SuperTravel Scenario Project Report

Group 7

Boyu Wang, Caisen Mo, Qinyun Zhang, Zhenyu Wang

APAN-5310: Final Project Report Submission

Nikolaos Machairas

Columbia University

August 8, 2023

Problem Statement

When individuals make the decision to embark on a journey, there are three main priorities that the vast majority tend to focus on. First and foremost is securing the airfare for the trip. Booking airfare is typically one of the initial steps in trip planning, as it determines the destination, as well as the trip's start and end dates. Once this is accomplished, the next consideration revolves around arranging suitable accommodation. Many travelers find selecting and booking a hotel to be a significant concern. Lastly, there is the matter of transportation during the journey. Increasingly, people are opting to rent cars for travel due to the convenience and punctuality it offers.

However, what seems like three simple aspects of travel - airfare, hotel, and car rental - have now become quite complex for people. An abundance of information is scattered across various websites and platforms. As a result, travelers find themselves needing to utilize multiple platforms to compare different options and make the most informed choices. Similarly, airlines, hotels, and car rental companies also face the challenge of managing the large volume of booking information from tourists while conducting their day-to-day operations.

Our team members have gained firsthand experience with the intricacies and challenges of this process in real-life scenarios. When it comes to booking flights, we typically choose between purchasing tickets from the airline's official website or utilizing platforms like StudentUniverse. However, we have observed that despite StudentUniverse advertising the best deals for students, the airline's official website often offers more competitive prices. Similarly, the abundance of hotel booking platforms, such as Expedia and Bookings, leads to substantial price variations for the same hotel, necessitating repetitive searches to secure the most advantageous rates. Renting a car adds another layer of complexity, involving multiple rental companies and numerous considerations, including vehicle type, condition, and rental costs. The multitude of factors to weigh makes the car rental process particularly intricate and time-consuming.

Therefore, SuperTravel, our client’s requirements entail the development of an integrated travel service platform that eliminates the need to use multiple websites for travel arrangements. Robust data management capabilities are essential to handle the large volume of booking information securely and data analytics and insights will enable continuous improvement of the platform's offerings and user experience.

Proposal

We hold a strong conviction that there is ample room for improvement in the current scenario. Let us consider the example of Ctrip, a prominent Chinese travel service software that integrates three essential services: airfare purchases, hotel bookings, and vehicle rentals. This platform

empowers users to consolidate their entire travel itinerary onto a unified and efficient interface. A pivotal factor contributing to Ctrip's success is its strategic collaboration with a diverse array of travel service providers, facilitating an extensive and diverse selection of choices within each service category. This strategic partnership enables users to effortlessly compare and evaluate different options, enabling more expedient and well-informed decision-making for optimizing their travel preferences. By offering such a comprehensive and all-encompassing platform, Ctrip effectively streamlines the travel planning process, obviating the need for users to engage in time-consuming searches and comparisons across multiple websites. The result is a considerable time-saving advantage, coupled with a superior user experience that ensures optimal satisfaction and convenience for travelers.

Therefore, we firmly believe that SuperTravel is poised to revolutionize the vacation planning industry and is going to be the next big thing in the market. SuperTravel sets itself apart by providing an all-in-one solution for travelers. The platform enables users to access a wide range of flight options, car rentals, and hotel accommodations, all within a single, unified platform. This seamless and convenient booking experience saves travelers time and effort, streamlining the entire planning process and eliminating the need to visit multiple websites. Meanwhile, the platform's cutting-edge recommendation website utilizes advanced algorithms and data analytics to seek out the best deals on flights, hotels, and car rentals. This innovative approach promises customers highly personalized and cost-effective travel options, making their vacation planning experience not only convenient but also economical.

Team Structure and Timeline

The team contract outlines the responsibilities and timelines for each team member in the development of the travel service platform's database and user interface. Starting on July 20, Caisen Mo is tasked with gathering flight-related, car rental services, and hotel-related information for the database. On July 21, Zhenyu Wang is assigned to perform data cleaning, ensuring that inconsistencies, duplicates, and errors are removed from the collected data. On July 22, Boyu Wang, Zhenyu Wang, Caisen Mo, and Qinyun Zhang collaborated to normalize the database structure using optimization techniques. On July 23, the team continued their joint efforts, as Boyu Wang, Zhenyu Wang, Caisen Mo, and Qinyun Zhang worked together to create an Entity-Relationship (ER) diagram, visually illustrating the database structure. Simultaneously, they implement the planned design and create the database. On July 25, Qinyun Zhang took on the task of data transformation and import, developing a strategy for loading the cleaned and structured data into the database system. The subsequent stage, set for July 26, involves Boyu Wang and Caisen Mo's collaboration in designing a user-friendly interface that facilitates easy interaction with the database. Finally, on August 5, Zhenyu Wang, Qinyun Zhang, Caisen Mo, and Boyu Wang combined their efforts to compile all project details, methodologies, and outcomes into a comprehensive final report.

By adhering to this contract and timeline, the team aims to create a successful and efficient travel service platform, catering to both travelers and travel service providers.

Database Sample

In order to better build our own comprehensive database schema, we found three databases online covering three different aspects including flight, hotel, and car rental. Here are the links to those full datasets. To improve the clarity of the data presentation in this report, we have selectively extracted a portion of the database as an illustrative example for inclusion.

