Conceptual Data Modeling

Group 4 - SQL Issue

Chato, Terran Matthew M.

Lim, Hannah Marie L.

Macasaet, James Emmanuelle

Ribo, Alyssa Mari R.

Sevillejo, Jasper Jake

Magis Air

December 2, 2024

Information Management - CSCI 41-F

TABLE OF CONTENTS

1. Discussion of Implementation

The proposed system serves to catalog, track, and organize the various facets of Magis Air's airline operations, providing useful information that can help ensure that efficiency and punctuality is maintained. The system's entities fall into the four categories that were outlined as important to Magis Air's operations: Flight Scheduling, Flight Routes, Flight Booking, and Crew Assignments. Each of these categories, displayed as entity clusters, include the respective entities, as well as their information, that are important for each category of operations to function up to Magis Air's standards.

Flight Scheduling involves a SCHEDULE that concerns itself with much of the planning out and dating of the many FLIGHTs within it. These flights include information on their country of origin, the expected duration of the flight and their cost for flight by itself.

Flight Route mainly deals with the ROUTES that each flight takes, with data on the city it departs from, what time it departs and arrives, and the actual time the flight takes. Additionally, for further data categorization, the DESTINATION of each flight route is kept track of separately, listing the city of arrival and the associated country and airport.

Flight Bookings deal with the many BOOKINGs that can be made, noting when they were booked and their total costs. Additionally, each booking is associated with a PASSENGER. Therefore the system also keeps track of their name, birthdate, and gender for any purposes related to prospective data analysis. Finally, the bookings also have ADDITIONAL ITEMs such as baggage or meals whose descriptions, quantities, and summed costs are also tracked to ensure accurate, and personalized services.

Crew Assignments have to do with CREW PERSONNEL, listing their employee identification, name, phone number, email address, and role within the crew. Additionally, each PLANE is also tracked along with its make and model in order to account for differing safety features or mechanisms.

A few assumptions had to be made for the system. It was assumed that flights could not have multiple origins. In the case of crew assignment, it was also assumed that route details could be accessed through the flight route. Additionally, although not explicitly stated, it was assumed that Magis Air would also like to keep track of booked passengers, as well as the plane used for each flight.

Finally, the system's restrictions are as follows: The system is restricted to one booking per crew member, meaning that it must make sure that they are not double booked. The system also only allows passengers to book flights that exist, so creating flights sufficiently in advance in order to allow passengers time to book is also something that the system does not account for.

Given the task of implementing the system into an actual database and frontend, we chose the platform of Django as we believed that it would give us the greatest degree of control over system implementation, as well as the changes and revisions we would implement before and after the creation of the system for quality and testing purposes, as well as for expansion of the database.

1.1. Software Requirements: operating system, RDBMS, other applications

- OS: Can run on windows 10/11, macOS or Linux for development/deployment.
- RDBMS: MySQL (There is a need to install the mysqlclient library for Django to properly interface with the database).
- Other requirements: Django Framework, Python 3.11+, VS Code(code editor), Git (version control)

1.2. Limitations of the System

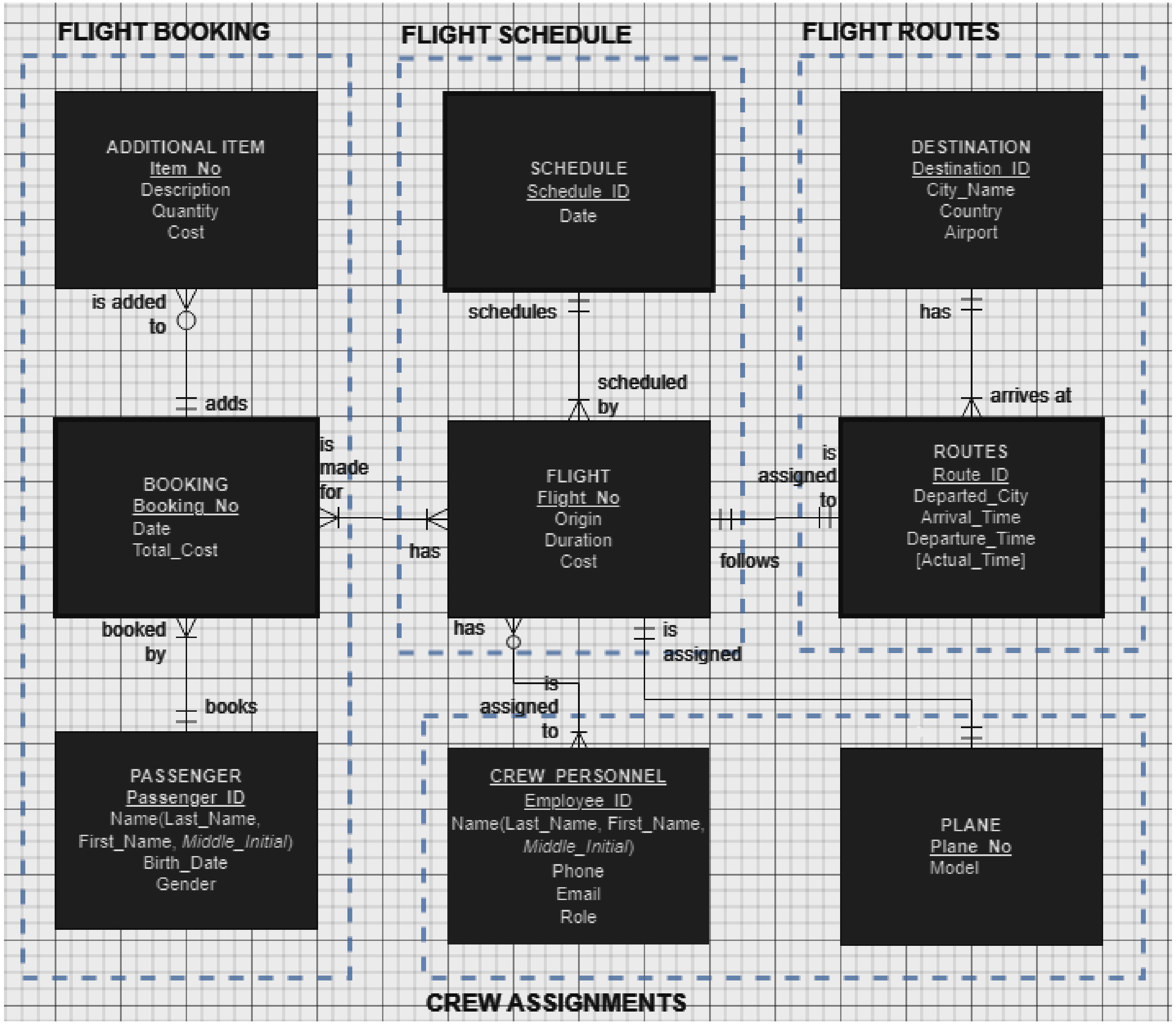
In the case of limitations, we identified two within the system. The first limitation involved the capability to update flight routes, given emergency issues such as changing weather conditions and potentially active warzones. The second limitation is related to this issue, in that flight delays caused by issues such those previously mentioned, as well as mechanical problems, are not considered.

