Relational Database Project Report: Group 3

Customer Scenario Statement (Xueqi Zhou)

When clients are planning a trip, tickets on third-party sites are always a little more expensive. Some clients may choose to book directly through the provider's direct sale platforms. However, if customers want to book airline tickets, rental cars, and hotels all at once, visiting their respective websites can be time-consuming and inefficient. Customers prefer SuperTravel because it is a third-party one-stop shop where they can find all of the best offers. In summary, we found this SuperTravel concept to be really exciting and realistic, and we would love to construct the database they requested.

As SuperTravel advisors, we found their concept of a one-stop platform for purchasing airline tickets, rental cars, and hotels appealing and relevant for customers looking for ease. Our motivation is to build a strong database that meets their requirements and improves their decision-making process.

To get ready, we looked into trip planning sites like Booking.com, Airbnb, and TripAdvisor to learn about the standard data needed for a travel booking database. SuperTravel's data will be centralized under our proposed design, allowing for rapid and efficient access for better business decisions. To imitate real-life settings, we'll collect and manipulate sample data from sites like Kaggle.

Our work will provide three main benefits to SuperTravel:

1. Centralized platform: An improved database design will enable efficient access to customer data, booking information, user reviews, and feedback, enhancing customer service.
2. Better understanding of customer preferences: Analyzing customer data will help SuperTravel make informed strategic decisions and tailor their offerings to user preferences.
3. Enhanced data consistency and accuracy: A new, reformed database with 3NFs will address previous issues of untimely updates and inaccurate data records, increasing customer satisfaction and reducing complaints.

By providing these benefits, our work will give SuperTravel a competitive advantage, attracting more customers and driving business success for the company.

Responsibility Statement

Yanlin Zhou: proposal (25%), dataset source (75%), initial database design(50%), ER diagram revision(50%), ETL (47%), 3 of the queries, report(30%)
Xueqi Zhou: proposal (25%), dataset (25%), initial database design-normalization(50%), ER diagram design(40%) and revision(50%), ETL (53%), 7 of the queries, report(70%)
Ziting Guo: proposal (25%), database creation code (100%), data generation (100%), Metabase interface design (100%)

Candice Huang: proposal (25%), ER diagram (60%), ETL (data insertion), PowerPoint
The following report is written by the members mentioned in parentheses in each part.

Data Source （ Yanlin Zhou)

The major datasets we used are directly retrieved from Kaggle. We combined three datasets to fulfill the basic need of this project. The Kaggle links are the following:
https://www.kaggle.com/datasets/leomauro/argodatathon2019 for flights and hotels data
https://www.kaggle.com/datasets/kushleshkumar/cornell-car-rental-dataset for car rental data
https://www.kaggle.com/datasets/michelhatab/hotel-reviews-bookingcom for hotel_review text and ratings

We are choosing these datasets because they fulfill our need to simulate a database for our clients that would satisfy the needs of incorporating the functionalities of hotels, flights, and car rental bookings.

The first two datasets are the main datasets we used for this project. The missing data that are considered necessary for the data schema but are not available by internet search are generated using random and faker Python packages. The code to generate the missing datasets is uploaded to the space for reference.

