multiple baggages
- Passengers – Checking: one passenger may have multiple security checks
- Passengers – Security: a passenger can go through many security checks

Before 3NF condition is satisfied:
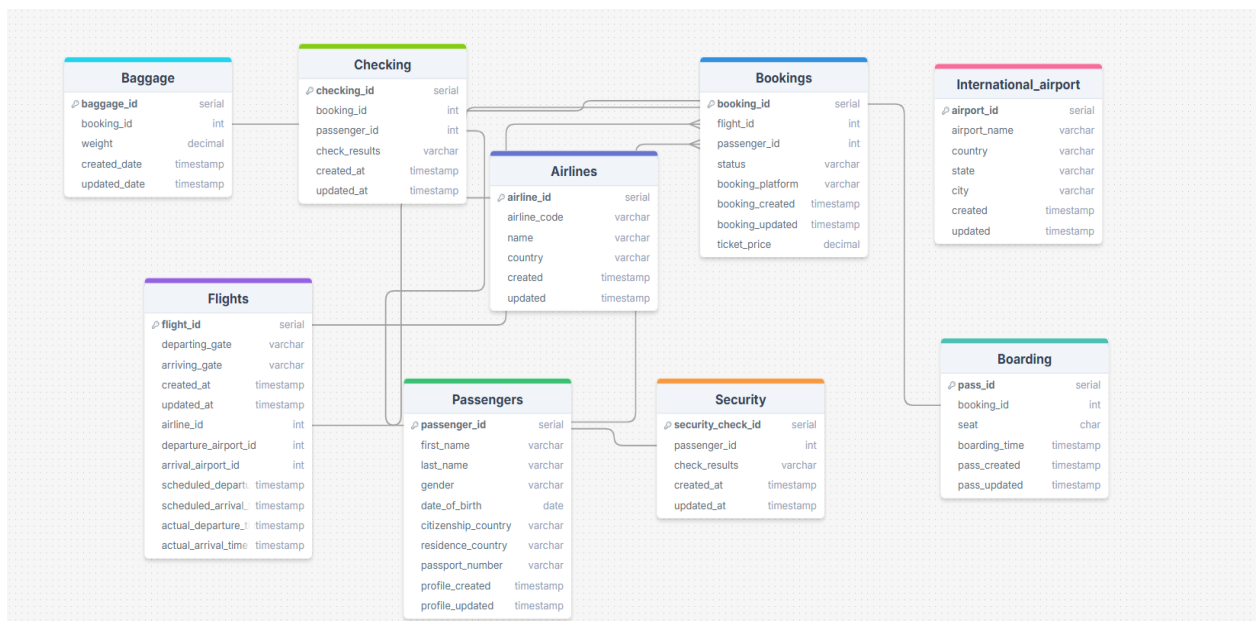
After all conditions are satisfied:
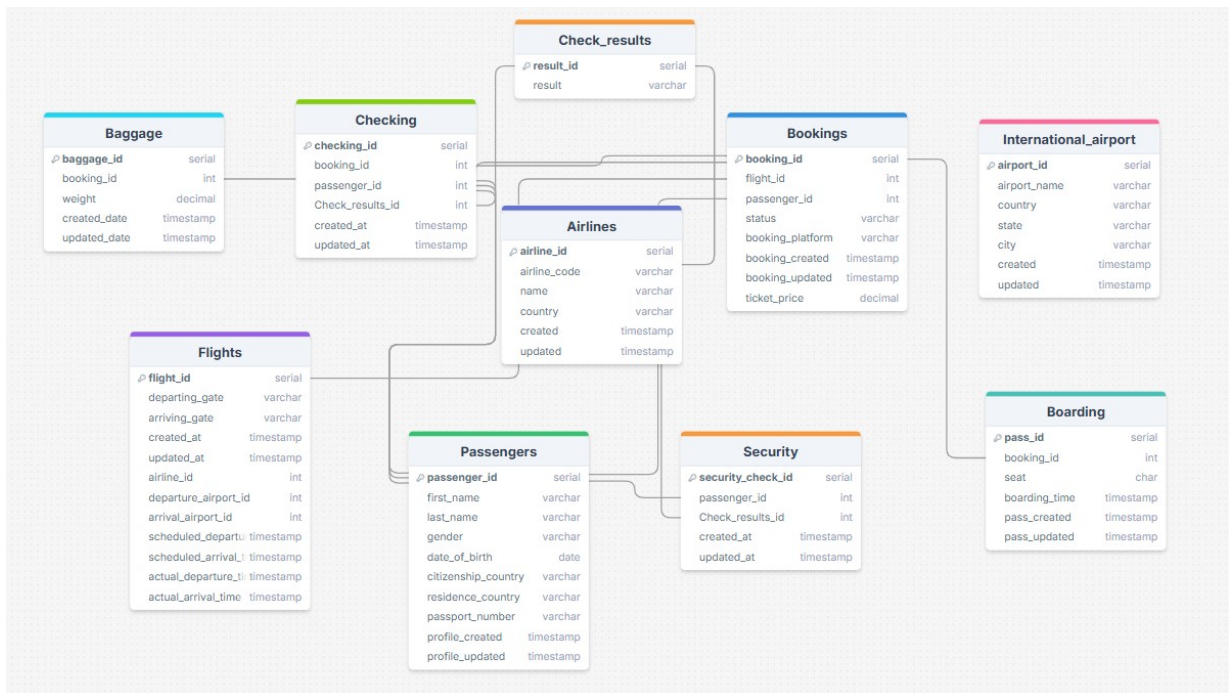
Legend:

□ PK	Primary Key
Lines FK	Foreign Key
Necessary to link	not null
Unnecessary to link	nullable