University of Texas at Arlington

Database Term Project

Airline Management System

Phase 4

Rudy Orozco

1001864902

rjo4902@mavs.uta.edu

Dr. Kyungseo Park

3330 - 002

Abstract

This project aims to build a Database Management System (DBMS) from the ground up and with it creating a system of our own. The Airline Management System aims to use the DBMS to track and record important aspects of an airline like the aircraft, the pilots, the destinations, and the routes available. With Phase 4, this aims to implement and show how a database is connected using a server, coding language and other methods to use.

Step 1: Consider what to do

This database project will be an AIRLINE_MANAGEMENT_SYSTEM in which we'll be keeping track of all the AIRCRAFTs, PILOTs, DESTINATIONs, AIRLINEs, and ROUTEs that are available to the airline and will be treated as entities. What this database does is keep track of the current aircraft in service for the "company", the pilots that are in service and what aircraft they can use, what routes they take and so on.

The reason why a DBMS is needed for something like this is because it helps with organization and keeps track of certain important details like the life of an aircraft and when it is needed for maintenance, who is flying what aircraft and what routes they are going to take, who will be in command and who will be a pilot and who may be training.

In order to link the entities together, we will use relationships that will help us track several things like where a route may Depart_To and Arrive_At, which pilots can Operate_ this route and what aircraft are Assigned_To the route. Alongside with the destinations we can track which pilots are Based_At, and what aircraft are Parked_At. With each pilot, there are pilots that can Command_ an aircraft and also a Copilot_For other pilots flying.

Step 2: Problem definition, user requirements

The system will be organized into separate AIRCRAFTs which the database stores and each will have a unique tail number, unique serial number, hours flown, Manufacturer, Model, code/size, range, start date, and hours until maintenance.

- The database will keep track of the hours of each aircraft flown and the hours until maintenance.
- Once an aircraft has been serviced, the amount of hours remaining resets to the recommended amount assigned by the manufacturer.

The database will also store every individual PILOTs name, a unique SSN, pilot ID, email, gender, birthdate, total hours flown, salary, and start date.

- The database will keep track of the hours flown for the pilot.
- A pilot may be a co-pilot to another if the aircraft requires one.
- A pilot may be assigned to command an aircraft, alongside any other co-pilots.
- An aircraft may not have more than 3 pilots per each one.

We will also keep track of the available DESTINATIONs the Airline provides, which will include the country's name, city, name of the airport, and a unique IATA code (DFW for example).

- Multiple pilots may be based at one of these DESTENATIONs
- Multiple aircraft may be parked at one of these DESTINATIONs
- The database will track if a parked aircraft or a based pilot changes locations/DESTINATIONs.

The ROUTE is another entity that will be tracked as it will provide a unique Flight number, distance of the route, and the departing and arriving airports using the IATA codes. There will be a unique route per each flight.

- The database tracks the ROUTE departing DESTINATION and arrival DESTINATION.
- an aircraft may be assigned to fly a route.
- a pilot may be assigned to operate a route, alongside any other co pilots or pilots in training.

The AIRLINE is an entity that holds and maintains a number of aircraft as it has a unique Airline ID, a name, and the country or destination it is originally from.

- The database will track what AIRCRAFTs are currently under that airline
- The database will also track what ROUTEs the airline is able to use
- There will also be a list of PILOTs that will work under the airline

Step 3: Formulate 10 realistic English queries

1. Gather all aircraft that have been flown between 3,000 to 8,000 hours.
2. Get all the names of pilots that can fly an aircraft currently based in Dallas.
3. What aircraft(s) has a range of at least 3,000 nautical miles.
4. Which pilots can command the 787-8 aircraft?
5. Retrieve all pilots that are female and that fly an Airbus aircraft.
6. Which pilots can fly an aircraft that is set for a route from DFW to NRT?
7. List the number of aircraft that each airline owns.
8. What airline does "Nathan Jones" fly for?
9. List all A320 Aircraft Tailnumbers currently in service.
10. Gather all aircraft that are owned by American Airlines?