1.3. Problems Encountered during Development

We encountered an issue with how Django handled pks when editing existing records. Django relies on primary keys to uniquely identify database objects. If a primary key, such as schedule_id or flight_no, is modified via a form, Django interprets the change as an instruction to create a new record instead of updating the existing one. This behavior resulted in unintended duplication of objects, with the original record remaining unchanged. Additionally, it should be noted that storing sensitive credentials (e.g., Django SECRET_KEY) in settings.py poses a security risk, so .gitignore will be used for actual implementation of the system for proper usage, and a proper settings.py file will be provided to system administrators and other high level users.

2. Logical Database Design

2.1. Final Conceptual ERD



2.2. Final Data Dictionary

System Title:	Magis Air	Date:	December 2, 2024
Analyzed by:	SQL Issue		

Entity / Relationship Name	Schedule
Entity / Relationship Description	All instances of a flight schedule that have been generated for use by Magis Air.

Attribute Name	Description	Primary Key?	Default Value	Possible Values	Can be Null?
Schedule_ID	The unique ID that identifies each flight schedule.	Y	SCHD0000	String	N
Date	Date the flight schedule was made.	N	05-12-2024	DD-MM-YYYY	N

Entity / Relationship Name	Flight
Entity / Relationship Description	The details for every flight instance.

Attribute Name	Description	Primary Key?	Default Value	Possible Values	Can be Null?
Flight_No.	The unique ID that identifies each flight.	Y	XXXXX-00000	String	N
Origin	The location in which the flight departs from.	N	XXXXX	String	N
Duration	The time the aircraft is expected to take to arrive at the destination from the origin.	N	00:00	HH:MM	N
Cost	The amount the passenger will have to pay for the flight.	N	0.00	Double	N

Entity / Relationship Name	Routes
Entity / Relationship Description	The flightpath of an aircraft to the destination.

Attribute Name	Description	Primary	Default	Possible	Can be

		Key?	Value	Values	Null?
Route_Id	The unique ID that identifies each flight route.	Y	R0000	String	N
Departed _City	The city where the flight departs from.	N	XXXXXX	String	N
Arrival_Time	The time at which the flight route will arrive at the intended destination.	N	12:00 AM	HH:MM AM/PM	N
Departure_Time	The time at which the flight route will depart the origin.	N	12:00 AM	HH:MM AM/PM	N
Actual_Time	The time the aircraft actually takes to arrive at the destination from the origin.	N	00:00	HH:MM	N

Entity / Relationship Name	Destination
Entity / Relationship Description	The specific location of the end-goal of the flight route

Attribute Name	Description	Primary Key?	Default Value	Possible Values	Can be Null?
City_Id	The unique ID that identifies the city.	Y	C000	String	N
City_Name	The name of the city the aircraft will be landing in.	N	XXXX	String	N
Country	The country the destination is located in.	N	XXXX	String	N
Airport	The name of the airport in which the plane will arrive.	N	XXXX	String	N

Entity / Relationship Name	Crew Personnel
Entity / Relationship Description	The details of the crew personnel assigned to that particular flight

Attribute Name	Description	Primary Key?	Default Value	Possible Values	Can be Null?
Employee_ID	The unique identifier of the employee.	Y	E0000	String	N
Name (Last name,	The name of the employee.	N	XXX,	String	N

first name, middle initial)			XXXXX X.		
Phone	The contact number of the employee.	N	+00-0000-000-0000	String	N
Email	The email address of the employee.	N	XXX.XXX@magisair.com	String	Y
Role	The role of the employee.	N	XXXXX	String	N

Entity / Relationship Name	Plane
Entity / Relationship Description	The plane assigned to the flight

Attribute Name	Description	Primary Key?	Default Value	Possible Values	Can be Null?
Plane_No	The unique identifier of the aircraft.	Y	ABC-000	String	N
Model	The make and model of the aircraft.	N	XXXXX X-000	String	N

Entity / Relationship Name	Booking
Entity / Relationship Description	The details for a flight booking, tracking a passenger's flights and luggage.

