Airport Database Project - Relational Model

Normalization 1NF
This database is in 1NF. Each table has only a single(atomic) valued attribute/columns. Values stored in a column are of the same domain. All the columns in a table have unique names. And the order in which data is stored, does not matter. The following changes were made to get to 1NF:

- The crew attribute in the flight table was not single valued, and when separated caused the primary key to be duplicated. We created a new table "CrewInfo" that paired crew with designated flight (new primary key).

Normalization 2NF:
The tables in this database are in 2NF. It is already in 1NF and there are no longer any partial dependencies. The following changes were made to get to 2NF:

- Listed all dependencies and found candidate keys existed in the Passenger, Flight, and Booking tables.
- Functional dependencies that needed to change: Passenger (passenger_id, passport —> other attributes), Flight(capacity, dep_time, arr_time, seats_booked, open_seats, status, destination —> dep_gate), Booking(status, ticket_price, booking_date —> seat)
- Created new tables for attributes (passport_no, dep_gate, seat)

Normalization 3NF:
The tables in this database are in 3NF. It is already in 2NF and no longer has any transitive dependencies. The following changes were made to get to 3NF and make tables more dynamic:

- These new tables now allow for Updates, Insertions, and Deletions to be changed without redundancy. We can now make changes like switching "y" to "yes" in one area.
- Created Tables: Is_Seat_Filled, IsFirstClass, Aircraft, JobTitle, Flight_Status, Price_Per_Weight, Checkin_status, Ticket_Price, and Booking_Status

Normalization BCNF:
The tables in this database are in BCNF. The table is in 3NF and the determinant is the superkey (or primary key).

Normalization 4NF:
The tables in this database are in 4NF. The tables are in BCNF and there are no multivalued dependencies.

Assumptions
We are LGA airport
A passenger can only book a flight with our airline. No outside bookings.
A booking can contain a one-way trip or round trip only.
Flight manifest can be pulled using reports and queries.
Tickets are only first class or economy and have a fixed price.
Each occupation has the same salary, for example, all pilots are paid $120,000.

Note: We have the data types for every attribute, it is just not in this document.

**Need help with Ticket_Price table, see note below

Relational Model		
Employee	Airline	Airport
Primary Key: emplid **Foreign Key(s)**: title (*reference* JobTitle) *Attributes*: firstname lastname Functional Dependency: emplid -->> firstname, lastname	**Primary Key:** airline_id **Foreign Key(s):** base_airport (*reference* Airport) *Attributes:* airline_name Functional Dependency: airline_id -->> airline_name	**Primary Key:** airport_name **Foreign Key(s)**: n/a *Attributes:* location no_of_gates Functional Dependency: airport_name -->> location, no_of_gates
Passenger	Flight	CrewInfo
Primary Key: passenger_id **Foreign Key(s)**: passport_no *(reference* Passport) *Attributes:* firstname lastname street city state zipcode dob Functional Dependency: passenger_id -->> lastname, firstname, street, city, state, zipcode, dob	**Primary Key:** flight_no **Foreign Key(s):** airline_id (*reference* Airline) dep_gate (*reference* Gate) aircraft_id (*reference* Aircraft) status (*reference* Flight_Status) *Attributes:* dep_time arr_time seats_booked open_seats destination Functional Dependency: flight_no -->> dep_time, arr_time, seats_booked, open_seats, destination	**Primary Key:** id **Foreign Key(s):** flight_no (*reference* Flight) emplid (*reference* Employee) *Attributes:* n/a Functional Dependency: id -->> flight_no, employee
Baggage_Claim	Gate	Check_in
Primary Key: bag_id **Foreign Key(s):** flight_no (*reference* Flight) passenger_id (*reference* Passenger) gate_id (*reference* Gate) class (*reference* Price_per_Weight) *Attributes:* n/a Functional Dependency: bag_id -->> bag_id (given)	**Primary Key:** gate_id **Foreigh Key(s):** n/a *Attributes:* n/a Functional Dependency: gate_id -->> gate_id	**Primary Key:** passport_no **Foreign Key(s):** passport_no (*reference* Passport) passenger_id (*reference* Passenger) checked_in (*reference* Checkin_Status) *Attributes:* n/a Functional Dependency: passport_no -->> passport_no
Booking	Aircraft	Seat

Primary Key: booking_id **Foreign Key(s):** flight_no (*reference* Flight) passenger_id (*reference* Passenger) bag_id (*reference* Baggage_Claim) seat_id (*reference* Seat) status (*reference* Booking_Status) ticket_price (*reference* Ticket_Price) *Attributes:* booking_date Functional Dependency: booking_id --↠ booking_date	**Primary Key:** id **Foreign Key(s):** n/a *Attributes:* type capacity Functional Dependency: id --↠ type, capacity	**Primary Key:** seat_id **Foreign Key(s):** aircraft_id (*reference* Aircraft) is_firstclass (*reference* IsFirstClass) is_booked (*reference* Is_Seat_Filled) *Attributes:* n/a Functional Dependency: seat_id --↠ seat_id
IsFirstClass	JobTitle	Price_Per_Weight
Primary Key: is_firstclass **Foreign Key(s):** n/a *Attributes:* price Funtional Dependency: is_firstclass --↠ price	**Primary Key:** title **Foreign Key(s):** n/a *Attributes:* salary Functional Dependency: title --↠ salary	**Primary Key:** class **Foreign Key(s):** n/a *Attributes:* weight_range price Functional Dependency: class --↠ weight_range, price
Checkin_Status	Booking Status	Flight_Status
Primary Key: status **Foreign Key(s):** n/a *Attributes:* n/a Functional Dependency: status --↠ status (this is given for all the others)	**Primary Key:** status **Foreign Key(s):** n/a *Attributes:* n/a Functional Dependency: status--↠ status (given)	**Primary Key:** status **Foreign Key(s):** n/a *Attributes:* n/a Functional Dependency: status --↠ status
Passport	Is_Seat_Filled	
Primary Key: passport_no **Foreign Key:** passenger_id *(reference* Passenger) *Attribute:* country_of_orgin expiration Functional Dependency: passport_no --↠country_of_orgin, expiration	**Primary Key:** id **Foreign Key(s):** n/a *Attributes:* n/a Functional Dependency: id --↠ id	