Step 4: Entity-Relationship with assumptions

- AIRCRAFT entity type

    Attributes

    - <u>Tail Number</u> (string), Key Attribute
        - Aircraft are assigned a unique tail number, based on ICAO registration requirements. No aircraft should have the same number identification.
    - Serial Number (string)
        - The identification of the aircraft from the manufacturers when the selected aircraft was made. Each aircraft model will have a unique serial number for their respective category and manufacturer.
    - Hours Flown (Decimal)
    - Hours Until Maintenance (Decimal)
        - The amount of flight time until the aircraft needs maintenance. The amount resets after service has been done and is set to the manufacturer's recommendation.
    - Manufacturer (string)
    - Model (string)
    - Code/Size (string)
        - The specific variant of that model, if applicable.
    - Range (decimal)
    - Start of Service (String, mm/dd/yyyy)

    ………

- PILOT entity type
    - Attributes
        - First Name (string)
        - Last Name (string)
        - Middle Initial (character)
        - Pilot ID (string), Key Attribute
            - The unique identification number given to the pilot by the company. No person should have the same Identification.
        - Email (string)
        - Gender (Male, Female, N/A)
        - Birthdate (string, mm/dd/yyyy)
        - SSN (character[9]), Key Attribute
            - The 9 digit characters assigned to the person from the US Government. No person should not have the same SSN.
        - Start date (string, mm/dd/yyyy)
        - Salary (decimal)
        - Flight Hours (decimal)
    - Sub-classes & Attributes
        - Trainee
            - Start date
            - Expected Finish
        - First Officer
            - Year of rank
        - Captain
            - Year of rank

        ………

- DESTINATION entity type
    - Attributes
        - Country (string)
        - City (string)
        - Airport Name (string)
        - IATA Airport Code (Character[3]), Key Attribute
            - The unique 3 letter code assigned to the airports. No airport should have the same code as another airport.

        ………

- ROUTE entity type
    - Attributes
        - Flight Number (string), Key Attribute

        - The unique sequence of characters assigned for the route/flight.
    - Distance (decimal)
    - Departure IATA (Character[3])
        - The departing airport or the start of the route.
    - Arrival IATA (Character[3])
        - The arrival airport or the end of the route.

    ………

- AIRLINE entity type
  - Attributes
    - <u>Name</u>
        - The unique name of the company
    - <u>AirlineID</u>
        - The unique ID that identifies the airline
    - Country
        - The Country that the airline is based at

- Depart_To relationship type
  - An aircraft at a DESTINATION can Depart_To an existing ROUTE.
    - Cardinality is 1:1 (A route is started from the origin destination and can only be one departing origin)

- Arrive_At relationship type
  - An aircraft at an existing ROUTE can Arrive_At a DESTINATION.
    - Cardinality is 1:1 (A unique route can only end at a singular destination. It is possible that the Depart_To may be the same as the Arrive_At)

- Operate_ relationship type
  - a PILOT Operate_ this ROUTE
    - Cardinality is N:M (Multiple Pilots can fly multiple routes)

- Assigned_To relationship type
  - an aircraft is Assigned_To a ROUTE
    - Cardinality is 1:1 (Each aircraft has a unique route)

- Based_at relationship type
  - PILOT is Based_at DESTINATION
    - Cardinality is N:1 (several pilots can be based at an airport)

- Parked_at relationship type
  - PLANE is Based_at DESTINATION

        - Cardinality is N:1 (several planes can be parked at an airport)