Attribute Name	Description	Primary Key?	Default Value	Possible Values	Can be Null?
Booking_No	The unique identifier for each booking.	Y	B000	String	N
Date	The date that the booking was accomplished.	N	05-12-2024	DD-MM-YYYY	N
Total_Cost	The total monetary cost of the booking.	N	0.00	Double	N

Entity / Relationship Name	Passenger
Entity / Relationship Description	The passenger and assigned to be on booked flights, and their

	related personal information.

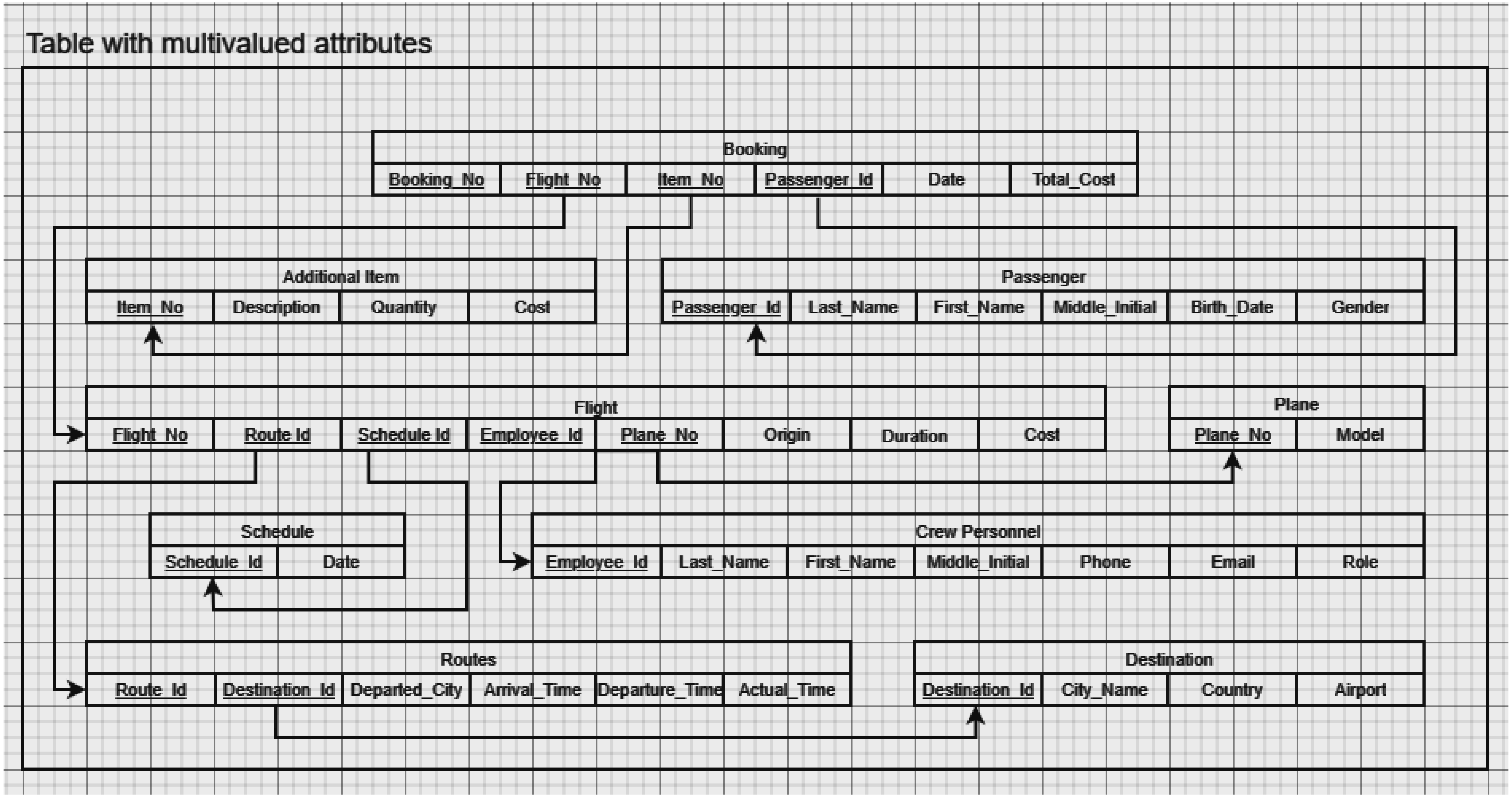
Attribute Name	Description	Primary Key?	Default Value	Possible Values	Can be Null?
Passenger_ID	The unique identifier for each passenger.	Y	P0000	String	N
Name(Last_Name, First_Name, *Middle_Initial*)	The name of each passenger	N	XXX, XXXX, X.	String	N
Birth_Date	The passenger's birth date.	N	2024-12-05	YYYY-MM-DD	N
Gender	The passenger's gender.	N	XXXX	String	N

Entity / Relationship Name	Additional Item
Entity / Relationship Description	The cost of an additional item that is not included in the standard booking cost, to be added to the final booking cost.

