SkyLogix : Airline Database Management System PHASE 3

By:- Data Chronicles (Team Number 24)

Team Members -

Archisha Panda (2022111019)
Gargi Shroff (2022114009)
Paridhi Jain (2022101119)
Manda Vaishnavi Reddy (2022111018)
Vyakhya Gupta (2022101104)

Mapping ER To Relational Model

1. Created a relation for each strong entity type. Decomposed composite attributes to simple attributes (eg. Name: Fname, Lname).
   For Employee (which is a superclass) - one relation comprising common attributes to all subclasses.
   For all the disjoint subclasses - Individual relations containing Primary Key as EmployeeID and only those attributes which are unique to each subclass.

2. Created a relation for each weak entity type and included primary key of owner as foreign key in the attributes.

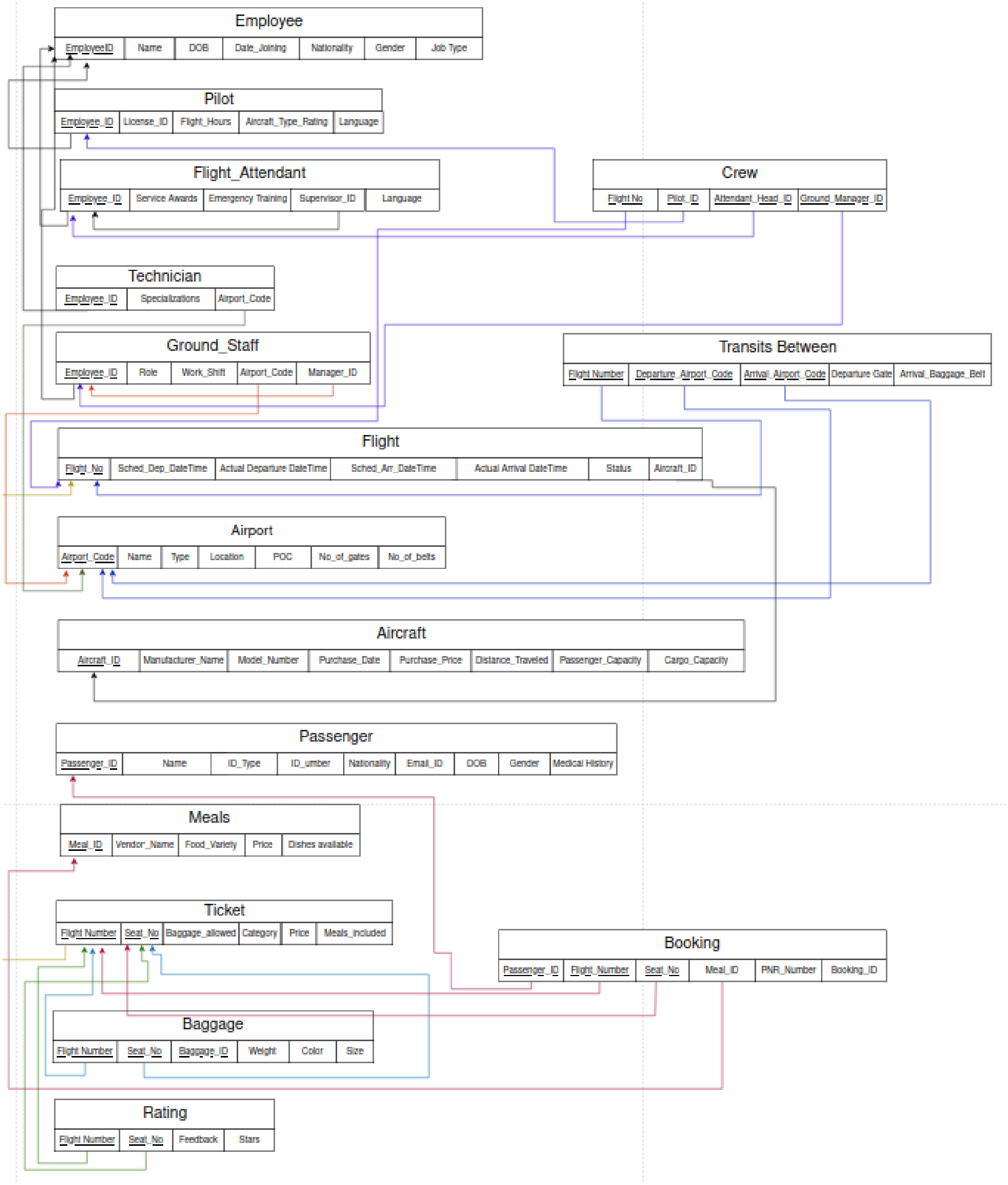
3. Incorporated 1:1 binary relationships using foreign key approach, where the primary key of the entity is added as a foreign key attribute to the entity with total participation.
   Example: Flight USES Aircraft