1. Flight: https://www.kaggle.com/datasets/deepankurk/flight-take-off-data-jfk-airport

MONTH	DAY_OF_MONTH	DAY_OF_WEEK	OP_UNIQUE_CARRI	TAIL_NUM	DEST	DEP_DELAY	CRS_ELAPSED_TIN	DISTANCE	CRS_DEP_M	DEP_TIME_M	CRS_ARR_M
11	1	5	B6	N828JB	CHS	-1	124	636	324	323	448
11	1	5	B6	N992JB	LAX	-7	371	2475	340	333	531
11	1	5	B6	N959JB	FLL	40	181	1069	301	341	482
11	1	5	B6	N999JQ	MCO	-2	168	944	345	343	513
11	1	5	DL	N880DN	ATL	-4	139	760	360	356	499

Temperature	Dew Point	Humidity	Wind	Wind Speed	Wind Gust	Pressure	Condition	sch_dep	sch_arr	TAXI_OUT
48	34	58	W	25	38	29.86	Fair / Windy	9	17	14
48	34	58	W	25	38	29.86	Fair / Windy	9	17	15
48	34	58	W	25	38	29.86	Fair / Windy	9	17	22
48	34	58	W	25	38	29.86	Fair / Windy	9	17	12
46	32	58	W	24	35	29.91	Fair / Windy	9	17	13

2. Hotel: https://www.kaggle.com/datasets/mojtaba142/hotel-booking

hotel	is_canceled	lead_time	arrival_date_year	arrival_date_month	arrival_date_week_n	arrival_date_day_of_	stays_in_weekend_n	stays_in_week_night
Resort Hotel	0	342	2015	July	27	1	0	0
Resort Hotel	0	737	2015	July	27	1	0	0
Resort Hotel	0	7	2015	July	27	1	0	1
Resort Hotel	0	13	2015	July	27	1	0	1
Resort Hotel	0	14	2015	July	27	1	0	2

adults	children	babies	meal	country	market_segment	distribution_channel	is_repeated_guest	previous_cancellatic	previous_bookings_n	reserved_room_type
2	0	0	BB	PRT	Direct	Direct	0	0	0	C
2	0	0	BB	PRT	Direct	Direct	0	0	0	C
1	0	0	BB	GBR	Direct	Direct	0	0	0	A
1	0	0	BB	GBR	Corporate	Corporate	0	0	0	A
2	0	0	BB	GBR	Online TA	TA/TO	0	0	0	A

3. Car rental: https://www.kaggle.com/datasets/kushleshkumar/cornell-car-rental-dataset?resource=download

fuelType	rating	renterTripsTaken	reviewCount	location.city	location.country	location.latitude	location.longitude
ELECTRIC	5	13	12	Seattle	US	47.449107	-122.308841
ELECTRIC	5	2	1	Tijeras	US	35.11106	-106.276551
HYBRID	4.92	28	24	Albuquerque	US	35.127163	-106.566681
GASOLINE	5	21	20	Albuquerque	US	35.149726	-106.711425
GASOLINE	5	3	1	Albuquerque	US	35.208659	-106.601008

location.state	owner.id	rate.daily	vehicle.make	vehicle.model	vehicle.type	vehicle.year
WA	12847615	135	Tesla	Model X	suv	2019
NM	15621242	190	Tesla	Model X	suv	2018
NM	10199256	35	Toyota	Prius	car	2012
NM	9365496	75	Ford	Mustang	car	2018
NM	3553565	47	Chrysler	Sebring	car	2010

We have chosen these three datasets that revolve around flights, hotels, and car rentals as the foundation for building our own database schema. The decision to incorporate these datasets stems from several compelling reasons. First and foremost, the combination of flights, hotels, and car rentals presents us with a comprehensive set of data related to the travel industry. These datasets cover crucial elements of travel arrangements, allowing us to establish a database that addresses various facets of the travel domain. Moreover, the interconnected nature of flights, hotels, and car rentals makes them logical choices for inclusion in our database schema. Travelers often book these services together, creating a seamless travel experience. By integrating these datasets, we can establish meaningful relationships between flights, hotels, and car rentals, enabling us to analyze and optimize travel itineraries more effectively.

Normalization Plan and Database Schema

First, we need to identify entities and attributes. In this step, we carefully analyzed the datasets covering flights, hotels, and car rentals to identify the entities and their attributes. For example, from the flights dataset, we identified entities like airlines and flights with attributes such as airline_id, carrier_name, flight_id, tail_num, etc. Similarly, from the hotels dataset, we identified entities like hotels and hotel bookings with attributes like hotel_id, hotel_type, booking_id, arrival_date, etc. From the car rentals dataset, we identified entities like renters, vehicles, rentals, and car rental reservations with their respective attributes.

Second, we need to define primary keys for each entity identified in step above, we have determined a primary key that uniquely identifies each record in the table. The primary key is crucial for data integrity and uniqueness. For example, in the airlines table, the airline_id will be the primary key, in the flights table, the flight_id will be the primary key, and so on for other entities.

Next, we have eliminated the repeating groups. In this step, we will ensure that each attribute contains atomic values and does not contain repeating groups. If there are attributes with repeating groups, we have created separate tables to represent those entities. For example, a hotel can have multiple room types. We have created a separate room_type table with room_type_id as the primary key and store the room types there. This step helps in reducing data redundancy and improving data integrity.