Attribute Name	Description	Primary Key?	Default Value	Possible Values	Can be Null?
Item_No	The unique number assigned to each additional item.	Y	0000	Integer	N
Description	The description of each additional item.	N	XXXXX	String	N
Quantity	How many of the additional items are being paid for.	N	1	Integer	N
Cost	The cost of each individual item, multiplied by its quantity.	N	0.00	Double	N

2.3. Transformation of Conceptual ERD to Logical ERD with Normalized Relations

2.3.1. Transform the final conceptual ERD to the logical model.



Booking

Booking_No	Flight_No	Item_No	Passenger_Id	Date	Total_Cost

Additional Item

Item_No	Description	Quantity	Cost

Passenger

Passenger_Id	Last_Name	First_Name	Middle_Initial	Birth_Date	Gender

Flight

Flight_No	Route Id	Schedule Id	Employee_Id	Plane_No	Origin	Duration	Cost

Plane

Plane_No	Model

Schedule

Schedule_Id	Date

Crew Personnel

Employee_Id	Last_Name	First_Name	Middle_Initial	Phone	Email	Role

Routes

Route_Id	Destination_Id	Departed_City	Arrival_Time	Departure_Time	Actual_Time

Destination

Destination_Id	City_Name	Country	Airport

2.3.2. Normalize the models.

Identify the Normal Forms and show how the entities were transformed from one Normal Form to another by showing the functional and transitive dependencies and how these were eliminated. Normalize up to 3NF. Show any other steps that occurred in the process. In the final normalized relations, underline primary keys and show foreign keys with broken (or alternating) underlines. Example: EMPLOYEE (Emp_ID, Name, Bday, Manager_ID)

Base

BOOKING(Booking_No., Flight_No., Item_No., Passenger_ID, Date, Total_Cost)
ADDITIONAL_ITEM(Item_No., Description, Quantity, Cost)
PASSENGER(Passenger_ID, Last_Name, First_Name, Middle_Initial, Birth_Date, Gender)
FLIGHT(Flight_No., Route_ID, Schedule_ID, Employee_ID, Plane_No., Origin, Duration, Cost)
PLANE(Plane_No., Model)
SCHEDULE(Schedule_ID, Date)
CREW PERSONNEL(Employee_ID, Last_Name, First_Name, Middle_Initial, Phone, Email, Role)
ROUTES(Route_ID, Destination_ID, Departed_City, Arrival_Time, Departure_Time, Actual_Time)
DESTINATION(Destination_ID, City_Name, Country, Airport)

- Multivalued attributes
  - BOOKING can have several Item_No.'s
  - FLIGHT can have several Employee_ID's

1NF

BOOKING(Booking_No., Flight_No., Billing_ID, Passenger_ID, Date, Total_Cost)
→ ITEM_BILL(Billing_ID, Item_No.)
ADDITIONAL_ITEM(Item_No., Description, Quantity, Cost)
PASSENGER(Passenger_ID, Last_Name, First_Name, Middle_Initial, Birth_Date, Gender)
FLIGHT(Flight_No., Route_ID, Schedule_ID, Crew_ID, Plane_No., Origin, Duration, Cost)
→ FLIGHT CREW(Crew_ID, Employee_ID)
PLANE(Plane_No., Model)
SCHEDULE(Schedule_ID, Date)
CREW PERSONNEL(Employee_ID, Last_Name, First_Name, Middle_Initial, Phone, Email, Role)
ROUTES(Route_ID, Destination_ID, Departed_City, Arrival_Time, Departure_Time, Actual_Time)
DESTINATION(Destination_ID, City_Name, Country, Airport)

- Aside from the multivalued attributes, we have no additional dependencies, meaning that our table is in 3F.

3F

BOOKING					
Booking No.	Flight_No.	Billing_ID	Passenger_ID	Date	Total_Cost

ITEM_BILL	
Billing_ID	Item_No.

ADDITIONAL ITEM			
Item_No.	Description	Quantity	Cost

PASSENGER					
Passenger ID	Last_Name.	First_Name	Middle_Initial	Birth_Date	Gender

FLIGHT							
Flight No.	Route_ID	Schedule_ID	Crew_ID	Plane No.	Origin	Duration	Cost

FLIGHT CREW	
Crew ID	Employee_ID

PLANE	
Plane_No.	Model

SCHEDULE	
Schedule_ID	Date

CREW PERSONNEL						
Employee_Id	Last_Name	First_Name	Middle_Initial	Phone	Email	Role

ROUTES					
Route Id	Destination_ID	Departed_City	Arrival_Time	Departure_Time	Actual_Time

