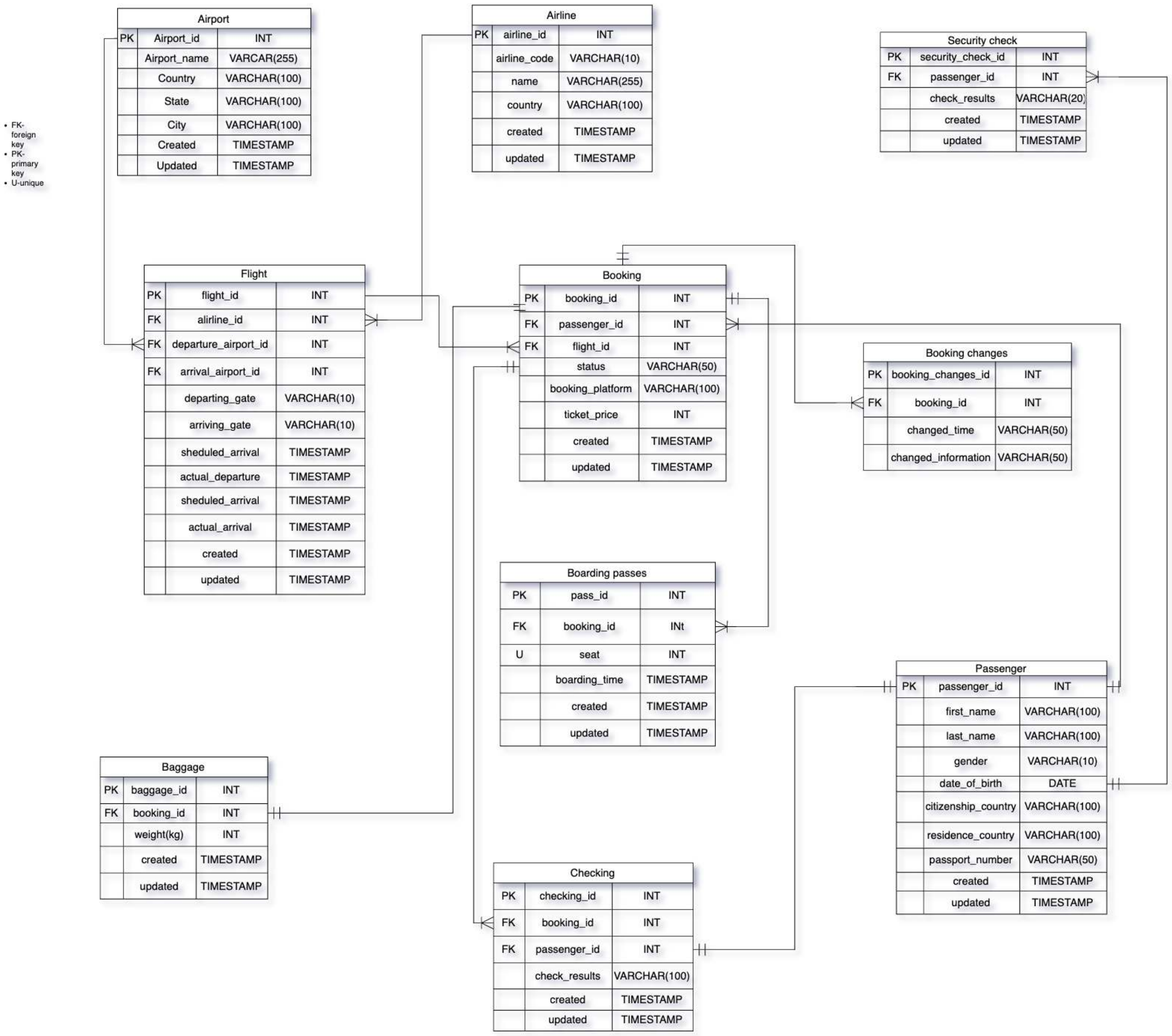

- FK-foreign key
- PK-primary key
- U-unique

Airport		
PK	Airport_id	INT
	Airport_name	VARCAR(255)
	Country	VARCHAR(100)
	State	VARCHAR(100)
	City	VARCHAR(100)
	Created	TIMESTAMP
	Updated	TIMESTAMP