Fourth, we have checked for partial dependencies, where a non-key attribute depends on only part of the primary key. If we find any partial dependencies, we will move the dependent attribute to a new table. This ensures that each table represents a single, self-contained entity. For example, if a hotel_booking table has hotel_id and room_type_id as the primary key, but is_canceled depends only on room_type_id, we will move is_canceled to a new table with

room_type_id as the primary key. This step further improves data organization and reduces data anomalies.

In the next section, we checked for transitive dependencies, where a non-key attribute depends on another non-key attribute. If we find any transitive dependencies, we will move the dependent attribute to a new table. This further normalizes the schema and prevents data redundancy. For example, the renter table has user_id as a foreign key and trips_taken depend on user_id, we will move trips_taken to a new table with user_id as the primary key. This step enhances data integrity and avoids data duplication.

By applying all the normalization steps, we have reviewed the resulting schema to ensure that it meets the requirements and efficiently represents the relationships between entities. We will optimize the schema based on our specific use case and performance requirements. This involved making adjustments to the structure, adding indexes for faster querying, and ensuring data integrity through appropriate constraints. The optimized schema will provide a solid foundation for building comprehensive applications related to the travel industry, allowing efficient management and analysis of data from flights, hotels, and car rentals.

After normalizing the data to 3NF state, the process of building our database schema involves dividing it into four main parts, namely: users, flights, hotels, and car rentals. Each part contains several related tables that store specific information to efficiently manage and organize the data. Let's explain the rationale behind each part and its associated tables:

Users related tables:

```
CREATE TABLE users (
    user_id SERIAL PRIMARY KEY,
    first_name VARCHAR(50),
    last_name VARCHAR(50),
    email VARCHAR(100),
    phone_number VARCHAR(20)
);

CREATE TABLE payment_info (
    id SERIAL,
    payment_type_id SERIAL NOT NULL PRIMARY KEY,
    account_details VARCHAR(100),
    FOREIGN KEY (id) REFERENCES users(user_id)
);
```

1. Users Table (users):

   This table stores information about users, including a unique user_id (primary key), first_name, last_name, email, and phone_number. It serves as the foundation for managing user details throughout the database.

2. Payment Info Table (payment_info):

The payment_info table contains details related to payment methods used by users. It includes an id (serial), payment_type_id (primary key, referencing the payment type), and account_details. The foreign key constraint connects payment in account information to a user in the users table.

Flights related tables:

```sql
CREATE TABLE airline (
    airline_id SERIAL PRIMARY KEY,
    carrier_name VARCHAR(50)
);

CREATE TABLE flight (
    flight_id SERIAL PRIMARY KEY,
    airline_id SERIAL NOT NULL,
    tail_num VARCHAR(50),
    dest VARCHAR(50),
    elapsed_time INT,
    dep_time INT,
    weather_condition VARCHAR(100),
    FOREIGN KEY (airline_id) REFERENCES airline(airline_id)
);

CREATE TABLE flight_user (
    id SERIAL PRIMARY KEY,
    user_id INT NOT NULL,
    flight_id INT NOT NULL,
    FOREIGN KEY (user_id) REFERENCES users(user_id),
    FOREIGN KEY (flight_id) REFERENCES flight(flight_id)
);
```

3. Airline Table (airline):

   This table stores data related to airlines. It includes an airline_id (primary key) and carrier_name representing the name of the airline.

4. Flight Table (flight):

   This table stores flight details, including flight_id, airline_id, tail_num, dest (destination), elapsed_time, dep_time (departure time), and weather_condition. The airline_id references the airline table.

5. Flight-User Relationship Table (flight_user):

   This table establishes a many-to-many relationship between users and flights. It includes an id (serial), user_id (foreign key referencing the users table), and flight_id (foreign key referencing the flight table). It helps track which users are associated with specific flights.

Hotels related tables:

```sql
CREATE TABLE hotel (
    hotel_id SERIAL PRIMARY KEY,
    hotel_type VARCHAR(50)
);

CREATE TABLE room_type (
    room_type_id SERIAL PRIMARY KEY,
    room_type_name VARCHAR(50)
);

CREATE TABLE hotel_booking (
    booking_id SERIAL PRIMARY KEY,
    hotel_id INT NOT NULL,
    room_type_id INT NOT NULL,
    is_canceled BOOLEAN,
    arrival_date_year INT,
    arrival_date_month VARCHAR(20),
    arrival_date_week_number INT,
    arrival_date_day_of_month INT,
    lead_time INT,
    stays_in_weekend_nights INT,
    stays_in_week_nights INT,
    FOREIGN KEY (hotel_id) REFERENCES hotel(hotel_id),
    FOREIGN KEY (room_type_id) REFERENCES room_type(room_type_id)
);


CREATE TABLE customer_type (
    customer_type_id SERIAL PRIMARY KEY,
    customer_type_name VARCHAR(50)
);

CREATE TABLE guest (
    guest_id SERIAL,
    booking_id INT NOT NULL,
    adults INT,
    children INT,
    babies INT,
    previous_cancellations INT,
    previous_bookings_not_canceled INT,
    is_repeated_guest BOOLEAN,
    deposit_type VARCHAR(50),
    customer_type_id INT,
    required_car_parking_spaces INT,
    FOREIGN KEY (guest_id) REFERENCES users(user_id),
    FOREIGN KEY (booking_id) REFERENCES hotel_booking(booking_id),
    FOREIGN KEY (customer_type_id) REFERENCES customer_type(customer_type_id)
);
```

6. Hotel Table (hotel):

   The hotel table stores data related to hotels. It includes a unique hotel_id (primary key), and hotel_type.