Data Storage

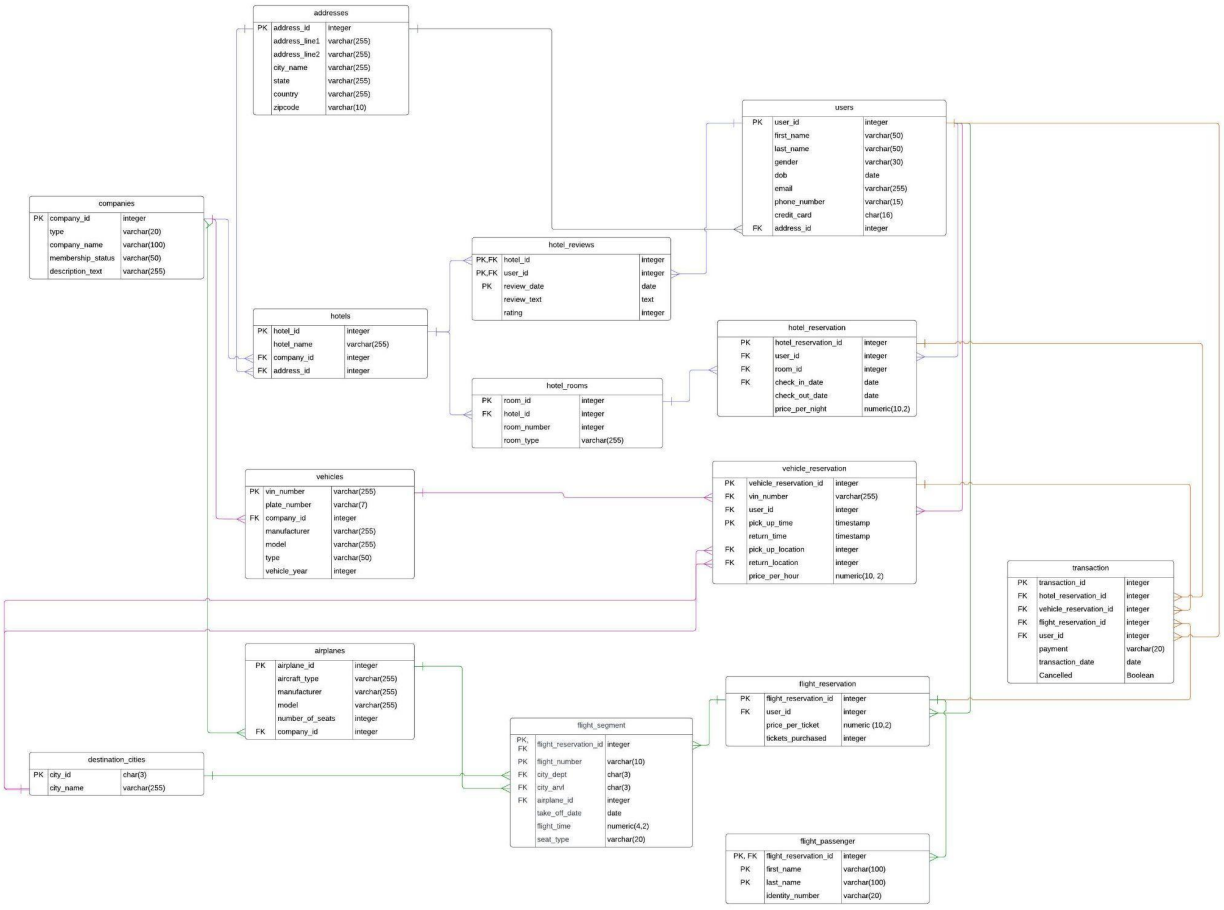
Part one: Database Design (Yanlin Zhou)

To design a database that would properly work for our client's (SuperTravel) need, we designed a database with 15 3NF tables. The ER diagram that represents the database design and the code for generating the database is included in the project folder. The following is a brief introduction to how the database was designed.

In general, this database schema is designed to support all the functionality of a travel booking system. This system would allow users to book hotels, cars, and flights, make reviews for their hotel bookings, as well as make transactions and manage their reservations. As SuperTravel serves as a platform that connects companies and individual users, we included the "company" table to record the company's information and their membership status with SuperTravel (for managing promotions, for example).

The following is the ER diagram for this database project (both the diagram itself and the LucidChart link to the diagram are included in the work folder). The table records the relationship between schemas and the data type of each

attribute.



Here I will be explaining some of the attributes that may be confusing in meanings or data forms. As mentioned, in the company table, the “membership_status” attributes would record the collaboration membership of the specific company with SuperTravel as SuperTravel is an intermediate platform. Each vehicle/hotel/airplane would correspond to a car rental company/hotel managing group/airline represented by the foreign key: company ids.