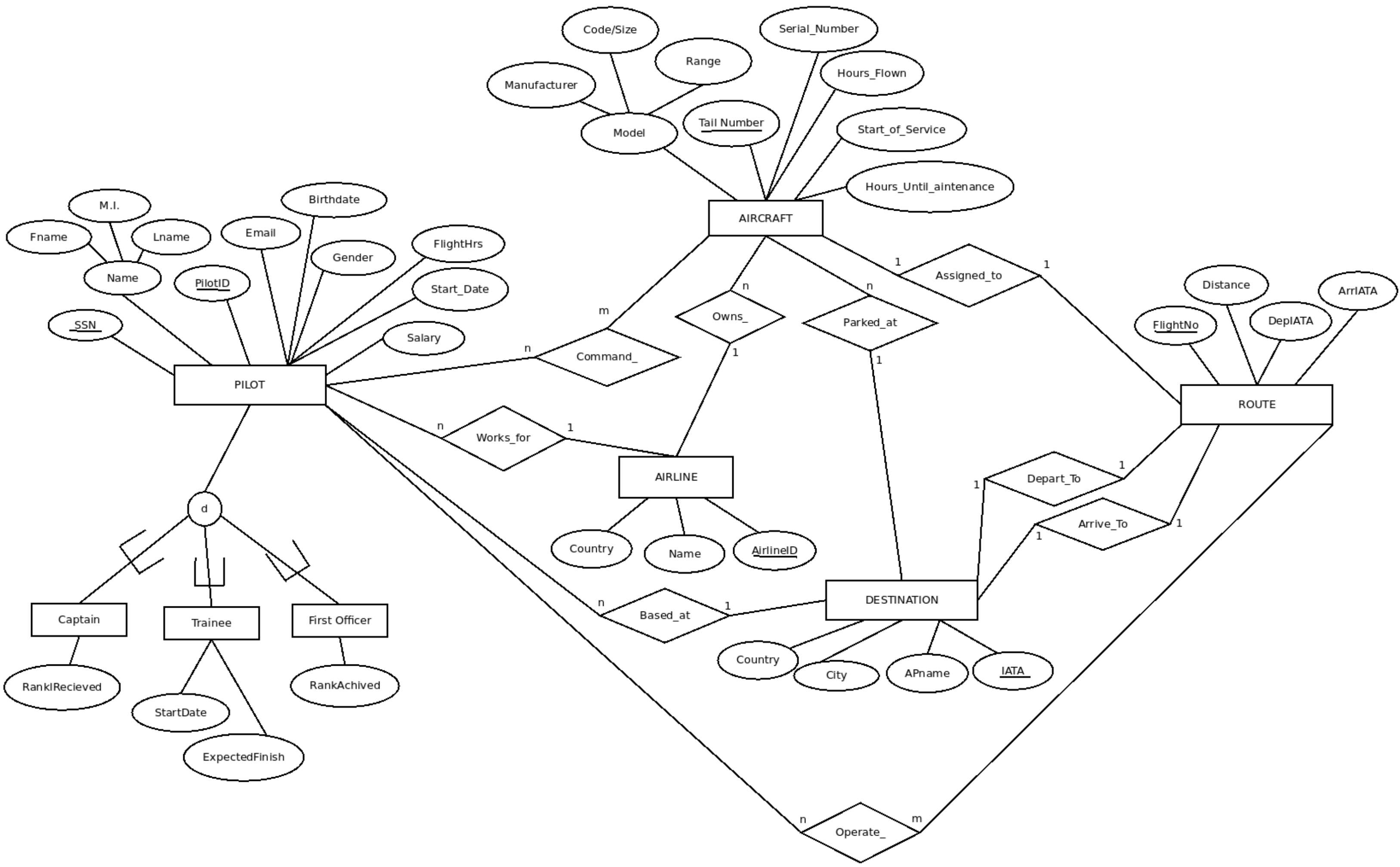
- Copilot_For relationship type
    - PILOT can be a Copliot_For another Pilot
        - Cardinality is N:M (Several pilots may be a copilot to other pilots)

- Command_ relationship type
    - PILOT can command_ an AIRCRAFT
        - Cardinality is N:M (Multiple Pilots may be able to fly multiple aircraft)

- Works_For relationship type
    - PILOT works for a specific AIRLINE
        - Cardinality is N:1 (Multiple pilots may work for a single airline)