7. Room Type Table (room_type):

   This table contains various types of hotel rooms. It includes a unique room_type_id (primary key) and room_type_name.

8. Hotel Booking Table (hotel_booking):

   This table stores hotel booking data, including booking_id, hotel_id, room_type_id, is_canceled, arrival_date_year, arrival_date_month, arrival_date_week_number, arrival_date_day_of_month, lead_time, stays_in_weekend_nights, and stays_in_week_nights. The hotel_id and room_type_id are references to the hotel and room_type tables, respectively.

9. Customer Type Table (customer_type):

This table stores data about different types of hotel customers. It includes a unique customer_type_id (primary key) and customer_type_name.

10. Guest Table (guest):

    This table stores guest-related information, including guest_id, booking_id, demographic data (adults, children, babies), booking history (previous_cancellations, previous_bookings_not_canceled), is_repeated_guest, deposit_type, customer_type_id, and required_car_parking_spaces. The table has foreign keys to link with other relevant tables.

Car Rentals related tables:

```sql
CREATE TABLE renter (
    renter_id INT PRIMARY KEY,
    trips_taken INT,
    review_count INT,
    user_id INT NOT NULL,
    FOREIGN KEY (user_id) REFERENCES users(user_id)
);

CREATE TABLE vehicle (
    vehicle_id SERIAL PRIMARY KEY,
    fuel_type VARCHAR(50),
    rating FLOAT,
    make VARCHAR(100),
    model VARCHAR(100),
    type VARCHAR(50),
    year INT
);

CREATE TABLE rental (
    rental_id SERIAL PRIMARY KEY,
    vehicle_id INT NOT NULL,
    daily_rate NUMERIC(10, 2),
    FOREIGN KEY (vehicle_id) REFERENCES vehicle(vehicle_id)
);

CREATE TABLE car_rental_user (
    car_rental_user_id SERIAL PRIMARY KEY,
    first_name VARCHAR(50),
    last_name VARCHAR(50),
    email VARCHAR(100),
    phone_number VARCHAR(20)
);

CREATE TABLE car_rental_reservation (
    reservation_id SERIAL PRIMARY KEY,
    user_id INT NOT NULL,
    rental_id INT NOT NULL,
    renter_id INT NOT NULL,
    vehicle_id INT NOT NULL,
    FOREIGN KEY (user_id) REFERENCES car_rental_user(car_rental_user_id),
    FOREIGN KEY (rental_id) REFERENCES rental(rental_id),
    FOREIGN KEY (renter_id) REFERENCES renter(renter_id),
    FOREIGN KEY (vehicle_id) REFERENCES vehicle(vehicle_id)
);
```

11. Renter Table (renter):

    The renter table contains data about car renters. It includes a unique renter_id (primary key), trips_taken, review_count, and user_id (foreign key referencing the users table).

12. Vehicle Table (vehicle):

    This table holds information about vehicles available for rental. It includes a unique vehicle_id (primary key), fuel_type, rating, make, model, type, and year.

13. Rental Table (rental):

    The rental table captures rental-specific details. It includes a unique rental_id (primary key), vehicle_id (foreign key referencing the vehicle table), and daily_rate.

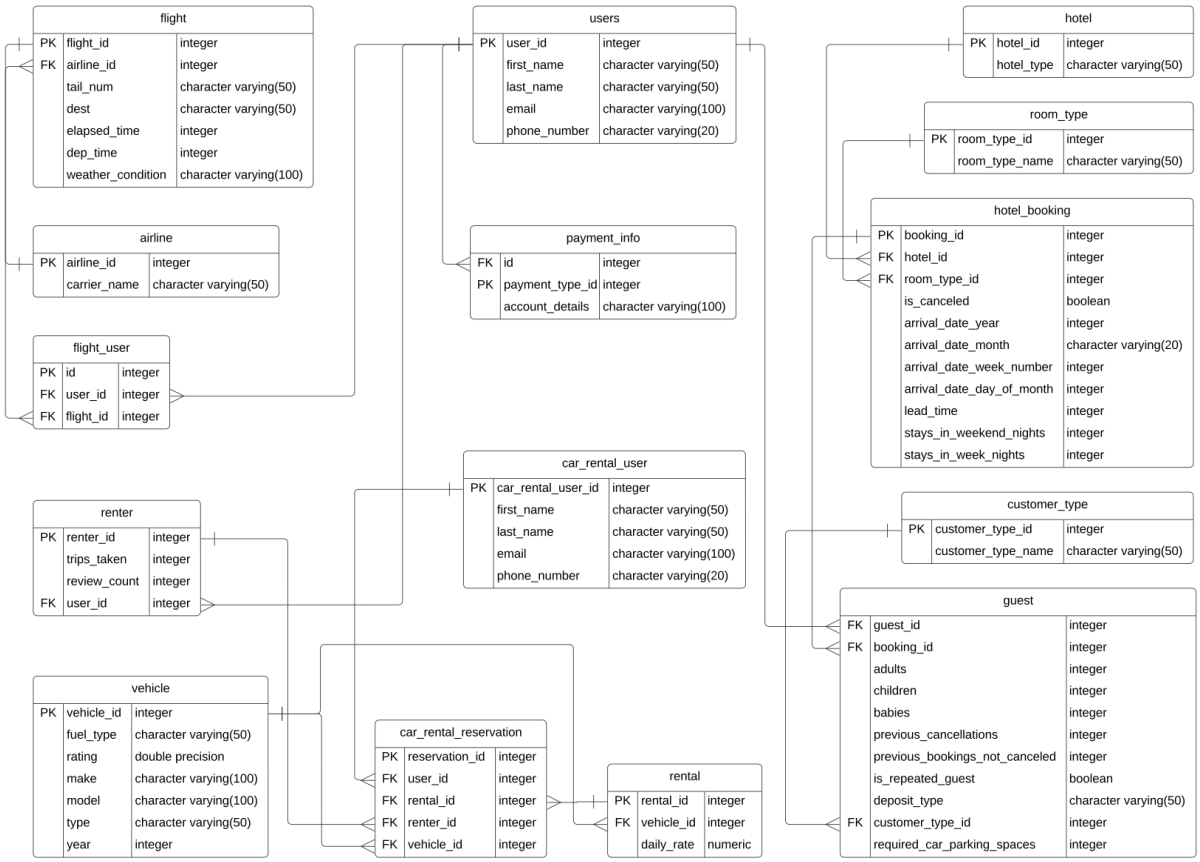
14. Car Rental User Table (car_rental_user):