DESTINATION			
Destination ID	City_Name	Country	Airport

3. Actual Implementation

3.1. Tables and Integrity Constraints

```
CREATE TABLE schedule (
  schedule_id VARCHAR(20) NOT NULL PRIMARY KEY,
  date DATE NOT NULL DEFAULT '2024-12-05'
);

CREATE TABLE flight (
  flight_no VARCHAR(20) NOT NULL PRIMARY KEY,
  origin VARCHAR(100) NOT NULL DEFAULT 'XXXXX',
  duration TIME NOT NULL DEFAULT '00:00:00',
  cost DECIMAL(10, 2) NOT NULL DEFAULT 0.00,
  schedule_id VARCHAR(20) NOT NULL,
  FOREIGN KEY (schedule_id) REFERENCES schedule(schedule_id)
);

CREATE TABLE route (
  route_id VARCHAR(20) NOT NULL PRIMARY KEY,
  departed_city VARCHAR(100) NOT NULL,
  destination_id VARCHAR(20) NOT NULL,
  departure_time TIME NOT NULL,
  arrival_time TIME NOT NULL,
  expected_travel_time TIME NOT NULL,
  FOREIGN KEY (destination_id) REFERENCES destination(destination_id)
);

CREATE TABLE destination (
  destination_id VARCHAR(20) NOT NULL PRIMARY KEY,
  city_name VARCHAR(100) NOT NULL,
  country VARCHAR(100) NOT NULL,
  airport VARCHAR(100) NOT NULL
);

CREATE TABLE crew_personnel (
  employee_id VARCHAR(20) NOT NULL PRIMARY KEY,
  last_name VARCHAR(50) NOT NULL,
  first_name VARCHAR(50) NOT NULL,
  middle_initial VARCHAR(5),
  phone VARCHAR(15) NOT NULL,
  email VARCHAR(100),
  role VARCHAR(50) NOT NULL
);

CREATE TABLE flight_crew (
  crew_id VARCHAR(20) NOT NULL PRIMARY KEY,
  employee_id VARCHAR(20) NOT NULL,
  FOREIGN KEY (employee_id) REFERENCES crew_personnel(employee_id)
);
```

```
CREATE TABLE plane (
   plane_no VARCHAR(20) NOT NULL PRIMARY KEY,
   model VARCHAR(100) NOT NULL
);

CREATE TABLE flight (
   flight_no VARCHAR(20) NOT NULL PRIMARY KEY,
   origin VARCHAR(100) NOT NULL DEFAULT 'XXXXX',
   duration TIME NOT NULL DEFAULT '00:00:00',
   cost DECIMAL(10, 2) NOT NULL DEFAULT 0.00,
   schedule_id VARCHAR(20) NOT NULL,
   route_id VARCHAR(20) NOT NULL,
   plane_no VARCHAR(20) NOT NULL,
   FOREIGN KEY (schedule_id) REFERENCES schedule(schedule_id),
   FOREIGN KEY (route_id) REFERENCES route(route_id),
   FOREIGN KEY (plane_no) REFERENCES plane(plane_no)
);

CREATE TABLE passenger (
   passenger_id VARCHAR(20) NOT NULL PRIMARY KEY,
   last_name VARCHAR(50) NOT NULL,
   first_name VARCHAR(50) NOT NULL,
   middle_initial VARCHAR(5),
   birth_date DATE NOT NULL,
   gender VARCHAR(10) NOT NULL
);

CREATE TABLE additional_item (
   item_no VARCHAR(20) NOT NULL PRIMARY KEY,
   description VARCHAR(255) NOT NULL,
   quantity INT NOT NULL DEFAULT 1,
   cost DECIMAL(10, 2) NOT NULL DEFAULT 0.00
);

CREATE TABLE item_billing (
   billing_id VARCHAR(20) NOT NULL PRIMARY KEY,
   item_no VARCHAR(20) NOT NULL,
   FOREIGN KEY (item_no) REFERENCES additional_item(item_no)
);

CREATE TABLE booking (
   booking_no VARCHAR(20) NOT NULL PRIMARY KEY,
   flight_no VARCHAR(20) NOT NULL,
   item_billing_id VARCHAR(20) NOT NULL,
   passenger_id VARCHAR(20) NOT NULL,
   date DATE NOT NULL,
   total_cost DECIMAL(10, 2) NOT NULL DEFAULT 0.00,
   FOREIGN KEY (flight_no) REFERENCES flight(flight_no),
   FOREIGN KEY (item_billing_id) REFERENCES item_billing(billing_id),
   FOREIGN KEY (passenger_id) REFERENCES passenger(passenger_id)
);
```

3.2. Sample Queries

minimum of five SELECT statements used in the actual system

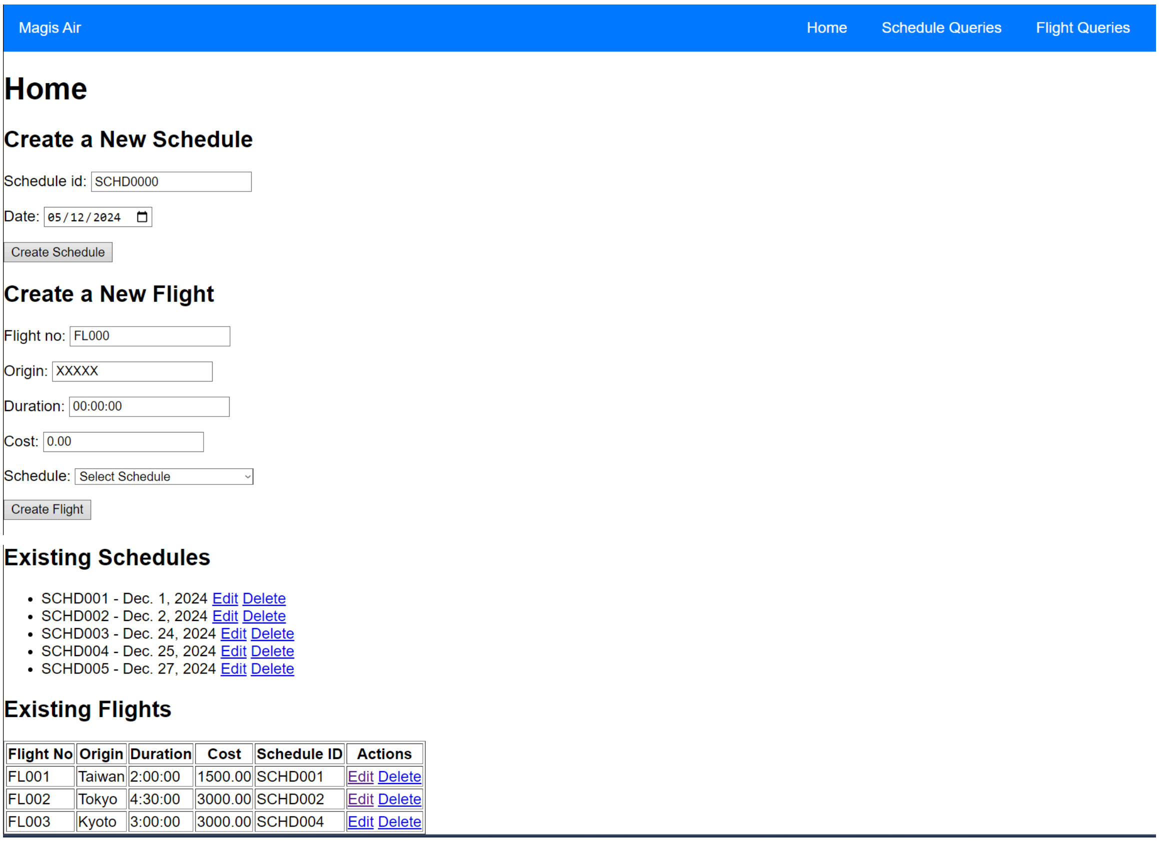
```
SELECT * FROM schedule;

SELECT * FROM flight;

SELECT * FROM schedule WHERE schedule_id = SCHD0001;

SELECT * FROM flight WHERE origin = 'Kyoto';

SELECT * FROM flight WHERE cost = 500.00;
```