In the addresses table, the zipcode is stored as varchar (10) to incorporate the different lengths of zipcodes in different countries (though in our sample data, we are only using American zip codes). In the vehicles table, vin_number is a unique identifier of each vehicle, so it’s set as the primary key to identify each vehicle.

In the hotel_reservation tables, users are checking into hotel rooms instead of hotels, since the price of each room may vary based on room instead of based on hotel.
In the destination_cities table, cities are recorded for vehicle_reservation and flight_segment schemas. This corresponds to all the unique cities mentioned in our data source and is recorded as the cities for flight departure/arrivals, and the locations for car pick-up/returns.

Lastly, the transaction table references the user table and three reservation tables. This means users are able to make one, two, or all three kinds of reservations and make one

transaction for all three. This table also has a boolean attribute: canceled, to record cancellation which is very common for travel booking websites.

Part two: ETL and Data Insertion (Yanlin Zhou&Xueqi Zhou)

Caveat: **due to the fact that a portion of the data used is randomly generated, please use the backup database to retrieve the same query results as presented.**
Please look at the Python file (ETL) for more information.

Here's a detailed explanation of the ETL code on the specific parts of the database:
First, we imported necessary libraries, such as pandas, random, and sqlalchemy. We then read in data from various CSV files, including user data, flight data, hotel data, car rental data, etc. We use a random sample to sample several hundreds of rows of data from the large datasets for this project.

1. User, company, address, and destination city data (Yanlin Zhou)
    - Generate user data that fulfills the need of the database schema for simulation purposes, append real ages into the users' dataframe but convert them into DOBs to ensure age records are up to date. Generate random addresses for each of the users.
    - For destination cities, they are selected as the unique values in the flight destination dataset.
    - Three of the companies are from the original data that represent specific airlines. We randomly generate three hotel groups and three car rental companies and assign them to each hotel/vehicle.
2. Hotel data (Yanlin Zhou)
    - Hotels' names are subtracted from the unique hotels in our original dataset
    - Assigned a random 'company_id' to each hotel, using the 'company_id' values from 'companies_df' filtered by type "hotel".
    - Randomly assign a 'city_id' to each hotel from the 'cities_df' DataFrame.
    - Generate hotel rooms for each hotel, and randomly assign them to each of the reservation records using dictionary mapping, use room_ids instead of room_numbers in the reservation table
    - The general idea is to select rows from the original data that fit into the specific row names in the SQL database
    - Append ids to the hotel, hotel room, and hotel reservation tables.
    - Load data into SQL using the sqlalchemy package
    - Select hotel reviews (using the titles only for demonstration purposes) and ratings from the hotel_ratings dataset. Append these to the hotel_reservation dataset and select the required rows for the hotel_reviews schema using the check_out_date as the date of review to avoid contradiction. Load data into SQL using the sqlalchemy package
    - 3. Vehicle data (Xueqi Zhou)

- Sample vehicle data and store it in 'vehicle_sample'.
- Read vehicles and vehicle reservations data from CSV files and store them in DataFrames.
- Assign a random 'company_id' to each vehicle, using the 'company_id' values from 'companies_df' filtered by type "car rental".
- Randomly assign matched car data (manufacturer, model, type, and year) to each vehicle in 'new_df'.
- Randomly assign pick-up and return locations to each vehicle reservation in 'new_df'.
- Load the transformed DataFrames 'vehicles_df' and 'vehicle_reservation' into their respective SQL tables: 'vehicles' and 'vehicle_reservation'.