    This table stores user information specifically related to car rentals. It includes a unique car_rental_user_id (primary key), first_name, last_name, email, and phone_number.

15. Car Rental Reservation Table (car_rental_reservation):

    The car_rental_reservation table manages car rental reservations. It includes a unique reservation_id (primary key), user_id (foreign key referencing the car_rental_user table), rental_id (foreign key referencing the rental table), renter_id (foreign key referencing the renter table), and vehicle_id (foreign key referencing the vehicle table).

By dividing the database schema into these four parts and creating multiple related tables, we can efficiently store, manage, and analyze data from the flights, hotels, and car rentals datasets. This organized approach allows us to build powerful applications and gain valuable insights from the integrated travel data.



ETL process

In order to better analyze our data, we made a plan for Transforming and Entering the Data into the Database System:

Our data transformation and entry process involves several key steps to ensure the CSV data is effectively integrated into our PostgreSQL database while adhering to the database schema requirements. And here is the link to our code and detailed process: https://github.com/ZhenyuWangg/5310_final_project_group7.git

Data Extraction:

Initially, we will retrieve data from the CSV files containing information about users, flights, car rentals, hotels, reservations, and reviews.

Data Cleaning:

Once the data is extracted, we will employ pandas, a powerful data manipulation library, to perform thorough data cleaning. This involves multiple tasks such as eliminating duplicate records, handling missing data either by filling it with generic values like "Unknown" or dropping the rows entirely to maintain data integrity. We will also verify and adjust data types, ensuring, for instance, that user IDs are represented as integers. This data cleaning process will be applied to each table individually.

Data Transformation:

Following data cleaning, the next crucial step is data transformation to align the data with the database schema requirements. We will utilize pandas to carry out various transformations, such as splitting columns when necessary to match the database schema structure. For example, if our CSV has a 'name' column, but the database requires separate 'first_name' and 'last_name' fields, we will split the 'name' column accordingly. Additionally, we will convert date and time formats to match the database's expectations, like changing 'dd-mm-yyyy hh:mm:ss' to 'yyyy-mm-dd hh:mm:ss'. For boolean fields, we will map 'Yes'/'No' in the CSV to true/false in the database, ensuring consistency, particularly in fields like 'availability_status' for car_inventory. Moreover, we will validate foreign keys to ensure that referenced data exists in the parent table. In specific cases, such as the hotel_rooms table, we may need to convert 'price_per_night' from one currency to another, such as converting Euros to US dollars.

Data Loading:

After data cleaning and transformation, the cleaned and formatted data will be loaded into our PostgreSQL database. We will follow an order that respects foreign key constraints to guarantee

the integrity of data relationships. For example, tables like users, airlines, airports, car_rental, car_make, hotels should be loaded before tables like flights, car_inventory, hotel_rooms, etc.

Handling Missing Data:

In some cases, certain tables like car inventory and car rental reservations might lack corresponding data in the CSV files. To address this, we will apply logical rules or assumptions to generate appropriate data to maintain consistency and completeness within the database.

By systematically executing the data extraction, cleaning, transformation, and loading steps, we will successfully integrate the travel service platform's data into our PostgreSQL database, ensuring accuracy, adherence to schema requirements, and facilitating seamless user interactions with the platform's various features.

Analytics Applications

After we had finished building the database and had finished the ETL process, we began to go through ten questions in an attempt to gain more insightful conclusions about flights, hotels, and car rental. .

1. Which airline carrier is mostly used?

```
SELECT
    a.carrier_name,
    COUNT(*) AS num_flights
FROM
    flight f
    JOIN airline a ON f.airline_id = a.airline_id
GROUP BY
    a.carrier_name
ORDER BY
    num_flights DESC
LIMIT 10;
```

Insight: This question delves into the preferences of travelers, shedding light on the airline carriers that are most frequently chosen by customers. Understanding the dominant carrier provides valuable insights into market share, customer loyalty, and the effectiveness of marketing efforts. By identifying the preferred carrier, airlines can tailor their services, partnerships, and customer engagement strategies to enhance customer satisfaction and retention.