4. Incorporated 1:N and N:1 binary relationships using foreign key approach. Included as foreign key in N-side entity, the primary key of the other entity type.
   Example: Ground Staff POSTED AT Airport

5. Mapped binary M:N relationships using Cross Reference approach, where a separate relation is made which will have the combination of primary keys of the participating entity types as its primary key. Each of these primary keys is a foreign key for the new relation. Other attributes represent the attributes of the relationship in the ER Model.
   Example: Flight SERVES Meals

6. Mapped multivalued attributes- For each entity type, create a separate relation for multi-valued attributes whose foreign key is the primary key of the corresponding entity.
   Example: Service Awards for Flight Attendants

7. Mapped N-ary relation types- Created a relation whose primary key is combination of primary keys of the participating entities with other attributes being the attributes of the relationship type. Each of these primary keys is a foreign key for the new relation.
   Example: TRANSITS BETWEEN relationship


Employee
EmployeeID | Name | DOB | Date_Joining | Nationality | Gender | Job Type
Pilot
Employee_ID | License_ID | Flight_Hours | Aircraft_Type_Rating | Language
Flight_Attendant
Employee_ID | Service Awards | Emergency Training | Supervisor_ID | Language
Crew
Flight No | Pilot_ID | Attendant_Head_ID | Ground_Manager_ID
Technician
Employee_ID | Specializations | Airport_Code
Ground_Staff
Employee_ID | Role | Work_Shift | Airport_Code | Manager_ID
Transits Between
Flight Number | Departure_Airport_Code | Arrival_Airport_Code | Departure Gate | Arrival_Baggage_Belt
Flight
Flight_No | Sched_Dep_DateTime | Actual Departure DateTime | Sched_Arr_DateTime | Actual Arrival DateTime | Status | Aircraft_ID
Airport
Airport_Code | Name | Type | Location | POC | No_of_gates | No_of_belts
Aircraft
Aircraft_ID | Manufacturer_Name | Model_Number | Purchase_Date | Purchase_Price | Distance_Traveled | Passenger_Capacity | Cargo_Capacity
Passenger
Passenger_ID | Name | ID_Type | ID_umber | Nationality | Email_ID | DOB | Gender | Medical History
Meals
Meal_ID | Vendor_Name | Food_Variety | Price | Dishes available
Ticket
Flight Number | Seat_No | Baggage_allowed | Category | Price | Meals_included
Booking
Passenger_ID | Flight_Number | Seat_No | Meal_ID | PNR_Number | Booking_ID
Baggage
Flight Number | Seat_No | Baggage_ID | Weight | Color | Size
Rating
Flight Number | Seat_No | Feedback | Stars

Minor Changes

- The attribute ‘Address’ in entity types- Employee and Passenger was removed.
- Some attribute names have been abbreviated (For example Flight Number written as Flight No.)

- The attribute ‘Meal Included’ from Ticket was removed.
- The attribute type for a few attributes were changed like Work-Shift, Role, Color and Size changed from multi-valued to simple.
- Removed address from employee and passenger

First Normal Form (1NF)

If a relation contains a composite or multi-valued attribute, it violates the first normal form, or the relation is in the first normal form if it does not contain any composite or multi-valued attribute. A relation is in first normal form if every attribute in that relation is a single-valued attribute.

