**Air Cargo Analysis...**

Description

Air Cargo is an aviation company that provides air transportation services for passengers and freight. Air Cargo uses its aircraft to provide different services with the help of partnerships or alliances with other airlines. The company wants to prepare reports on regular passengers, busiest routes, ticket sales details, and other scenarios to improve the ease of travel and booking for customers.

**Project Objective:**

You, as a DBA expert, need to focus on identifying the regular customers to provide offers, analyze the busiest route which helps to increase the number of aircraft required and prepare an analysis to determine the ticket sales details. This will ensure that the company improves its operability and becomes more customer-centric and a favorable 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.

**Dataset description:**

**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 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 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

**Following operations should be performed:**

1. Write a query to create route\_details table using suitable data types for the fields, such as route\_id, flight\_num, origin\_airport, destination\_airport, aircraft\_id, and distance\_miles. Implement the check constraint for the flight number and unique constraint for the route\_id fields. Also, make sure that the distance miles field is greater than 0.

***create table route\_details***

***( route\_id int not null,***

***flight\_num int not null,***

***origin\_airport varchar(50) not null,***

***destination\_airport varchar(50) not null,***

***aircraft\_id varchar(50) not null,***

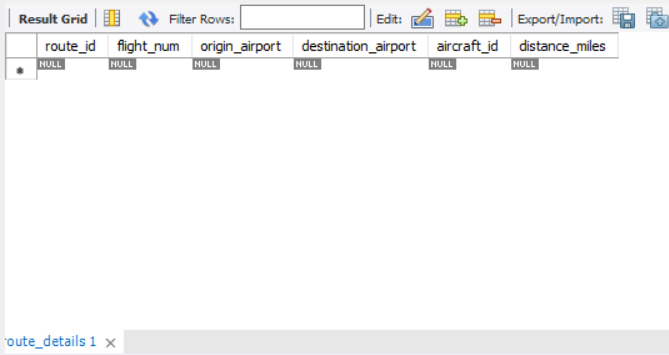
***distance\_miles int not null,***

***primary key (route\_id),***

***constraint flight\_num\_check check ((substr(flight\_num,1,2)=11)),***

***constraint route\_id\_check check (distance\_miles>0) )***

***select \* from route\_details***

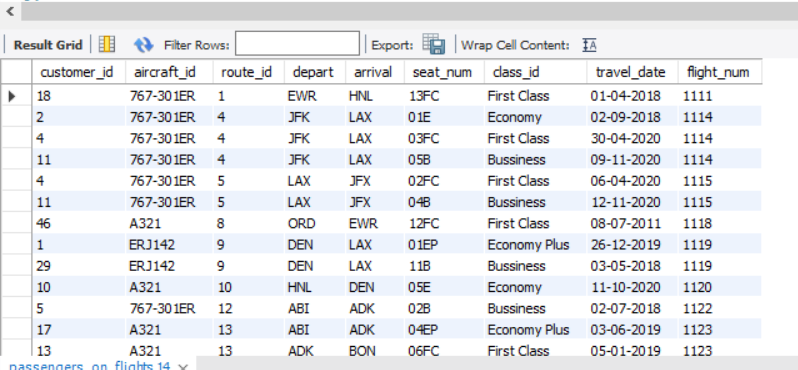
******

1. Write a query to display all the passengers (customers) who have travelled in routes 01 to 25. Take data  from the passengers\_on\_flights table.

***select \* from passengers\_on\_flights***

***where route\_id between 1 and 25***

***order by route\_id ;***

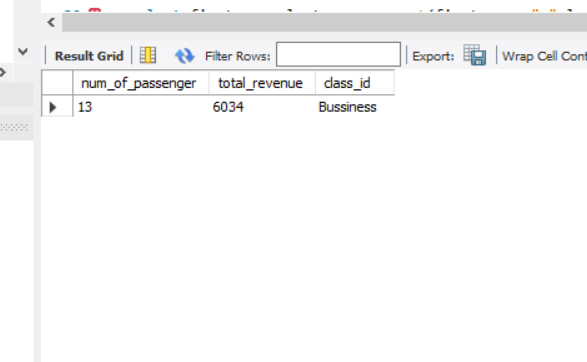
******

1. Write a query to identify the number of passengers and total revenue in business class from the ticket\_details table.