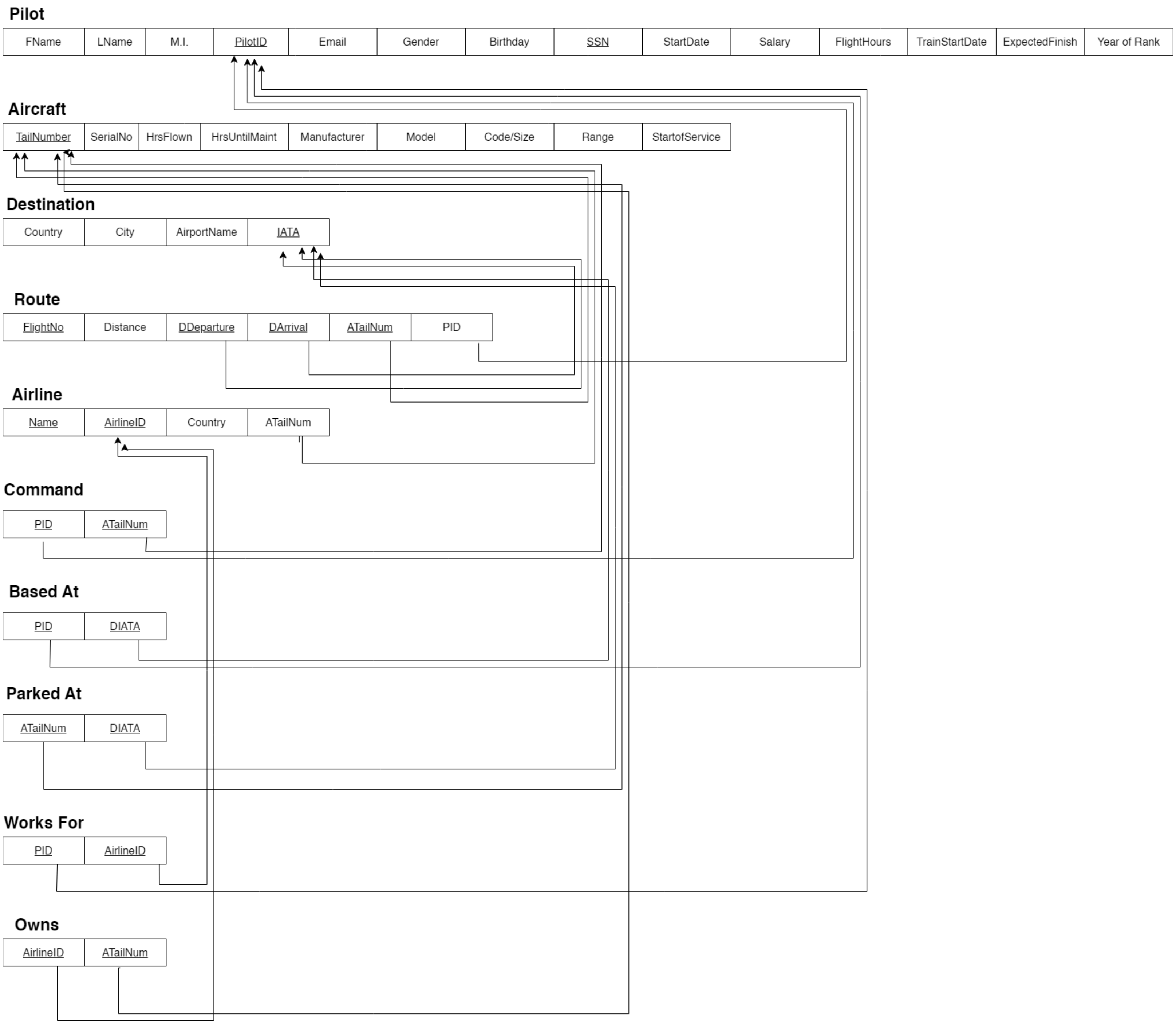
- Owns_ relationship type
    - An AIRLINE can own a AIRCRAFT
        - Cardinality is N:1 (Multiple Aircraft may be owned by a single airline)

EER MODEL



Code/Size
Serial_Number
Range
Hours_Flown
Manufacturer
Model
Tail Number
Start_of_Service
Hours_Until_aintenance
AIRCRAFT
M.I.
Birthdate
Fname
Lname
Email
FlightHrs
Gender
Name
PilotID
Start_Date
SSN
Salary
PILOT
Command_
Owns_
Parked_at
Assigned_to
Distance
ArrIATA
FlightNo
DepIATA
ROUTE
Works_for
AIRLINE
Country
Name
AirlineID
Depart_To
Arrive_To
d
Captain
Trainee
First Officer
RankIRecieved
RankAchived
StartDate
ExpectedFinish
Based_at
DESTINATION
Country
City
APname
IATA
Operate_
m
n
1

RELATIONAL SCHEMA



RELATIONAL ALGEBRA

1. Gather all aircraft that have been flown between 3,000 to 8,000 hours.

➔ π(Model)(σHrsFlown > 3,000 ∧ HrsFlown < 8,000 (AIRCRAFT))

2. Get all the names of pilots that can fly an aircraft currently based in Dallas.

➔ π(Name)(σCity = ‘Dallas’ (PILOT⋈DESTINATION))

3. What aircraft(s) has a range of at least 3,000 nautical miles.

➔ π(Model)(σRange < 3000(AIRCRAFT))

