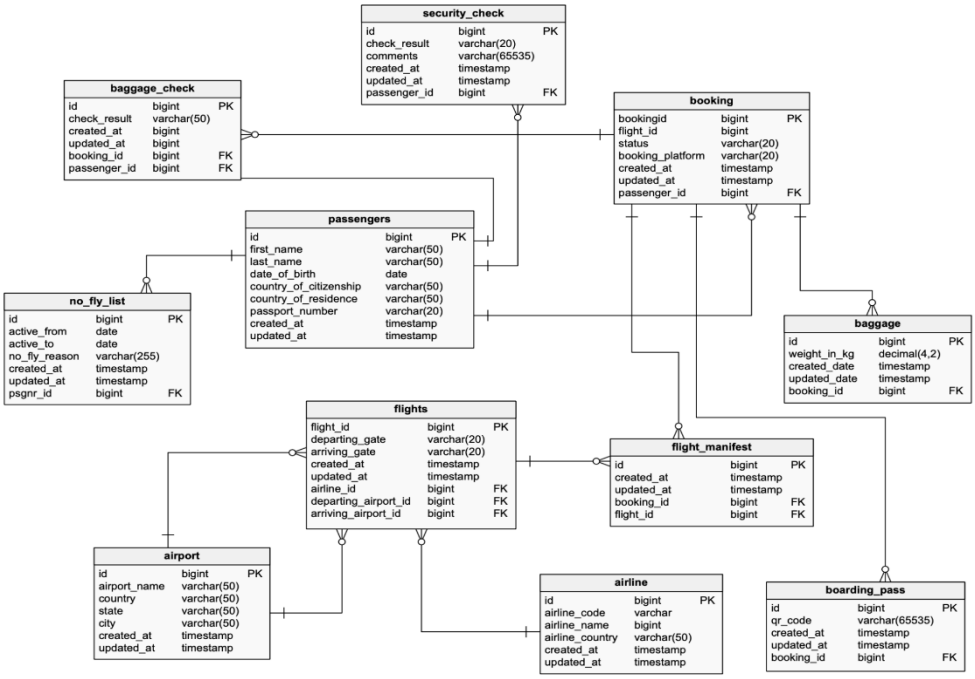
**Question 1:**

**Analyze a given business scenario and create an ER diagram that includes entities, relationships, attributes, and cardinality. Ensure that the diagram reflects proper normalization up to the third normal form.**

Business Scenario:




- Passengers:
    - The system stores passenger information such as first name, last name, date of birth, gender, country of residence, phone number, and email address.
    - Each passenger has a unique identifier (passenger_id).
- Bookings:
    - Passengers can make bookings for flights.
    - Each booking has a unique identifier (booking_id) and contains information such as booking date, flight (flight_id), booking status, and the associated passenger (passenger_id).
- Flights:

- The system manages flight details, including flight number, departure date, arrival date, departure airport (departing_airport_id), arrival airport (arriving_airport_id), and airline (airline_id).
- Each flight has a unique identifier (flight_id).

- Airports:
  - The system stores information about airports, including airport code, airport name, country, city, and other relevant details.
  - Each airport has a unique identifier (airport_id).

- Airlines:
  - The system maintains information about airlines, such as the airline code, airline name, and other relevant details.
  - Each airline has a unique identifier (airline_id).

- Baggage:
  - The system tracks baggage information associated with each booking, including baggage weight, baggage_id, and the booking_id it is linked to.

- Boarding Passes:
  - Passengers receive boarding passes for their booked flights.
  - Each boarding pass has a unique identifier (boarding_pass_id) and is linked to a specific booking (booking_id).

- Flight Manifest:
  - The system generates a flight manifest for each flight, which contains information about the passengers on that flight.
  - Each flight manifest has a unique identifier (flight_manifest_id) and is linked to a specific flight (flight_id).

- Security Checks:
  - Passengers go through security checks before boarding their flights.
  - Each security check has a unique identifier (security_check_id) and is linked to a specific passenger (passenger_id) and flight (flight_id).

- Baggage Checks:
  - Passengers' baggage undergoes security checks.
  - Each baggage check has a unique identifier (baggage_check_id) and is linked to a specific booking (booking_id) and passenger (passenger_id).

The provided image is an Entity-Relationship (ER) diagram that represents a system or database schema for an airline or airport management scenario. Let me analyze and explain the different components of the diagram:

**Entities:**

1. baggage_check: This entity represents the baggage check process for passengers.
2. security_check: This entity represents the security check process for passengers.
3. passengers: This entity stores information about passengers.
4. no_fly_list: This entity likely contains a list of individuals who are not allowed to fly.
5. flights: This entity represents information about individual flights.
6. airport: This entity stores information about airports.
7. airline: This entity represents airlines or carriers.
8. booking: This entity contains information about passenger bookings or reservations.
9. baggage: This entity stores information about passenger baggage.
10. flight_manifest: This entity represents the flight manifest, which is a detailed document listing passengers, crew, and cargo for a specific flight.
11. boarding_pass: This entity likely represents the boarding passes issued to passengers for their flights.

**Relationships:**

1. baggage_check has a many-to-one relationship with passengers and booking.
2. security_check has a many-to-one relationship with passengers.
3. no_fly_list has no explicit relationship shown.
4. flights has a many-to-one relationship with airport (for both departing and arriving airports) and airline.
5. booking has a many-to-one relationship with passengers and flights.
6. baggage has a many-to-one relationship with booking.
7. flight_manifest has a many-to-one relationship with booking and flights.
8. boarding_pass has a many-to-one relationship with booking.

**Attributes:**

The attributes for each entity are appropriately defined, including primary keys and foreign keys where applicable.

**Normalization:**

The diagram appears to be in the third normal form (3NF) as there are no transitive dependencies present, and all non-key attributes are fully dependent on the primary keys of their respective entities.

**Cardinality:**

The cardinalities of the relationships are represented correctly, indicating the maximum number of instances that can be associated with the related entities.

Overall, the provided ER diagram accurately models the given airline/airport management system scenario, including the necessary entities, relationships, attributes, and cardinalities. It also adheres to the principles of normalization up to the third normal form (3NF), ensuring data integrity and consistency.