***select count(customer\_id) as num\_of\_passenger, sum(Price\_per\_ticket) as total\_revenue, class\_id***

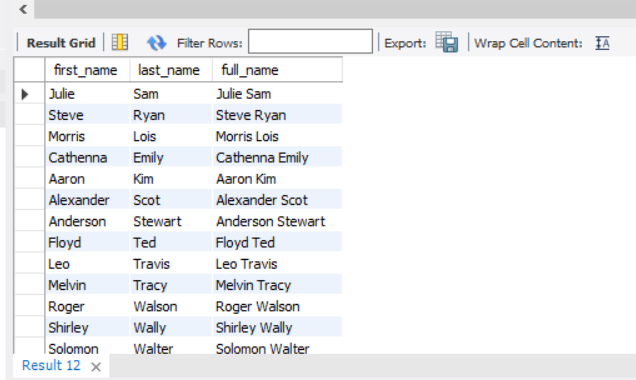
***from ticket\_details***

***where class\_id="Bussiness"***

******

1. Write a query to display the full name of the customer by extracting the first name and last name from the customer table.

***select first\_name,last\_name,concat(first\_name," ",last\_name) as full\_name from customer***

******

1. Write a query to extract the customers who have registered and booked a ticket. Use data from the customer and ticket\_details tables.

***SELECT***

***customer.customer\_id,***

***CONCAT(first\_name, " ", last\_name) AS full\_name,***

***COUNT(ticket\_details.no\_of\_tickets) AS total\_booked\_ticket***

***FROM***

***customer***

***JOIN***

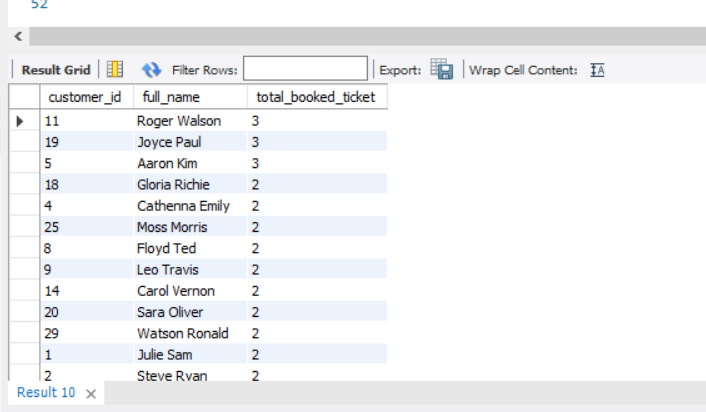
***ticket\_details USING (customer\_id)***

***GROUP BY***

***customer.customer\_id, full\_name***

***ORDER BY***

***total\_booked\_ticket DESC;***

******

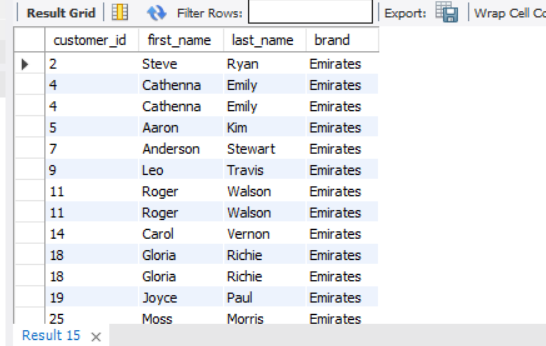
1. Write a query to identify the customer’s first name and last name based on their customer ID and brand (Emirates) from the ticket\_details table.

***select customer.customer\_id,first\_name,last\_name,ticket\_details.brand***

***from customer***

***join ticket\_details on customer.customer\_id=ticket\_details.customer\_id***

***where brand="Emirates"***

******

1. Write a query to identify the customers who have travelled by *Economy Plus* class using Group By and Having clause on the passengers\_on\_flights table.

***SELECT***

***customer.customer\_id,***

***CONCAT(customer.first\_name, " ", customer.last\_name) AS name,***

***ticket\_details.class\_id***

***FROM***

***customer***

***JOIN***

***ticket\_details***

***ON***

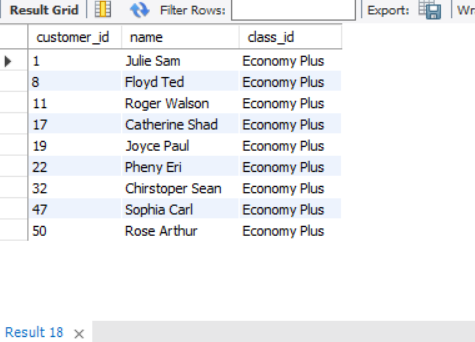
***ticket\_details.customer\_id = customer.customer\_id***

