Staff(id: int, name: string, email: string, username: string, password: string, title: string)
Primary Key: id

Candidate Keys: id, email, username

FDs:

- id —> name, email, username, password, title
  A staff account id should determine the information associated with the staff

Already in BCNF.

Customer(id: int, name: string, email: string, username: string, password: string, travel_document: string, billing_address: string, phone_number: string, seat_preference: enum, payment_information: string)
Primary Key: id

Candidate Keys: id, email, username, travel_document

FDs:

- id —> name, email, username, password, travel_document, billing_address, phone_number, seat_preference, payment_information
  A customer's account id should determine the information associated with the customer

Already in BCNF.

Loyalty_Member(**id**: int, points: int)
Primary Key: id

Candidate Key: id

Foreign Key:

- id references Customer.id

FDs:

- id —> points
  A customer's account id should determine the point balance

Already in BCNF.

Ticket(id: int, seat_type: enum, price: int, **flightId**: int, **customerId**: int)
Primary Key: id

Candidate Keys: id, (flightId, customerId)

Foreign Keys:

- flightId references Flight.id
- customerID references Customer.id

FDs:

- id —> seat_type, price, flightId, customerId
  A ticket id should determine the seat, price, flight, and customer information

Already in BCNF.

Aircraft(id: string, type: string, first_class_seats: int, business_seats: int, economy_seats: int, purchase_date: date, status: enum)
Primary Key: id

Candidate Key: id

FDs:

- id —> type, purchase_date, status
- type —> first_class_seats, business_seats, economy_seats
  The aircraft id can be used to determine its type. Then the type of aircraft can be used to determine the arrangement of seats.

Normalization (BCNF):

Type(type: string, first_class_seats: int, business_seats: int, economy_seats: int)

Aircraft(id: string, type: string, purchase_date: date, status: enum)

Airport(id: string, name: string, location: string)
Primary Key: id

Candidate Keys: id, location, name

FDs:

- id —> name, location
  An airport's id code (e.g. YVR) can be used to determine its full name, and the location of the airport

Already in BCNF.

Flight(id: int, date_time: date, **assigned**: string, **arrival**: string, **departure**: string)
Primary Key: id

Candidate Keys: id, assigned, (arrival, departure)

Foreign Keys:

- assigned references Aircraft.id
- arrival references Route.arrival
- departure references Route.departure

FDs:

- id —> date_time, assigned, arrival, departure
  A flight id should determine the date_time, the aircraft assigned to it, and the arrival and departure airports

Already in BCNF.

Route(**departure**: string, **arrival**: string, first_class: int, business: int, economy: int)
Primary Key: (departure, arrival)

Candidate Key: (departure, arrival)

Foreign Keys:

- arrival references Airport.id
- departure references Airport.id

FDs:

- departure, arrival —> first_class, business, economy
  A flights departure and arrival airports should determine the various ticket prices

Already in BCNF.