

Cristian Gasparesc 40209205
Dan Bondar 40213095
Gevorg Alaverdyan 40202177
Amirhossein Tavakkoly 40203604

Documentation - SOEN363 Project 1

1. Overview of the system:

This document provides an overview of the database structure and data model used for our project. The objective of the database is used to store and connect data of different plane passengers and their favorite pokemon. We can analyze the database to see if there is any correlation between the type of plane passenger and the choice of pokemon. Our database is composed of various tables representing entities such as Flights, Persons (passengers), Airplanes, Tickets, and Pokémon. The relationships between these tables establish a comprehensive data model to support the system's functionality. For each of these, we take into account different attributes that will help us differentiate passengers from each other and get interesting results when looking at their Pokemon preference.

2. Overview of data model:

Database Schema:

Airplane Table

- SNID (Primary Key): Unique identifier for each airplane.
- Capacity: The maximum seating capacity of the airplane.
- Max Speed: The maximum speed of the airplane.

PrivateJet Table

- SNID (Primary Key, Foreign Key): References the Airplane table.
- Owner Name: The name of the owner of the private jet.

CommercialPlane Table

- SNID (Primary Key, Foreign Key): References the Airplane table.
- Airline Name: The name of the airline operating the commercial plane.

Person Table

- p_id (Primary Key): Unique identifier for each person.
- Passenger Name: The name of the passenger.

Pokemon Table

- poke_id (Primary Key): Unique identifier for each Pokemon.
- Name: The name of the Pokemon.
- Pok Height: Height of the Pokemon.
- Pok Weight: Weight of the Pokemon.
- Pok Base Experience: Base experience of the Pokemon.

FavPokemon Table

- p_id (Foreign Key): References the Person table.
- poke_ID (Foreign Key): References the Pokemon table.

Flight Table

- flight_id (Primary Key): Unique identifier for each flight.
- Departure Location: The location from which the flight departs.
- Arrival Location: The destination of the flight.
- Departure Time: Timestamp indicating the departure time.
- Arrival Time: Timestamp indicating the arrival time.
- SNID (Foreign Key): References the Airplane table.

FlightTicketPerson Table

- seat_num: The seat number.
- flight_id (Foreign Key): References the Flight table.
- p_id (Foreign Key): References the Person table.
- Price: The price of the flight ticket.
- Class: The class of the flight ticket.

3. Approach and challenges:

We faced many challenges while working on this project. Here are some of the issues we faced:

One of the issues we faced was that we couldn't find a specific API that would fit our needs. This issue was resolved by making our own API by combining several other APIs.

Another issue we faced was that instead of writing a script that would convert an API response JSON to an SQL format, we had to manually do it by inserting comma separated values.

Finally, we had issues settling on a project theme. However after a thorough and heated discussion, we managed to find a compromise.