***WHERE***

***ticket\_details.class\_id = 'Economy Plus'***

***GROUP BY***

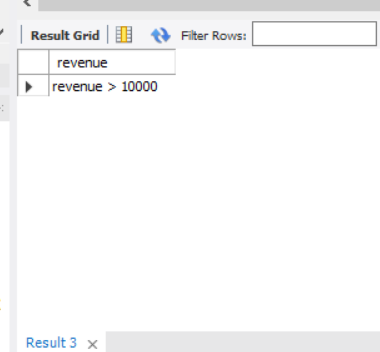
***customer.customer\_id, name, ticket\_details.class\_id;***

******

1. Write a query to identify whether the revenue has crossed 10000 using the IF clause on the ticket\_details table.

***select if( sum(no\_of\_tickets\*Price\_per\_ticket) > 10000 ,"revenue > 10000","revenue < 10000") as revenue\_status***

***from ticket\_details***

******

1. Write a query to create and grant access to a new user to perform operations on a database.

***create user 'ad'@'localhost' identified by 'password';***

***grant all privileges on \*.\* to 'ad'@'localhost' with grant option;***

1. Write a query to find the maximum ticket price for each class using window functions on the ticket\_details table.

***SELECT***

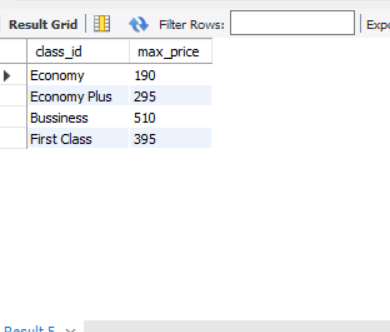
***class\_id,***

***MAX(Price\_per\_ticket) AS max\_price***

***FROM***

***ticket\_details***

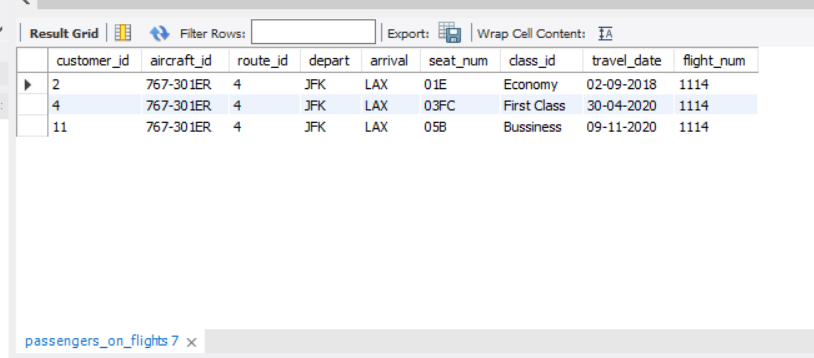
***GROUP BY class\_id***

******

1. Write a query to extract the passengers whose route ID is 4 by improving the speed and performance of the passengers\_on\_flights table.

***select \* from passengers\_on\_flights***

***where route\_id='4'***

******

1. For the route ID 4, write a query to view the execution plan of the passengers\_on\_flights table.

***CREATE VIEW passengers\_with\_route\_4 AS***

***SELECT \****

***FROM passengers\_on\_flights***

***WHERE route\_id = 4***

1. Write a query to calculate the total price of all tickets booked by a customer across different aircraft IDs using rollup function.

***SELECT***

***customer\_id,***

***aircraft\_id,***

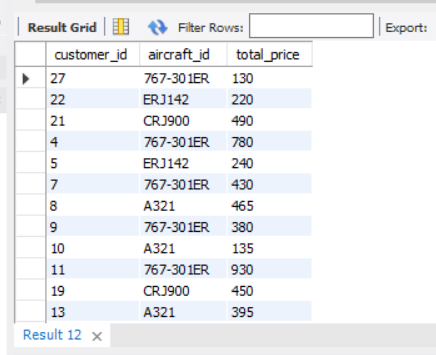
***SUM( Price\_per\_ticket\*no\_of\_tickets ) AS total\_price***