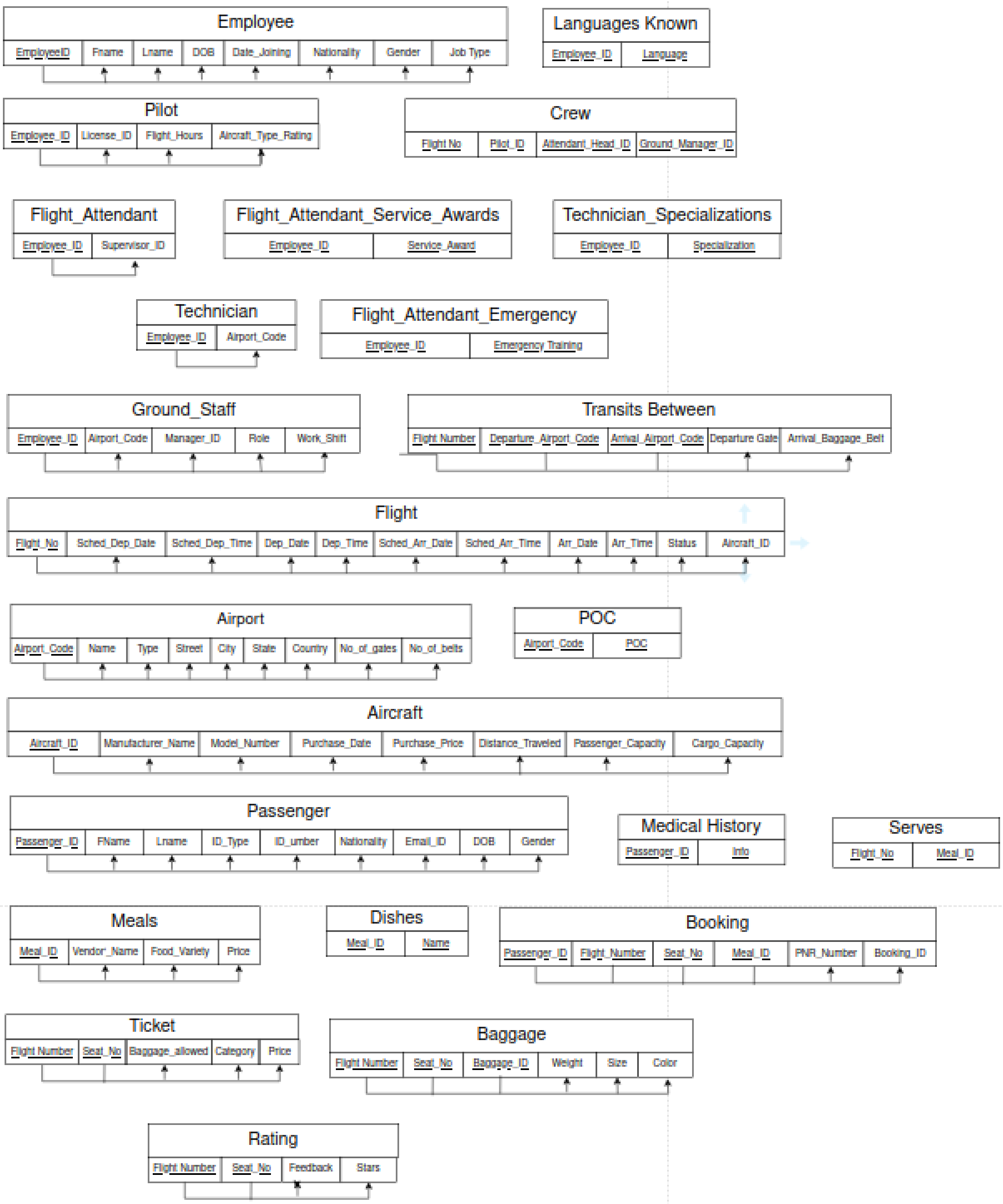
The table has following Multivalued Attributes:

- Language
- Flight Attendant’s Service awards
- Technician’s Specialization
- Flight Attendant’s Emergency Training
- Airport’s POC
- Serves
- Passangers’s Medical History
- Dishes Available (Meals)

The table has the following Composite Attributes:

- Scheduled Departure DateTime
- Actual Departure DateTime
- Scheduled Arrival DateTime
- Actual Arrival DateTime
- Name
- Location

1. To remove multivalued attributes for each entity type, create a separate relation for multi-valued attributes whose foreign key is the primary key of the corresponding entity. The Primary Key for the new table formed is the combination of the Foreign Key and the Multivalued Attribute.
   For Example: Service Awards for Flight Attendants, Medical history of Passengers.
2. Secondly to remove composite attributes, instead of a single column, a composite key uses multiple columns to form the identifier.
   For Example: Scheduled date and scheduled time are placed in separate columns instead of one in flights, name of employee broken down into first name and last name in employee.