3.3. Final Set of Screens and Reports

These are the ones to be shown during the presentation, with individual descriptions.

Homepage View:

In this page, the user can create flights and schedules which will be displayed and/or added to the existing database. Existing items may also either be edited or deleted, which will bring them to different views for both.



Home

Create a New Schedule

Schedule id: SCHD0000

Date: 05/12/2024

Create a New Flight

Flight no: FL000

Origin: XXXXX

Duration: 00:00:00

Cost: 0.00

Schedule: Select Schedule

Existing Schedules

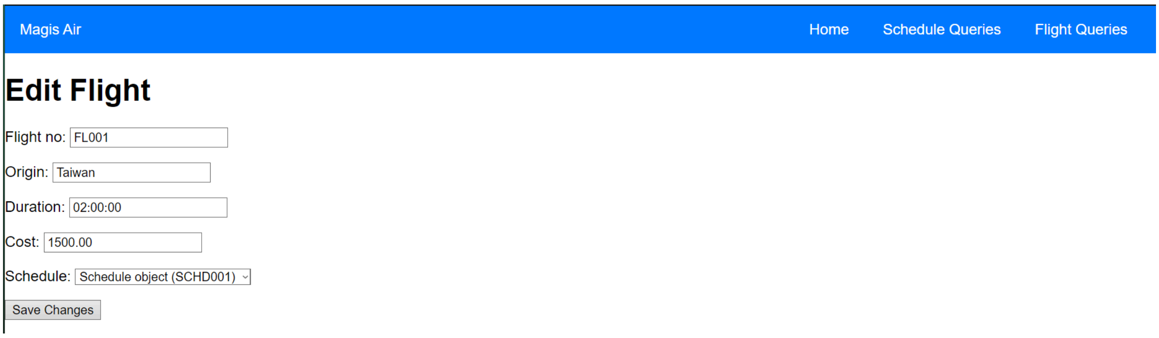
- SCHD001 - Dec. 1, 2024 Edit Delete
- SCHD002 - Dec. 2, 2024 Edit Delete
- SCHD003 - Dec. 24, 2024 Edit Delete
- SCHD004 - Dec. 25, 2024 Edit Delete
- SCHD005 - Dec. 27, 2024 Edit Delete

Existing Flights

Flight No	Origin	Duration	Cost	Schedule ID	Actions
FL001	Taiwan	2:00:00	1500.00	SCHD001	Edit Delete
FL002	Tokyo	4:30:00	3000.00	SCHD002	Edit Delete
FL003	Kyoto	3:00:00	3000.00	SCHD004	Edit Delete

Edit View:

This page allows the user to edit flights which includes the flight number, country of origin, total duration, the cost, and schedule of the particular flight.



Edit Flight

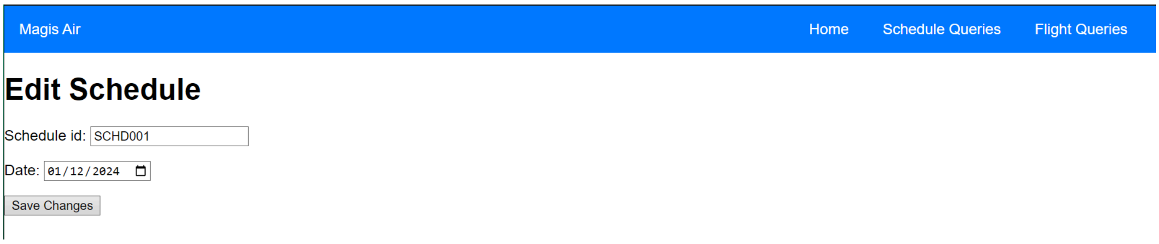
Flight no: FL001

Origin: Taiwan

Duration: 02:00:00

Cost: 1500.00

Schedule: Schedule object (SCHD001)



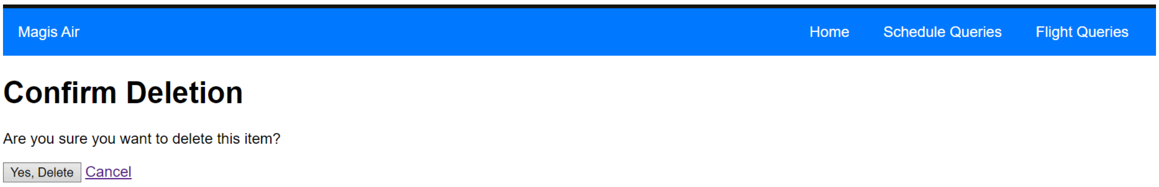
Edit Schedule

Schedule id: SCHD001

Date: 01/12/2024

Delete View:

This view will be used when the user wants to delete certain items or flights in the database. It will display a confirmation message to assure that the action will be permanently executed, preventing accidental deletions, and ensuring that the user is fully aware of the immediate effect of this action.