***FROM***

***ticket\_details***

***GROUP BY***

***ROLLUP(customer\_id, aircraft\_id);***



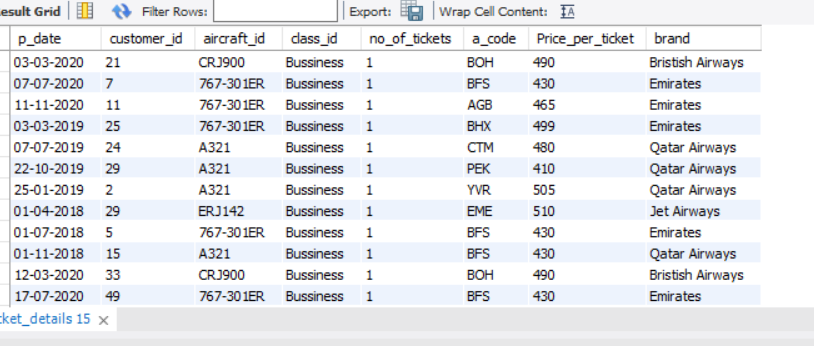
1. Write a query to create a view with only business class customers along with the brand of airlines.

***CREATE VIEW passengers\_with\_classid\_business AS***

***SELECT \****

***FROM ticket\_details***

***WHERE class\_id = 'Bussiness'***



1. Write a query to create a stored procedure to get the details of all passengers flying between a range of routes defined in run time. Also, return an error message if the table doesn't exist.

***DELIMITER //***

***CREATE PROCEDURE PASSENGERS\_BY\_ROUTES(IN min\_route INT, IN max\_route INT)***

***BEGIN***

***SELECT \* FROM passengers\_on\_flights***

***WHERE route\_id BETWEEN min\_route AND max\_route;***

***END //***

***DELIMITER ;***

1. Write a query to create a stored procedure that extracts all the details from the routes table where the travelled distance is more than 2000 miles.

***DELIMITER //***

***CREATE PROCEDURE DISTANCE\_BY\_ROUTES()***

***BEGIN***

***SELECT \* FROM routes***

***WHERE distance\_miles > 2000 ;***

***END //***

***DELIMITER ;***

1. Write a query to create a stored procedure that groups the distance travelled by each flight into three categories. The categories are, short distance travel (SDT) for >=0 AND <= 2000 miles, intermediate distance travel (IDT) for >2000 AND <=6500, and long-distance travel (LDT) for >6500.

DELIMITER //

***CREATE PROCEDURE DISTANCE\_CATEGORIES()***

***BEGIN***

***SELECT***

***flight\_num,***

***route\_id,***

***distance\_miles,***

***CASE***

***WHEN distance\_miles >= 0 AND distance\_miles <= 2000 THEN 'SDT'***

***WHEN distance\_miles > 2000 AND distance\_miles <= 6500 THEN 'IDT'***

***ELSE 'LDT'***

***END AS travel\_category***

***FROM***

***routes;***

***END //***

***DELIMITER ;***

1. Write a query to extract ticket purchase date, customer ID, class ID and specify if the complimentary services are provided for the specific class using a stored function in stored procedure on the ticket\_details table.

**Condition:**

* If the class is *Business* and *Economy Plus,* then complimentary services are given as *Yes,*else it is *No*

***DELIMITER //***

***CREATE FUNCTION complimentary\_services(class\_id VARCHAR(50))***

***RETURNS VARCHAR(3)***

***DETERMINISTIC***

***BEGIN***

***DECLARE complimentary\_services VARCHAR(3);***

***IF class\_id IN ('Business', 'Economy Plus') THEN***

***SET complimentary\_services = 'Yes';***

***ELSE***

***SET complimentary\_services = 'No';***

***END IF;***

***RETURN complimentary\_services;***

***END //***

***DELIMITER ;***

1. Write a query to extract the first record of the customer whose last name ends with Scott using a cursor from the customer table.

DELIMITER //

***CREATE PROCEDURE GetFirstCustomerWithLastNameScott()***

***BEGIN***

***DECLARE done INT DEFAULT FALSE;***

***DECLARE first\_name VARCHAR(50);***

***DECLARE last\_name VARCHAR(50);***

***DECLARE customer\_id INT;***

***DECLARE cur CURSOR FOR***

***SELECT customer\_id, first\_name, last\_name***

***FROM customer***

***WHERE last\_name LIKE '%Scott';***

***DECLARE CONTINUE HANDLER FOR NOT FOUND SET done = TRUE;***

***OPEN cur;***

***FETCH cur INTO customer\_id, first\_name, last\_name;***

***IF NOT done THEN***

***SELECT customer\_id, first\_name, last\_name;***

***END IF;***

***CLOSE cur;***

***END //***

***DELIMITER ;***