Employee

EmployeeID	Fname	Lname	DOB	Date_Joining	Nationality	Gender	Job Type

Languages Known

Employee_ID	Language

Pilot

Employee_ID	License_ID	Flight_Hours	Aircraft_Type_Rating

Crew

Flight No	Pilot_ID	Attendant_Head_ID	Ground_Manager_ID

Flight_Attendant

Employee_ID	Supervisor_ID

Flight_Attendant_Service_Awards

Employee_ID	Service_Award

Technician_Specializations

Employee_ID	Specialization

Technician

Employee_ID	Airport_Code

Flight_Attendant_Emergency

Employee_ID	Emergency Training

Ground_Staff

Employee_ID	Airport_Code	Manager_ID	Role	Work_Shift

Transits Between

Flight Number	Departure_Airport_Code	Arrival_Airport_Code	Departure Gate	Arrival_Baggage_Belt

Flight

Flight_No	Sched_Dep_Date	Sched_Dep_Time	Dep_Date	Dep_Time	Sched_Arr_Date	Sched_Arr_Time	Arr_Date	Arr_Time	Status	Aircraft_ID

Airport

Airport_Code	Name	Type	Street	City	State	Country	No_of_gates	No_of_belts

POC

Airport_Code	POC

Aircraft

Aircraft_ID	Manufacturer_Name	Model_Number	Purchase_Date	Purchase_Price	Distance_Traveled	Passenger_Capacity	Cargo_Capacity