4. Flight data (Xueqi Zhou)
    - Sample flight data from 'flights_raw' DataFrame and store it in 'flight_sample'.
    - Read flight reservations data from 'flight_reservations.csv', flight segments data from 'flight_segment.csv', flight passenger data from 'flight_passenger.csv', and airplanes data from 'airplanes.csv', storing them in DataFrames 'flight_reservation', 'flight_segment', 'flight_passenger', and 'airplanes' respectively.
    - Create a new DataFrame 'new_flight_df' with the desired columns, such as flight_reservation_id, user_id, price_per_ticket, tickets_purchased, flight_number, city_dept, city_arvl, airplane_id, take_off_date, flight_time, and seat_type, using data from the sampled and read DataFrames.
    - Drop dependent tables with CASCADE to remove any constraints, then create and load the transformed DataFrames 'flight_reservation_df', 'flight_segment_df', and 'flight_passenger' into their respective SQL tables: 'flight_reservation', 'flight_segment', and 'flight_passenger'.
    - Assign random 'company_id' values to each airplane in 'airplanes' DataFrame using the 'company_id' values from 'companies_df' filtered by type "airline". Load the transformed 'airplanes' DataFrame into the SQL table 'airplanes'.
5. Transaction data (Ziting Guo)
    - Merge hotel and vehicle reservations DataFrames based on 'user_id', then merge the result with the flight reservations DataFrame based on 'user_id' as well, creating the 'merged_reservations' DataFrame.
    - Create a 'transaction' DataFrame with the required columns, then fill it using data from 'merged_reservations'.
    - Assign random payment methods and cancellation statuses for transactions, and generate random transaction dates.
    - Load the 'transaction' DataFrame into the SQL table 'transaction'.

Queries(Xueqi Zhou):

We propose here 10 query application questions to justify the benefits of this database as the following (the query codes are uploaded in another SQL file for your reference):

1. What are the top 5 popular locations with the highest number of flight bookings made through SuperTravel?
2. What are the hotel bookings in the past n days and how much does each cost?
3. What are the hotels' ratings from high to low and what do people think of them?
4. Which vehicle models are most frequently rented through SuperTravel?
5. How many users made reservations over SuperTravel in the last 90 days under each reservation type(car rental, hotel, flight)?
6. What is the average spending per booking for each type of reservation made through SuperTravel?
7. Who are the users who are most loyal to SuperTravel (made the most number of bookings)?
8. What is the canceled transactions information?
9. Which customer segment(by age) is likely to make the most bookings in the future based on the current bookings?
10. Which users have made three reservations at the same time? The one with the highest total amount is sorted from low to high.

The results from this query can help SuperTravel in several ways:

1. Top 5 popular flight booking locations: Identifies trending destinations and can help SuperTravel tailor marketing campaigns, promotions, and travel packages to cater to the growing demand.
2. Hotel bookings in the past n days and their costs: Allows SuperTravel to analyze booking trends and pricing strategies to optimize revenue management and identify patterns that can inform future promotional offers.
3. Hotels' ratings and reviews: Helps SuperTravel recommend the best hotels to their customers, improving customer satisfaction and increasing the likelihood of repeat business.
4. Most frequently rented vehicle models: Provides insights into customers' preferences for car rentals, enabling SuperTravel to optimize vehicle offerings and collaborate with rental partners for better deals.
5. Reservation counts in the last 90 days by reservation type: Offers an overview of the booking performance across different reservation types, helping SuperTravel understand which segments are driving business growth and require more investment.
6. Average spending per booking for each reservation type: Helps SuperTravel analyze customer spending patterns, informing pricing strategies and allowing for targeted upselling or cross-selling opportunities.
7. Most loyal users: Identifies high-value customers for personalized marketing, special offers, and dedicated customer support to foster loyalty and drive repeat business.

8. Canceled transaction information: Gives insights into the reasons behind cancellations, allowing SuperTravel to improve its services, reduce cancellations, and enhance customer satisfaction.
9. Customer segment most likely to make future bookings based on age: Helps SuperTravel identify customer segments with the highest potential for future bookings, allowing for targeted marketing campaigns and tailored product offerings to cater to their needs.
10. Users with three reservations at the same time, sorted by the total amount spent: Highlights high-value customers making multiple bookings simultaneously, offering to upsell and cross-sell opportunities while allowing SuperTravel to analyze revenue and customer satisfaction within this specific segment.