4. Which pilots can command a 787-8 aircraft?

➔ π(Name)(σModel = ‘787’ ∧ Code/Size = ‘8’(PILOT⋈AIRCRAFT))

5. Retrieve all pilots that are female and that fly an Airbus aircraft.

➔ π(Name)(σManufactuer = ‘Airbus’ ∧ Gender = ‘female’(PILOT⋈AIRCRAFT))

6. Which pilots can fly an aircraft that is set for a route from DFW to NRT?

➔ π(Name)(σDDeparture = ‘DArrival’ = ‘NRT’(PILOT⋈ROUTE)))

7. List the number of aircraft that each airline owns.

➔ π(Manufact, COUNT(TailNumber))(σAirline.AlTailNum = Aircraft.TailNumber (AIRLINE⋈AIRCRAFT))

8. What airline does “Nathan Jones” work for?

➔ π(AirName)(σFName = ‘Nathan’ ∧ LName = ‘Jones’(PILOT⋈AIRLINE))

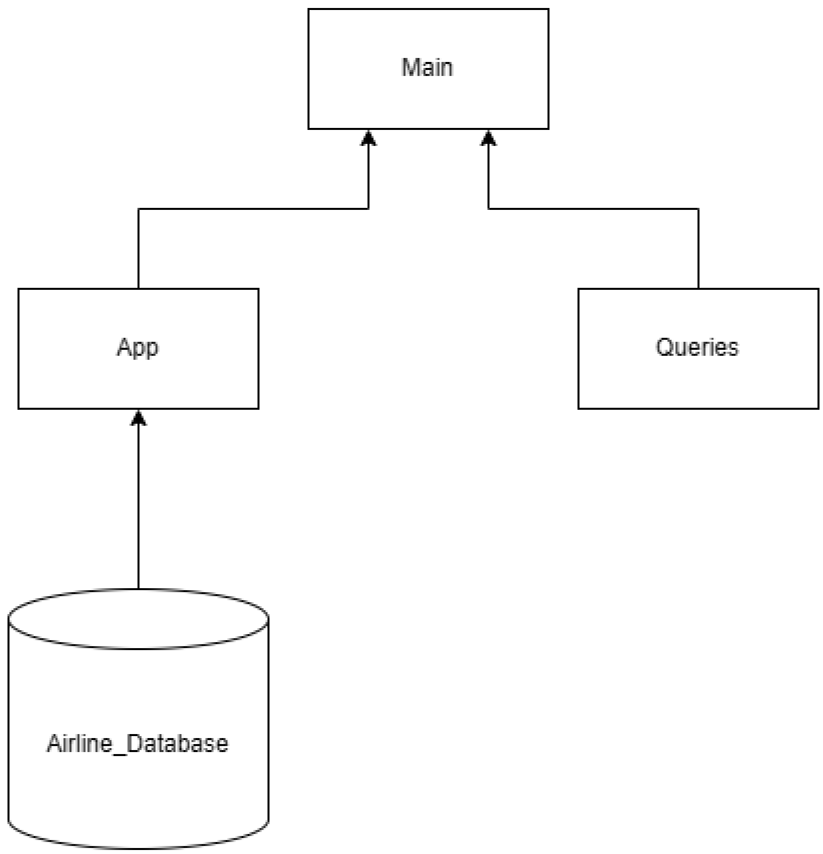
9. List all the A320 Aircraft tail numbers currently in service.

➔ π(TailNumber)(σModel = ‘A320’(AIRCRAFT))

10. Get all the aircraft Tail Numbers that are owned by American Airlines.

➔ π(TailNum)(σAirlName = ‘American Airlines’)(AIRCRAFT⋈AIRLINE))

SYSTEM DIAGRAM



APPLICATION SOURCE USED

- Application Type: Standalone CLI
- Frontend & Backend Language: Java
- DBMS: MySQL https://www.mysql.com/

Final Thoughts and Summary

This project provided a lot of insight into creating and knowing what a database is and how to implement it from start to finish. Knowing how to read and create relational expressions and the algebraic functions really helped with creating the queries and select statements for the database.

One thing that I would do to this project to improve this project than what it is currently is not only being able to connect to a remote database server rather than a local machine like my laptop, but also implement a more abstract way to write the queries rather than having it all be a separate functions that have to be called. It would greatly help with the complexity of the program and also the performance.

Overall this project really made me learn a lot about databases and what it is like to create both the front and back end of the project.