Passenger

Passenger_ID	FName	Lname	ID_Type	ID_umber	Nationality	Email_ID	DOB	Gender

Medical History

Passenger_ID	Info

Serves

Flight_No	Meal_ID

Meals

Meal_ID	Vendor_Name	Food_Variety	Price

Dishes

Meal_ID	Name

Booking

Passenger_ID	Flight_Number	Seat_No	Meal_ID	PNR_Number	Booking_ID

Ticket

Flight Number	Seat_No	Baggage_allowed	Category	Price

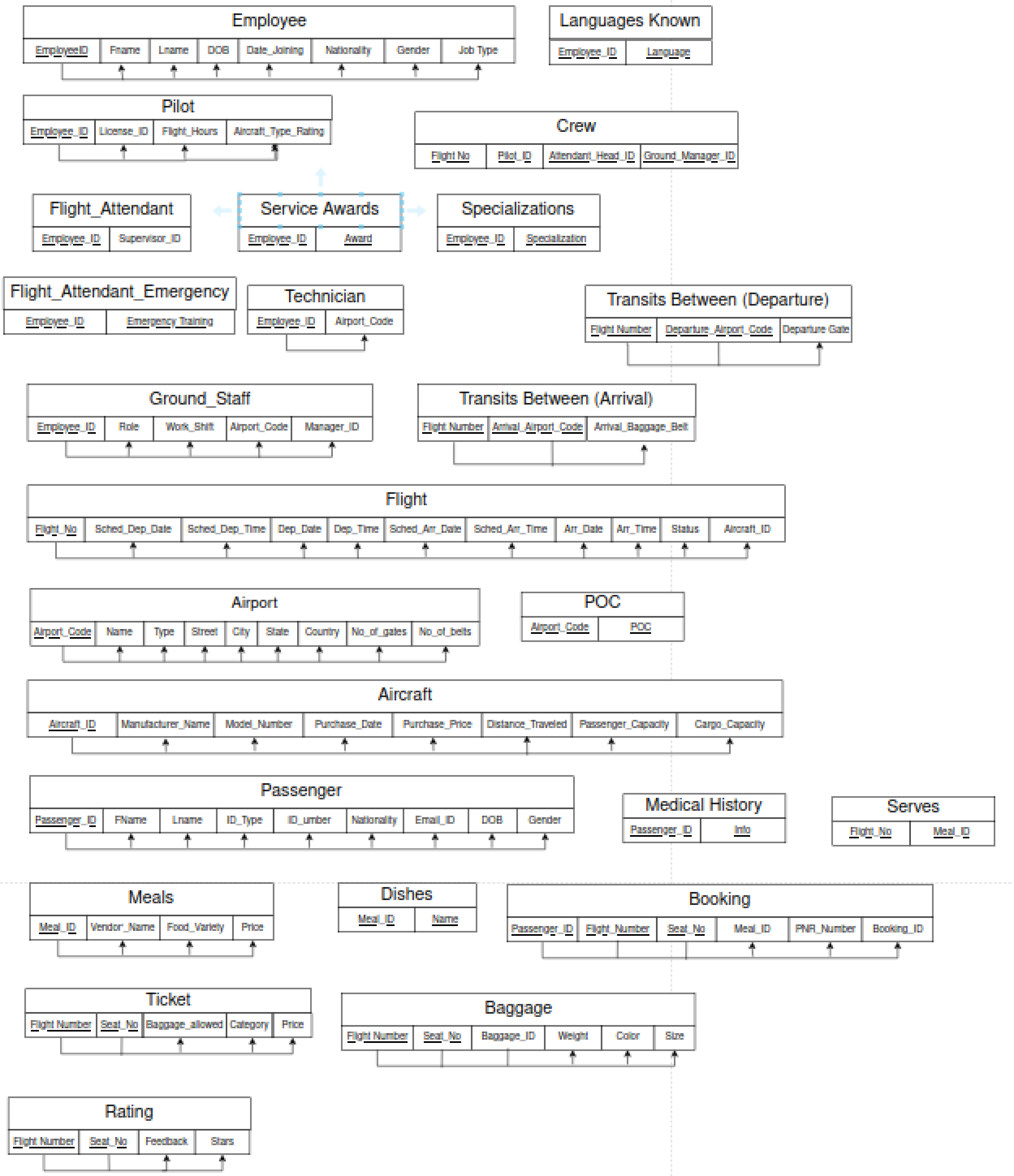
Baggage

Flight Number	Seat_No	Baggage_ID	Weight	Size	Color

Rating

Flight Number	Seat_No	Feedback	Stars

Second Normal Form (2NF)



Employee
EmployeeID | Fname | Lname | DOB | Date_Joining | Nationality | Gender | Job Type

Languages Known
Employee_ID | Language

Pilot
Employee_ID | License_ID | Flight_Hours | Aircraft_Type_Rating

Crew
Flight No | Pilot_ID | Attendant_Head_ID | Ground_Manager_ID

Flight_Attendant
Employee_ID | Supervisor_ID

Service Awards
Employee_ID | Award

Specializations
Employee_ID | Specialization

Flight_Attendant_Emergency
Employee_ID | Emergency Training

Technician
Employee_ID | Airport_Code

Transits Between (Departure)
Flight Number | Departure_Airport_Code | Departure Gate

Ground_Staff
Employee_ID | Role | Work_Shift | Airport_Code | Manager_ID

Transits Between (Arrival)
Flight Number | Arrival_Airport_Code | Arrival_Baggage_Belt

Flight
Flight_No | Sched_Dep_Date | Sched_Dep_Time | Dep_Date | Dep_Time | Sched_Arr_Date | Sched_Arr_Time | Arr_Date | Arr_Time | Status | Aircraft_ID

Airport
Airport_Code | Name | Type | Street | City | State | Country | No_of_gates | No_of_belts

POC
Airport_Code | POC

Aircraft
Aircraft_ID | Manufacturer_Name | Model_Number | Purchase_Date | Purchase_Price | Distance_Traveled | Passenger_Capacity | Cargo_Capacity

Passenger
Passenger_ID | FName | Lname | ID_Type | ID_umber | Nationality | Email_ID | DOB | Gender

Medical History
Passenger_ID | Info

Serves
Flight_No | Meal_ID

Meals
Meal_ID | Vendor_Name | Food_Variety | Price

Dishes
Meal_ID | Name

Booking
Passenger_ID | Flight_Number | Seat_No | Meal_ID | PNR_Number | Booking_ID