In summary, these 10 queries provide valuable insights into SuperTravel's customer behavior, preferences, and booking trends. By interpreting these insights, SuperTravel can make informed decisions for the companies it works with and itself, ultimately driving long-term success.

How can customers interact with our database system? (Xueqi Zhou)

In our proposed database system, we designed an efficient and user-friendly environment catering to different user groups, including analysts, executives, and non-technical personnel.
1. Analysts (Direct Querying):

Analysts can access the database through SQL clients and use programming languages like Python or R for more complex data manipulation and analysis tasks. They can collaborate on code using platforms like GitHub, and integrate with data visualization libraries for better insights.
2. "C" Level Officers (Reports):

Executives receive regular reports focused on key performance indicators and vital metrics that are prepared to utilize platforms such as Metabase Dashboard, Tableau, or Google Data Studio. We've decided to use Metabase for our dashboard display to C-level executives. They can get these reports via web-based dashboards or via email, with customization possibilities to dig further into certain areas of interest.

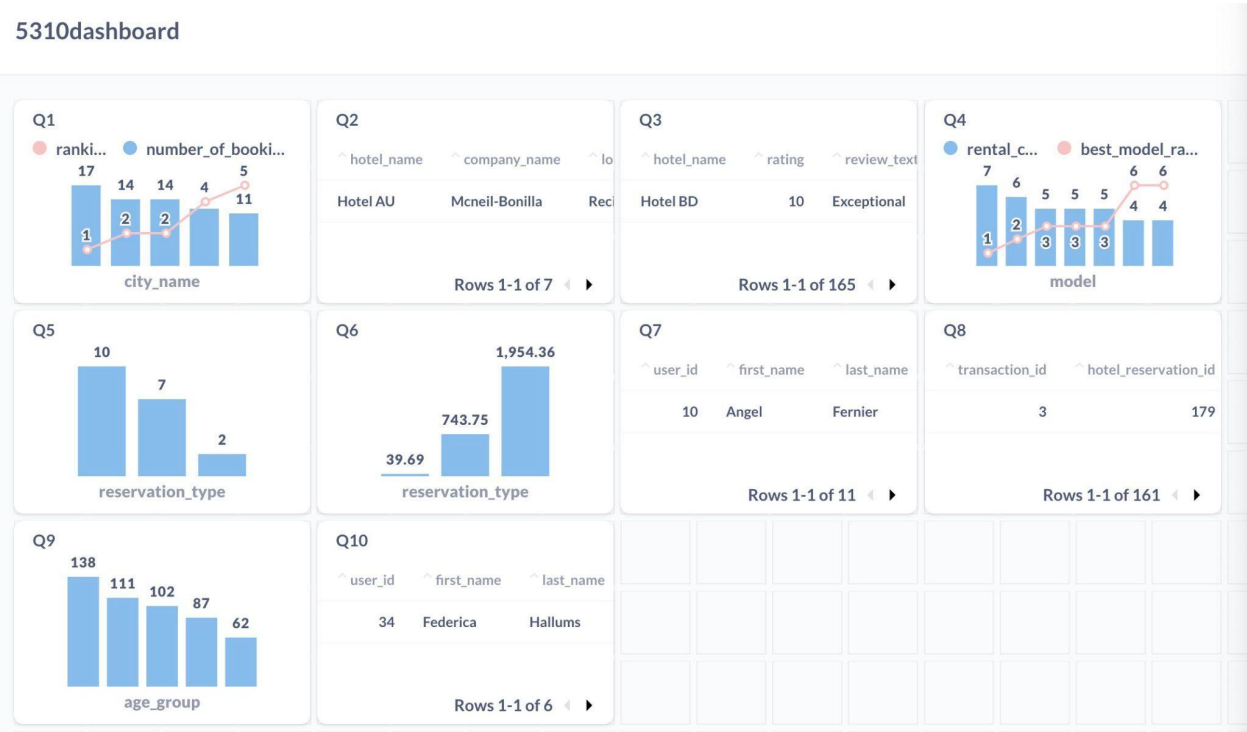
Non-technical personnel may interact with the database via a custom-built web or mobile app, which provides a user-friendly interface for data entry, retrieval, and basic reporting chores in future product development. To guarantee that data and feature access are limited to specific job tasks, user access will be regulated by role-based access control.