2. What is the average elapsed time for different destinations?

```
SELECT
    f.dest,
    AVG(f.elapsed_time) AS avg_elapsed_time
FROM
    flight f
GROUP BY
    f.dest;
```

Insight: Examining the average time it takes to travel to various destinations offers a comprehensive overview of travel durations. Airlines can use this information to optimize flight schedules, manage crew rotations, and offer passengers more accurate travel expectations. Additionally, insights into variations in travel times can aid in identifying routes that may require operational adjustments or improvements in order to enhance overall travel efficiency and customer experience.At the same time, for users, flight times allow them to choose more appropriate flights based on their flight preferences.

3. What is the most common departure time for different destinations?

```
SELECT
    f.dest,
    MAX(f.dep_time) AS most_common_dep_time,
    MAX(departure_count) AS max_departure_count
FROM (
    SELECT
        dest,
        dep_time,
        COUNT(*) AS departure_count
    FROM
        flight
    GROUP BY
        dest, dep_time
) f
GROUP BY
    f.dest;
```

Insight: This question highlights the preferred departure times for various destinations, providing airlines with valuable data to optimize flight schedules. Understanding peak departure times helps minimize congestion at airports, improve on-time performance, and provide travelers with options that align with their travel preferences. By analyzing departure patterns, airlines can enhance operational planning and offer more tailored travel choices to customers.

4. Which room type (room_type_name) is most commonly booked for stays that include weekend nights (stays_in_weekend_nights)?

```
SELECT
    rt.room_type_name,
    COUNT(*) AS num_bookings
FROM
    hotel_booking hb
    INNER JOIN room_type rt ON hb.room_type_id = rt.room_type_id
WHERE
    hb.stays_in_weekend_nights > 0
GROUP BY
    rt.room_type_name
ORDER BY
    num_bookings DESC
LIMIT 10;
```

Insight: This inquiry unveils guests' room preferences for weekend stays, offering hotels a deeper understanding of their customer base. By identifying the most sought-after room types

for weekend trips, hotels can allocate resources effectively, adjust pricing strategies, and optimize room availability. Tailoring room offerings to match guests' preferences can result in improved guest satisfaction, increased bookings, and enhanced revenue opportunities.For users, they can more clearly observe the popularity of different rooms and use it as a basis for room selection. At the same time, more popular rooms may need to be booked longer in advance.

5. Which month (arrival_date_month) has the highest number of bookings among hotel bookings?

```sql
SELECT
    arrival_date_month,
    COUNT(*) AS booking_count
FROM
    hotel_booking
GROUP BY
    arrival_date_month
ORDER BY
    booking_count DESC
LIMIT 10;
```

Insight: Exploring the month with the highest booking volume provides hotels with a crucial insight into peak demand periods. Armed with this knowledge, hotels can prepare for increased guest arrivals by adjusting staffing levels, planning promotional campaigns, and optimizing room rates. Understanding booking patterns allows hotels to strategically manage resources and deliver exceptional guest experiences during busy periods. For users, this data information can first and foremost help them make travel time choices. They can choose to avoid peak hotel booking periods on this basis. Secondly, it can help them to judge the booking time better. For peak months, they may need to book in advance.

6. Are bookings made by guests requiring car parking spaces (required_car_parking_spaces) more likely to be canceled (is_canceled) compared to bookings that don't require parking?

```sql
SELECT
    required_car_parking_spaces,
    COUNT(*) AS total_bookings,
    SUM(CASE WHEN hb.is_canceled THEN 1 ELSE 0 END) AS canceled_bookings,
    (SUM(CASE WHEN hb.is_canceled THEN 1 ELSE 0 END) * 100.0 / COUNT(*)) AS cancellation_percentage
FROM
    guest g
    JOIN hotel_booking hb ON g.booking_id = hb.booking_id
GROUP BY
    required_car_parking_spaces;
```

Insight: By analyzing the correlation between parking needs and booking cancellations, hotels can gain insights into customer behavior and preferences. This analysis can highlight potential pain points for guests requiring parking and guide improvements in parking facilities, policies, and communication strategies. It also enables hotels to segment their guest

base for targeted marketing efforts and tailor cancellation policies to address specific guest needs.

7. What is the average lead time for hotel bookings, and does this vary based on the type of hotel (hotel_type)?

```
SELECT
    h.hotel_type,
    AVG(hb.lead_time) AS avg_lead_time
FROM
    hotel h
    JOIN hotel_booking hb ON h.hotel_id = hb.hotel_id
GROUP BY
    h.hotel_type;
```

Insight: Investigating lead times in relation to hotel types unveils booking trends and customer behaviors. Hotels can identify whether certain types of accommodations attract more early planners or last-minute bookings, allowing for tailored marketing strategies and pricing adjustments. Understanding lead times helps hotels optimize resource allocation, manage inventory, and offer promotions to attract bookings during specific timeframes. For users, they can learn about the popularity of different hotel types and make reservations based on that.

