Data Acquisition and Manipulation Using SQL Certification Training

Course-End Project Problem Statement

Course-End Project - 2

Air Cargo Analysis

Overview

In this exercise, you will analyze and optimize data for Air Cargo, an aviation company, to improve the customer experience and operational efficiency. You will focus on identifying regular customers, analyzing the busiest routes, and preparing ticket sales details.

Instructions

- Review the learning materials in the SQL course
- Carefully read the situation, tasks, actions, and result sections to grasp the assignment fully
- Complete and submit your assignment via the Learning Management System (LMS)
- Follow the provided guidelines closely, ensuring your report includes all required analyses and interpretations

Situation

Air Cargo provides air transportation services for passengers and freight. The company aims to enhance customer satisfaction and operational efficiency. To achieve this, they must identify regular customers for personalized offers, analyze the busiest routes to optimize aircraft allocation, and prepare detailed ticket sales reports.

Task

Your tasks include identifying regular customers for targeted offers, analyzing the busiest routes to optimize aircraft allocation, and preparing detailed ticket sales

reports. This will help Air Cargo improve the customer experience and operational efficiency, making it a preferred choice for air travel.

Note: You must download the dataset from the course resource section in the LMS and create the tables to perform the above objective.

Input Dataset

Customer: Contains the information of customers

- customer_id – ID of the customer
- first_name – First name of the customer
- last_name – Last name of the customer
- date_of_birth – Date of birth of the customer
- gender – Gender of the customer

passengers_on_flights: Contains information about the travel details

- aircraft_id – ID of each aircraft in a brand
- route_id – Route ID of the from and to location
- customer_id – ID of the customer
- depart – Departure place from the airport
- arrival – Arrival place in the airport
- seat_num – Unique seat number for each passenger
- class_id – ID of the travel class
- travel_date – Travel date of each passenger
- flight_num – Specific flight number for each route

ticket_details: Contains information about the ticket details

- p_date – Ticket purchase date
- customer_id – ID of the customer
- aircraft_id – ID of each aircraft in a brand
- class_id – ID of travel class
- no_of_tickets – Number of tickets purchased
- a_code – Code of each airport
- price_per_ticket – Price of a ticket

- brand – Aviation service provider for each aircraft

routes: Contains information about the route details

- Route_id – Route ID of from and to location
- Flight_num – Specific fight number for each route
- Origin_airport – Departure location
- Destination_airport – Arrival location
- Aircraft_id – ID of each aircraft in a brand
- Distance_miles – Distance between departure and arrival location

Action

1. Create a database named **AirCargo** and import ticket_details.csv, routes.csv, passengers_on_flights.csv, and customer.csv from the given resources into it.
2. Create an **ER diagram** for the given airlines' database.
3. Write a query to display all the passengers who have traveled on routes **01** to **25** from the **passengers_on_flights table.**
4. Write a query to identify the **number of passengers** and **total revenue** in **business class** from the **ticket_details** table.
5. Write a query to display the **full name** of the **customer** by extracting the **first name** and **last name** from the **customer table.**
6. Write a query to extract the customers who have registered and booked a ticket from the customer and **ticket_details** tables.
7. Write a query to identify the **customer's first name** and **last name** based on their **customer ID** and brand (Emirates) from the **ticket_details** table.
8. Write a query to identify the customers who have traveled by Economy Plus class using the **sub-query** on the **passengers_on_flights** table.
9. Write a query to determine whether the revenue has crossed **10000** using the **IF** clause on the **ticket_details** table.
10. Write a query to **create** and **grant** access to a new user to perform database operations.
11. Write a query to find the **maximum ticket price** for each class using window functions on the **ticket_details** table.

12. Write a query to extract the passengers whose **route ID** is **4** by improving the speed and performance of the **passengers_on_flights table** using the index.
13. For **route ID 4**, write a query to view the execution plan of the **passengers_on_flights** table.
14. Write a query to calculate the **total price** of all tickets booked by a customer across different aircraft IDs using the **rollup function.**
15. Write a query to create a view with only business class customers and the airline brand.
16. Write a query to create a **stored procedure** that extracts all the details from the routes table where the traveled distance is more than **2000** miles.
17. Using **GROUP BY**, determine the total number of tickets purchased by each customer and the total price paid**.**
18. Calculate the average distance and average number of passengers per aircraft, considering only those routes with more than one departure date.

Result

Save your SQL queries with screenshots of the output in a Word document. Then, upload the solution document in the Learning Management System (LMS).