To summarize, our database system is designed to meet the needs of multiple user groups by offering immediate access to analysts, visually appealing reports to executives, and user-friendly interfaces to non-technical people. This strategy ensures that all users have efficient and effective data management and decision-making.

How are we dealing with redundancy and performance concerns? (Yanlin Zhou)

When designing a database system, it is critical to consider redundancy and performance. To address redundancy and performance, we would first strive to make our queries as efficient as feasible. Efficient queries would boost search speed while not slowing down query speed. Second, if the database is used in practice, the data would be stored on numerous distributed servers. This ensures performance even if one of the servers fails. Multiple servers would also allow backups of the database to be made. If a fatal error occurs during the database update process, we will be able to recover the database and maintain its essential functionality. Third, the relationships are normalized in their most basic form, with primary and foreign keys connecting them. We identified some columns that should be updated and streamlined depending on actual data throughout the data ETL process. Until now, our database designs in their 3NF variants had maintained data consistency and avoided potential data duplication issues.

Interactive Dashboard Display(Xueqi Zhou)



Some of the benefits of these dashboards include:

- Dashboards show complex data in a visually appealing and easy-to-understand format, which improves data visualization. Users may easily understand the trends and patterns in the data by using graphs, charts, and tables, allowing for better decision-making.

- Dashboards provide users with real-time insights into their KPIs, allowing them to monitor the current health of the business and make timely decisions. This real-time data access enables firms to react swiftly to new trends and opportunities.
- Improved cooperation and communication: Dashboards provide a single platform for teams, departments, and top management to share information and discuss discoveries. Dashboards promote greater communication and collaboration among team members by presenting data in a clear and simple manner.
- Dashboards offer data in a user-friendly format, allowing non-technical people to interact with the database and acquire insights without the requirement for specialist expertise or training in data analysis.
- Making informed decisions: Dashboards assist leaders and managers in making educated strategic decisions by offering a comprehensive perspective of the organization's performance across numerous measures. This data-driven decision-making strategy can result in better resource allocation, efficient procedures, and increased customer satisfaction.

Conclusion(Xueqi Zhou)

The project's purpose is to create and implement an efficient RDBMS for storing and managing travel-related data. We used ETL methods to clean, transform, and load data into the RDBMS from diverse sources. Finally, we ensure that our RDBMS can be used by clients to facilitate data analysis and derive insights that will assist them in making informed decisions and improving their business.

Throughout the project, we were able to accomplish the following:

1. Used SQL to create a well-structured RDBMS that enabled efficient data storage, retrieval, and management. By utilizing primary and foreign keys, as well as appropriate data types and constraints, the database design assured data integrity and consistency.
2. Developed Python ETL processes to read data from many sources (CSV files), clean and transform the data, and put it into the relevant RDBMS tables. This sped up data integration and gave the client access to up-to-date, consistent, and correct data from a single source.
3. The RDBMS and ETL processes enabled the client to undertake advanced data analysis and get important insights into the preferences, trends, and patterns of their customers. This data aided educated decision-making, resulting in better business strategies, higher customer happiness, and increased income.

In general, our RDBMS project provides SuperTravel with a strong and scalable data storage and management solution that ensures data integrity and security. It also allows for more complex data analysis, allowing our clients to acquire a complete and accurate grasp of their data. In the future, this database may boost customer satisfaction, including both businesses and

passengers, because our clients will be able to give more focused services and solutions. In other words, data will enable our customers to make more data-driven decisions and implement value-creating data management processes.