Ticket
Flight Number | Seat_No | Baggage_allowed | Category | Price

Baggage
Flight Number | Seat_No | Baggage_ID | Weight | Color | Size

Rating
Flight Number | Seat_No | Feedback | Stars

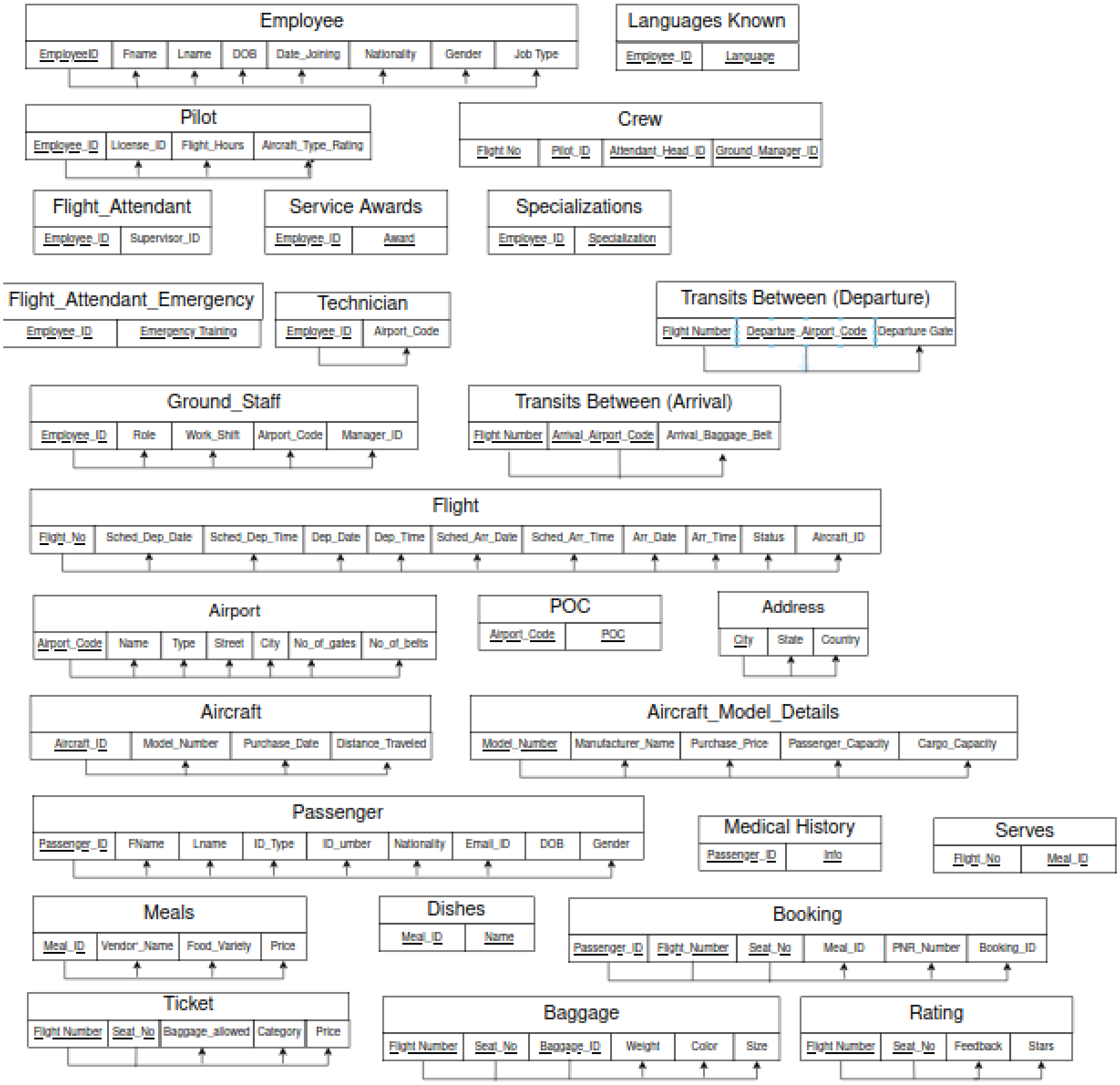
- The second Normal Form (2NF) is based on the concept of fully functional dependency.
- The second Normal Form applies to relations with composite keys, that is, relations with a primary key composed of two to or more attributes. A relation with a single-attribute primary key is automatically in at least 2NF.

- A relation is in 2NF if it has No Partial Dependency, i.e., no non-prime attribute (attributes that are not part of any candidate key) is dependent on any proper subset of any candidate key of the table.