8. What is the average daily rate for each vehicle type (type), and how does it vary based on the fuel type (fuel_type)?

```
SELECT
    type,
    fuel_type,
    AVG(daily_rate) AS avg_daily_rate
FROM
    rental r
    JOIN vehicle v ON r.vehicle_id = v.vehicle_id
GROUP BY
    type, fuel_type;
```

Insight: Analyzing the average daily rates for different vehicle types based on their fuel type provides car rental businesses with insights into customer preferences and market trends. It enables businesses to adjust pricing strategies for different vehicle categories and fuel options. This information also helps inform fleet management decisions, including vehicle acquisitions and fuel-efficient offerings, resulting in more competitive and appealing pricing for customers. For users, they can get more detailed pricing information related to car rentals. For example a user wants to rent a gasoline car, or a user wants to rent an electric buggy. Such a price display allows for a clearer categorization of car type and fuel type.

9. What is the most commonly rented car make and model, and what is its average daily rate?

```
SELECT
    v.make,
    v.model,
    COUNT(*) AS num_rentals,
    AVG(r.daily_rate) AS avg_daily_rate
FROM
    rental r
    JOIN vehicle v ON r.vehicle_id = v.vehicle_id
GROUP BY
    v.make, v.model
ORDER BY
    num_rentals DESC
LIMIT 10;
```

   Insight: Identifying the most popular car make and model, along with its average daily rate, empowers car rental companies with actionable insights. By understanding customer preferences, companies can ensure an adequate supply of high-demand vehicles, target marketing efforts effectively, and fine-tune pricing strategies. Offering the most sought-after vehicles at competitive rates can drive customer loyalty, increase bookings, and optimize revenue streams. For users, this information can tell them what kind of vehicle is most popular in their local area. They can use this as a basis for making choices and deciding on their appointment times.

10. What is the average number of trips taken by renters, and how does it correlate with their review counts?

```
SELECT
    AVG(trips_taken) AS avg_trips_taken,
    AVG(review_count) AS avg_review_count
FROM
    renter;
```

   Insight: Analyzing the relationship between the number of trips taken by renters and their review counts reveals valuable insights into customer engagement and satisfaction. This correlation highlights whether more frequent renters are more likely to provide feedback, contributing to the establishment of a strong online reputation. Businesses can leverage this insight to encourage customer reviews, enhance customer interactions, and ultimately improve the overall renter experience.

By asking these insightful questions and analyzing the corresponding data, businesses in the travel industry that are related to flight, hotel, and car rental can make informed decisions, optimize operations, and provide exceptional experiences for their customers. At the same time, users can also query more trustworthy and insightful data information according to their needs, and use it as a basis to make better travel choices.

Interaction Plan

Analyst Interaction:

1. Direct Querying (SQL): Analysts will have access to the PostgreSQL database and can use SQL to write custom queries for in-depth analysis and data manipulation. They can use SQL clients like the pgAdmin tool.
2. Python Integration: Analysts can use Python's psycopg2 library to programmatically connect to the database. They can write Python scripts to run SQL queries and perform more complex data manipulations and analyses.
3. Jupyter Notebooks: Analysts can leverage Jupyter notebooks to create interactive documents. They can write SQL code in notebook cells, mix it with Python code for analysis, and visualize results using libraries like pandas and matplotlib.

"C" Level Officer Interaction:

1. Metabase Reports: Predefined reports will be created in Metabase for "C" level officers. These reports will be carefully designed to provide high-level insights and trends.
2. Dashboard Access: "C" level officers will have access to Metabase's web-based interface. They can log in to view and interact with the reports and dashboards we've designed.
3. Scheduled Reports: We can set up scheduled report delivery in Metabase, ensuring that key insights are delivered to executives' inboxes at specified intervals.

Tools and Programming Languages:

1. SQL: SQL will be the primary language for querying the PostgreSQL database. Analysts will write custom SQL queries for data manipulation and analysis.
2. Python: Analysts will use Python, along with libraries like psycopg2, for programmatic database interactions and complex data processing.
3. Metabase: Metabase will serve as the user-friendly dashboard and reporting platform for both analysts and executives.

Benefits of Using Programming Languages:

SQL and Python offer the flexibility to perform custom analyses tailored to specific business questions and scenarios. Python allows analysts to automate repetitive tasks and conduct complex calculations efficiently. Using Python, analysts can process and transform data before analysis, enhancing the quality of insights.

Non-Technical Personnel Interaction:

Metabase is specifically designed to cater to non-technical users, offering an intuitive interface for interacting with databases without requiring SQL or programming knowledge. They can access dashboards, filter data, and gain insights without writing any code.

Redundancy and Performance:

We ensure that the PostgreSQL database is set up with appropriate redundancy mechanisms like database replication and load balancing will ensure data availability and distribute workloads efficiently. Performance optimization strategies including indexing, query tuning, and caching are also part of the plan to enhance system efficiency. As for hosting, the decision between on-premises and cloud hosting depends on factors such as scalability needs, data security preferences, budget constraints, and technical expertise. Both options have their benefits, with on-premises offering more control and cloud providing scalability and managed services. The choice should align with the client's specific requirements and strategic goals. However, we should choose to host their database system on-premises. This decision aligns with the benefits of greater control over infrastructure, data security, compliance, and cost predictability, while also allowing for tailored networking and resource allocation. Besides, it's essential to carefully plan for ongoing maintenance, monitoring, and disaster recovery to ensure the continued reliability and performance of the on-premises database system.

Metabase Dashboard:

1. Airline Carrier Name Distribution (Bar Chart):

   This chart offers insights into the most frequently chosen airlines by users. The bar chart's presentation allows users to quickly identify the most popular carriers, helping them make informed decisions when selecting an airline for their travel.