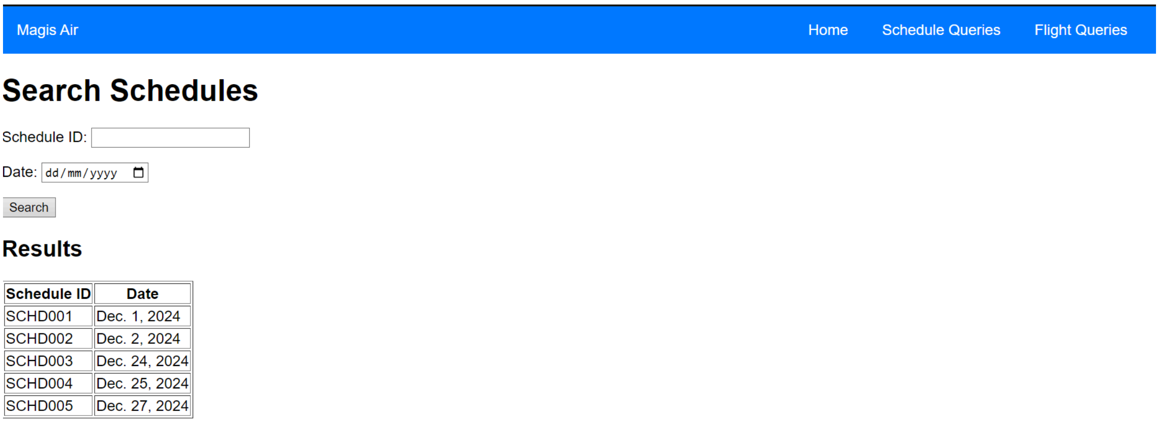


Confirm Deletion

Are you sure you want to delete this item?

Query View:

The query view allows users to retrieve and display specific information and access various data points from the database based on their search criteria.



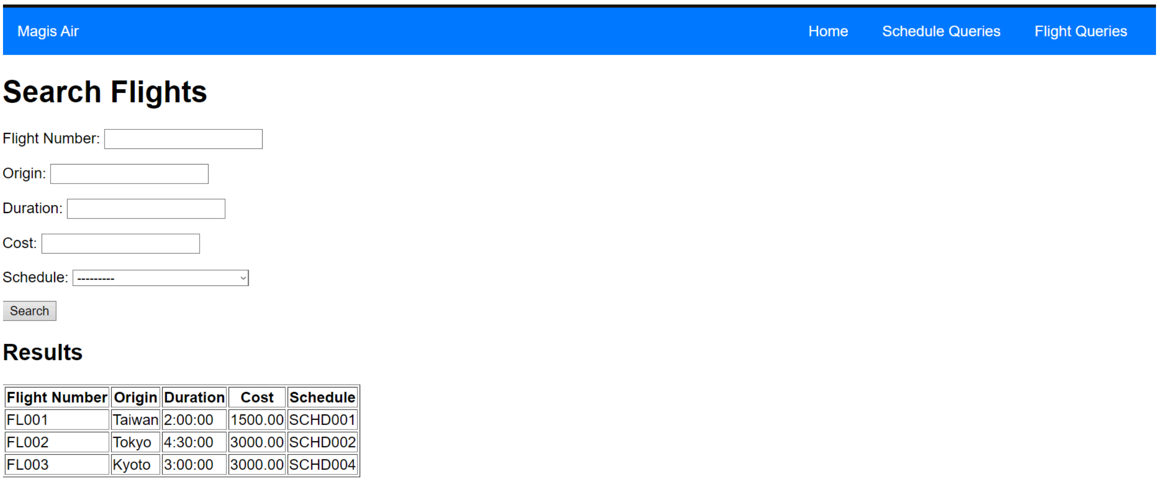
Search Schedules

Schedule ID:

Date: dd/mm/yyyy

Results

Schedule ID	Date
SCHD001	Dec. 1, 2024
SCHD002	Dec. 2, 2024
SCHD003	Dec. 24, 2024
SCHD004	Dec. 25, 2024
SCHD005	Dec. 27, 2024



Search Flights

Flight Number:

Origin:

Duration:

Cost:

