**Air Cargo Database Analysis – Theory**

**Introduction**

The Air Cargo Database project focuses on creating and analyzing a relational database system for an airline company. The aim is to organize data related to customers, flights, ticket bookings, routes, and aircraft operations in a structured format. The assignment demonstrates how SQL concepts—such as table creation, relationships, joins, aggregation, indexing, and stored procedures—can be applied to manage and analyze business data efficiently.

The main goal is to understand how relational databases can support airline operations, from recording customer information to analyzing revenue, flight distance categories, and customer travel patterns.

**Database Design**

The database was designed following the Entity-Relationship (ER) model, which identifies entities, attributes, and the relationships among them. The key entities in the airline system are:

* **Customer** – stores passenger information such as ID, name, and demographic details.
* **Aircraft** – contains details about aircraft models, manufacturers, and capacities.
* **Routes** – represents flight routes with origin, destination, and distance.
* **Class** – defines the travel classes such as Economy, Business, and Economy Plus.
* **Ticket Details** – captures information about booked tickets including prices, quantities, and airlines.
* **Passengers on Flights** – links customers to the specific routes and aircraft they travel on.

**Relationships**

Each customer can book multiple tickets, each ticket corresponds to a particular route and aircraft, and many passengers may share the same route. These one-to-many and many-to-one relationships ensure that the database follows normalization rules and maintains data integrity.

**Data Preparation and Normalization**

After importing datasets into the database, several cleaning steps were performed. Columns were renamed to meaningful titles, incorrect data types were corrected, and unwanted header rows were removed. The process of normalization ensured that data redundancy was minimized, and each table represented a single logical entity. This stage provided a clean foundation for executing SQL queries and analysis.

**Data Analysis and Operations**

Once the data was structured, various SQL operations were applied to extract insights.

* **Joins and Filtering:** Combined data from multiple tables, for example, displaying customers who booked tickets with specific airline brands.
* **Aggregation and Grouping:** Counted and summarized information such as the number of customers per travel class or total tickets booked.
* **Conditional Logic:** Used logical conditions to evaluate business outcomes such as whether total revenue crossed a particular threshold.
* **Window Functions:** Calculated metrics such as the maximum ticket price per class without losing detailed row-level data.
* **Rollup and Grouping:** Generated subtotals and grand totals for ticket revenue across customers and aircraft.
* **Indexing and Performance:** Created indexes on frequently searched columns to enhance query performance and reduce processing time.

These analytical steps mirrored real-world airline data processing, enabling quick and accurate insights into ticket sales, travel distance, and customer trends.

**Procedural and Advanced SQL Concepts**

The project went beyond simple queries by incorporating procedural SQL programming using PostgreSQL’s PL/pgSQL.

* **Stored Procedures:** Automated tasks such as identifying passengers who traveled within a range of routes or listing long-distance flights exceeding 2,000 miles.
* **Functions:** Returned outputs for reusable logic, for example, determining whether a customer’s travel class qualified for complimentary services.
* **Cursors:** Iterated through data to fetch specific records, such as customers whose last names ended with “Scott.”
* **Case and Conditional Statements:** Categorized flights into short, intermediate, and long-distance types based on mileage.

These procedures and functions transformed static queries into dynamic programs capable of handling decision-based logic and repetitive analysis automatically.

**ER Diagram Overview**

The ER diagram visually represents the relationships among all entities in the database.

* The **Customer** table connects to **Ticket Details** through the customer ID.
* **Ticket Details** links further to **Aircraft**, **Routes**, and **Class** tables through foreign keys.
* The **Passengers on Flights** table integrates these relationships, recording individual passenger journeys.  
  This relational design ensures referential integrity and supports efficient data retrieval for airline operations.

**Results and Findings**

Through the project, several meaningful results were obtained:

* The total ticket revenue exceeded the set business benchmark of 10,000 units.
* Passengers traveling in Business and Economy Plus classes were eligible for complimentary services.
* Most routes in the dataset fell under the Intermediate Distance category, while a smaller number were long-distance flights.
* Indexing significantly improved the speed of route-based searches.
* Stored procedures simplified repetitive analytical tasks and improved database maintainability.

**Conclusion**

The Air Cargo Database Assignment demonstrated how SQL can be used as both a data management and analytical tool. By developing normalized tables, establishing relationships, and implementing advanced SQL concepts, the project simulated a real-world airline information system.

Key learnings include:

* Understanding the process of designing normalized databases.
* Applying joins, grouping, and conditional logic for business analysis.
* Using stored procedures and functions to automate workflows.
* Employing indexing and query plans for performance optimization.

The project provided a complete, practical understanding of how relational databases can power business decision-making in the airline industry through accuracy, efficiency, and analytical insight.