Airline		
PK	airline_id	INT
	airline_code	VARCHAR(10)
	name	VARCHAR(255)
	country	VARCHAR(100)
	created	TIMESTAMP
	updated	TIMESTAMP

Security check		
PK	security_check_id	INT
FK	passenger_id	INT
	check_results	VARCHAR(20)
	created	TIMESTAMP
	updated	TIMESTAMP

Flight		
PK	flight_id	INT
FK	alirline_id	INT
FK	departure_airport_id	INT
FK	arrival_airport_id	INT
	departing_gate	VARCHAR(10)
	arriving_gate	VARCHAR(10)
	sheduled_arrival	TIMESTAMP
	actual_departure	TIMESTAMP
	sheduled_arrival	TIMESTAMP
	actual_arrival	TIMESTAMP
	created	TIMESTAMP
	updated	TIMESTAMP

Booking		
PK	booking_id	INT
FK	passenger_id	INT
FK	flight_id	INT
	status	VARCHAR(50)
	booking_platform	VARCHAR(100)
	ticket_price	INT
	created	TIMESTAMP
	updated	TIMESTAMP

Booking changes		
PK	booking_changes_id	INT
FK	booking_id	INT
	changed_time	VARCHAR(50)
	changed_information	VARCHAR(50)

Boarding passes		
PK	pass_id	INT
FK	booking_id	INt
U	seat	INT
	boarding_time	TIMESTAMP
	created	TIMESTAMP
	updated	TIMESTAMP

Passenger		
PK	passenger_id	INT
	first_name	VARCHAR(100)
	last_name	VARCHAR(100)
	gender	VARCHAR(10)
	date_of_birth	DATE
	citizenship_country	VARCHAR(100)
	residence_country	VARCHAR(100)
	passport_number	VARCHAR(50)
	created	TIMESTAMP
	updated	TIMESTAMP

Baggage		
PK	baggage_id	INT
FK	booking_id	INT
	weight(kg)	INT
	created	TIMESTAMP
	updated	TIMESTAMP

Checking		
PK	checking_id	INT
FK	booking_id	INT
FK	passenger_id	INT
	check_results	VARCHAR(100)
	created	TIMESTAMP
	updated	TIMESTAMP

How I created the tables and the ER diagram

First, I started by reading the system description about the international airport. The text explained that the system must keep information about airports, airlines, flights, passengers, bookings, boarding passes, baggage, baggage checks, and security checks. From this description, I decided what entities I need for my database. These entities are: Airport, Airline, Flight, Passenger, Booking, BookingLeg, BoardingPass, Baggage, BaggageCheck, and SecurityCheck.

Second, for each entity I wrote a list of attributes. For example, the **Passenger** entity has passenger_id (primary key), first_name, last_name, passport_number, date_of_birth, citizenship_country, and residence_country. The **Flight** entity has flight_id (primary key), flight_number, airline_id (foreign key), departure_airport_id (foreign key), arrival_airport_id (foreign key), and times of departure and arrival. For every entity I wrote which attribute is the primary key (PK) and which are foreign keys (FK). I also thought about data types, like INT, VARCHAR, DATE, and TIMESTAMP.

Third, I applied normalization rules. I checked that every table depends only on its primary key. I removed repeating groups and separated information into correct tables. For example, information about airports is in the Airport table, information about airlines is in the Airline table, and information about flights is in the Flight table. This helps to avoid redundancy and keep the database in the third normal form (3NF).

After that, I used **draw.io** to make the ER diagram. I created one block for each entity. Inside each block, I wrote the attributes. I marked primary keys with “PK” and foreign keys with “FK”. Then I connected the entities with lines. I used crow’s foot symbols to show the relationships: one-to-one, one-to-many, or many-to-many. For example, one passenger can have many bookings, and one booking can have many booking legs.

I also created a small legend in the diagram. The legend explains the meaning of PK, FK, and the symbols for relationships.

Finally, I exported my diagram from draw.io as a PNG file. I inserted this PNG image into my report. Under the diagram, I added this text description to explain step by step how I created the tables and why I designed the database this way. Then I exported the whole report as PDF.