Schedule: ---------

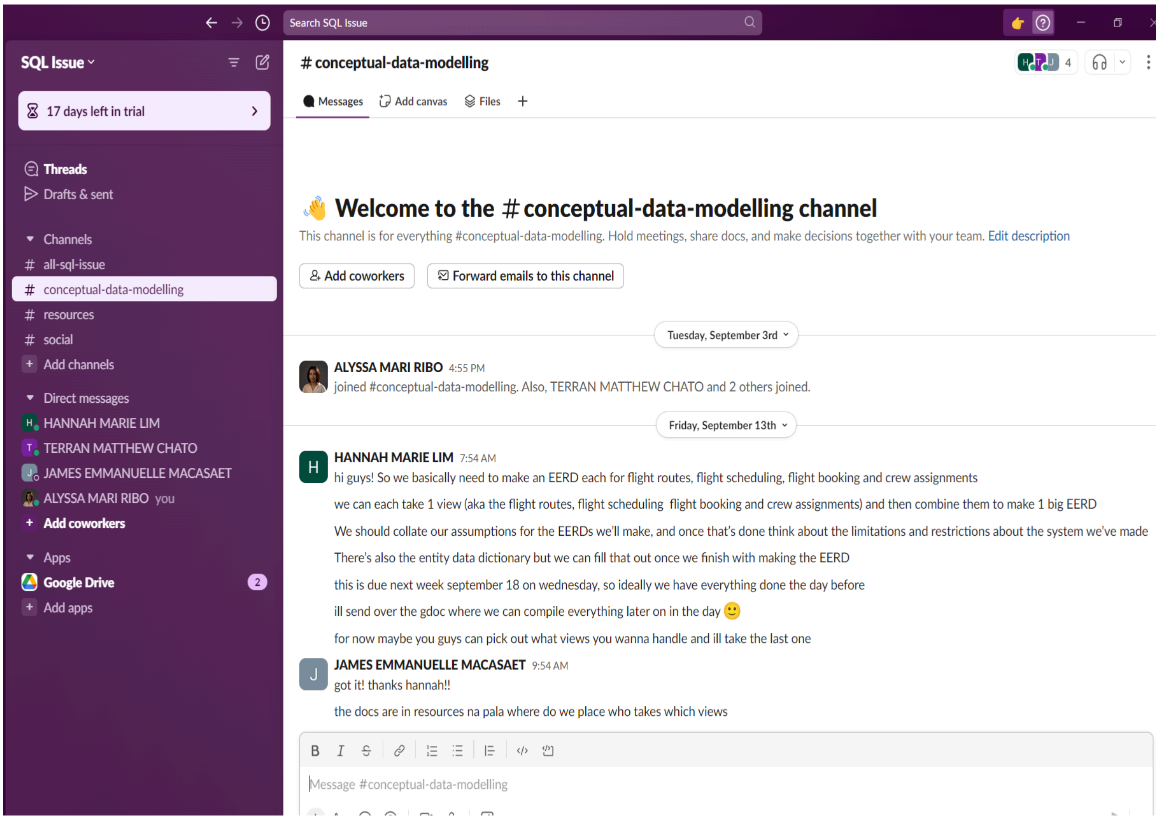
Search

Results

Flight Number	Origin	Duration	Cost	Schedule
FL001	Taiwan	2:00:00	1500.00	SCHD001
FL002	Tokyo	4:30:00	3000.00	SCHD002
FL003	Kyoto	3:00:00	3000.00	SCHD004

4. Appendix A:

Project Management Platform : Slack



Welcome to the # conceptual-data-modelling channel

This channel is for everything #conceptual-data-modelling. Hold meetings, share docs, and make decisions together with your team. Edit description

Tuesday, September 3rd

ALYSSA MARI RIBO 4:55 PM
joined #conceptual-data-modelling. Also, TERRAN MATTHEW CHATO and 2 others joined.

Friday, September 13th

HANNAH MARIE LIM 7:54 AM
hi guys! So we basically need to make an EERD each for flight routes, flight scheduling, flight booking and crew assignments

we can each take 1 view (aka the flight routes, flight scheduling flight booking and crew assignments) and then combine them to make 1 big EERD

We should collate our assumptions for the EERDs we'll make, and once that's done think about the limitations and restrictions about the system we've made

There's also the entity data dictionary but we can fill that out once we finish with making the EERD

this is due next week september 18 on wednesday, so ideally we have everything done the day before

ill send over the gdoc where we can compile everything later on in the day 🙂

for now maybe you guys can pick out what views you wanna handle and ill take the last one

JAMES EMMANUELLE MACASAET 9:54 AM
got it! thanks hannah!!

the docs are in resources na pala where do we place who takes which views

5. Appendix B:

Ateneo de Manila University
Department of Information Systems and Computer Science

CERTIFICATE OF AUTHORSHIP

Instructions

- Download and fill this PDF form completely.
- Each course requirement submission, unless otherwise specified by the Course Instructor, whether in electronic or paper form, must be accompanied by a corresponding properly accomplished Certificate of Authorship.

Description of Submission

Title of Submission: Project Deliverable 2

Type of Submission: □ Program ☒ Project □ Report □ Paper
□ Other (specify) ________

Date of Submission: February 12, 2024

Certification

We hereby certify that the submission described in this document abides by the principles stipulated in the DISCS Academic Integrity Policy document. We further certify that we are the authors of this submission and that any assistance we received in its preparation is fully acknowledged and disclosed in the documentation. We have also cited all sources from which we obtained data, ideas, or words that are directly copied or paraphrased in this document. Sources are properly credited according to accepted standards for professional publication.

- https://docs.djangoproject.com/en/5.1/ref/forms/fields/
- https://docs.djangoproject.com/en/5.1/ref/forms/widgets/
- https://docs.djangoproject.com/en/5.1/topics/forms/
- https://docs.djangoproject.com/en/5.1/topics/db/queries/
- CSCI 41 3b 3a Logical Design (from EERDs) slides
- CSCI 41 3b Logical Design (Normalization) slides

Declaration of Use of Generative AI

Tool: N/A

Purpose: N/A

We have reviewed and revised the content as we see fit. We take full responsibility for the content and ownership of the submitted / published work.

Group Information

Full Name	Signature
Terran Matthew Marcelo Chato	
Hannah Marie Lim	
James Emmanuelle Macasaet	James Macasaet

Course Code & Section
Course Title
Course Instructor