2. Flight Weather Conditions (Pie Chart):

   The pie chart showcases historical weather conditions during flight bookings. Users can gain an understanding of weather patterns, helping them anticipate potential weather-related challenges and make more informed travel plans.

3. Car Rent Fuel Types (Bar Chart):

   This bar chart illustrates the popularity of different fuel types chosen for car rentals. Users can easily identify the preferred energy sources, aiding them in selecting a car rental that aligns with their preferences and values.

4. Hotel Booking Arriving Month (Bar Chart):

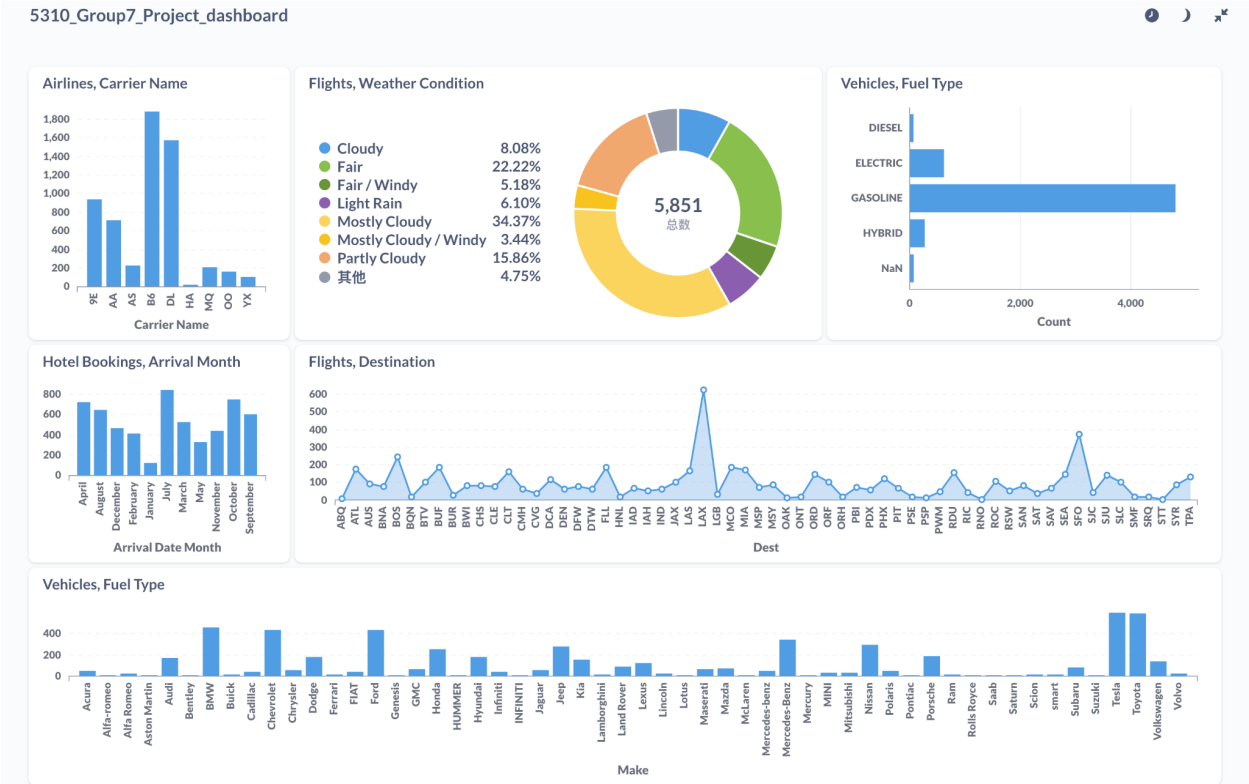
By presenting the most popular months for hotel arrivals, this bar chart assists users in planning their trips and adjusting travel dates. It provides insights into peak travel times, enabling users to optimize their travel plans.

5. Flight Destination Analysis (Line Chart):

   The line chart reveals the frequency of city destinations users arrive at most frequently. This information can help users explore trends in their travel history and make more informed decisions when booking flights, possibly influencing their choice of destinations.

6. Vehicle Make Brands (Bar Chart):

   The bar chart displaying trending vehicle make brands provides users with insights into the most favored car rental options. This information is valuable for users seeking the latest and most preferred vehicle brands for their rentals.



Benefits of Metabase Dashboard:

The visual nature of the dashboard makes it easy for users to quickly grasp important information about their travel choices. Users can make better decisions based on historical data and patterns provided by the charts. The dashboard can help users receive personalized

recommendations, such as flight destinations or car rental brands, based on their preferences and trends.

Conclusion

In conclusion, our project report offers a comprehensive analysis of the challenges confronting SuperTravel and the promising opportunities it stands to gain. Through meticulous research and the exploration of extensive datasets, the application of normalization techniques, the implementation of an ETL process, and the utilization of analytical tools like Metabase, we have delved deep into the intricacies of SuperTravel's business landscape to develop this innovative platform. As technology continues to evolve and with a solid foundation in database construction, coupled with the facilitation of user-company interactions, we hold the belief that SuperTravel is poised for success. By delivering convenience, personalization, and cost-effective solutions to consumers in the travel industry, SuperTravel is poised to redefine the landscape of travel planning.