- Removal of Partial Dependencies -
  1. In Transits Between Table -
     - ❖ The Type (of flight, International/Domestic) can be determined only with the help of Flight_Number. Hence, we made a different table for determining the Type.
     - ❖ The Departure_Gate can be retrieved by using only the Flight_Number and the Departure_Airport_Code similarly the Arrival_Baggage_Belt can be retrieved using the Flight_Number and the Arrival_Airport_Code. Hence, we can split the table into these different tables to ensure that there are No Partial Dependencies.

  2. In Rating Table -
     - ❖ The Feedback and Stars given for a particular rating can be retrieved using Flight_Number and Seat_Number or Flight_Number and Rating_Number (Since, a person in a given flight is allowed to give ratings only once). Therefore, we can split the table into two different tables to ensure that there are No Partial Dependencies.

Third Normal Form (3NF)



Employee

EmployeeID	Fname	Lname	DOB	Date_Joining	Nationality	Gender	Job Type

Languages Known

Employee_ID	Language

Pilot

Employee_ID	License_ID	Flight_Hours	Aircraft_Type_Rating

Crew

Flight No	Pilot_ID	Attendant_Head_ID	Ground_Manager_ID

Flight_Attendant

Employee_ID	Supervisor_ID

Service Awards

Employee_ID	Award

Specializations

Employee_ID	Specialization

Flight_Attendant_Emergency

Employee_ID	Emergency Training

Technician

Employee_ID	Airport_Code

Transits Between (Departure)

Flight Number	Departure_Airport_Code	Departure Gate

Ground_Staff

Employee_ID	Role	Work_Shift	Airport_Code	Manager_ID

Transits Between (Arrival)

Flight Number	Arrival_Airport_Code	Arrival_Baggage_Belt

Flight

Flight_No	Sched_Dep_Date	Sched_Dep_Time	Dep_Date	Dep_Time	Sched_Arr_Date	Sched_Arr_Time	Arr_Date	Arr_Time	Status	Aircraft_ID

Airport

Airport_Code	Name	Type	Street	City	No_of_gates	No_of_belts

POC

Airport_Code	POC

Address

City	State	Country

Aircraft

Aircraft_ID	Model_Number	Purchase_Date	Distance_Traveled

Aircraft_Model_Details

Model_Number	Manufacturer_Name	Purchase_Price	Passenger_Capacity	Cargo_Capacity

Passenger

Passenger_ID	FName	Lname	ID_Type	ID_umber	Nationality	Email_ID	DOB	Gender

Medical History

Passenger_ID	Info

Serves

Flight_No	Meal_ID

Meals

Meal_ID	Vendor_Name	Food_Variety	Price

Dishes

Meal_ID	Name

Booking

Passenger_ID	Flight_Number	Seat_No	Meal_ID	PNR_Number	Booking_ID

Ticket

Flight Number	Seat_No	Baggage_allowed	Category	Price

Baggage

Flight Number	Seat_No	Baggage_ID	Weight	Color	Size

Rating

Flight Number	Seat_No	Feedback	Stars

- In essence, for a database to be in third normal form (3NF):
  1. It must already be in 2NF.
  2. It should eliminate transitive dependencies. This means that every non-prime attribute (an attribute not part of any candidate key) should be functionally dependent on the primary key, rather than on another non-prime attribute.

Note: If P -> Q and Q -> R are two functional dependencies, then P -> R is known as a transitive dependency. When normalizing a 2NF relation to 3NF, we remove these transitive dependencies.

- By removing transitive dependencies, 3NF ensures that each non-prime attribute is directly related to the primary key, improving data integrity and minimizing update anomalies.
- Removing Transitive Dependencies -
    - In the AIRCRAFT table, the attributes manufacturer, purchase price, passenger capacity and cargo capacity have been eliminated as these are dependent on model_no. and model_no. inturn depends on the aircraft_id.
    - To eliminate these transitive dependencies, model_no., manufacturer, purchase price, passenger capacity and cargo capacity have been added to a new table Aircraft_Model_Details, with model_no. as the primary key.

Assumption: Aircrafts of the same model_no. have the same manufacturer, purchase price, passenger capacity and cargo capacity.

- Similarly, in the AIRPORT table the attributes state and country have been removed as these can be determined from the city which depends on the Airport_Code. These have been added to another table ADDRESS with city as the primary key.

Assumption: City names are unique and no two states have the same city names.