**Air Cargo Analysis.**

Course-end Project 2

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

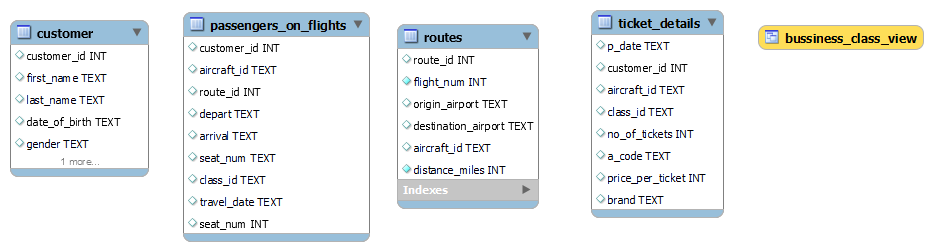
* p\_date – Ticket purchase date
* customer\_id – ID of the customer
* aircraft\_id – ID of each aircraft in a brand
* class\_id – ID of travel class
* no\_of\_tickets – Number of tickets purchased
* a\_code – Code of each airport
* price\_per\_ticket – Price of a ticket
* brand – Aviation service provider 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. Create an ER diagram for the given airlines database.



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.

**SQL Query :**

CREATE TABLE IF NOT EXISTS AIR\_CARGO.ROUTE(

ROUTE\_ID INT UNIQUE,FLIGHT\_NUM INT NOT NULL ,

ORIGIN\_AIRPORT VARCHAR(25),DESTINATION\_AIRPORT VARCHAR(25),

AIRCRAFT\_ID VARCHAR(10),DISTANCE\_MILES INT NOT NULL

CHECK (DISTANCE\_MILES>0),

CHECK(FLIGHT\_NUM>1000));

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.

**SQL Query :**

SELECT CUSTOMER\_ID,ROUTE\_ID FROM AIR\_CARGO.PASSENGERS\_ON\_FLIGHTS WHERE ROUTE\_ID BETWEEN 01 AND 25 ;

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

**SQL Query :**

SELECT COUNT(NO\_OF\_TICKETS) NO\_OF\_PASSENGERS,SUM(PRICE\_PER\_TICKET) TOTAL\_REVENUE ,CLASS\_ID FROM AIR\_CARGO.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.

**SQL Query :**

SELECT CONCAT(FIRST\_NAME,' ',LAST\_NAME) AS FULL\_NAME FROM AIR\_CARGO.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.

**SQL Query :**

SELECT CUSTOMER\_ID,CONCAT(FIRST\_NAME,' ',LAST\_NAME) AS FULL\_NAME FROM AIR\_CARGO.CUSTOMER WHERE CUSTOMER\_ID IN (SELECT DISTINCT(CUSTOMER\_ID) FROM AIR\_CARGO.TICKET\_DETAILS);

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.

**SQL Query :**

SELECT FIRST\_NAME,LAST\_NAME FROM AIR\_CARGO.CUSTOMER WHERE CUSTOMER\_ID IN (SELECT CUSTOMER\_ID FROM AIR\_CARGO.TICKET\_DETAILS 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.

**SQL Query :**

SELECT customer\_id,class\_id FROM AIR\_CARGO.PASSENGERS\_ON\_FLIGHTS where class\_id = 'ECONOMY PLUS' GROUP BY CUSTOMER\_ID ;

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

**SQL Query :**

select sum(price\_per\_ticket) revenue from air\_cargo.ticket\_details;

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

**SQL Query :**

create user 'garima' identified by 'chawda';

grant all on air\_cargo to 'garima';

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

**SQL Query :**

select class\_id,max(price\_per\_ticket) maximum\_ticket\_price from air\_cargo.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.

**SQL Query :**

select customer\_id,route\_id from air\_cargo.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.

**SQL Query :**

select aircraft\_id,sum(price\_per\_ticket) from air\_cargo.ticket\_details group by aircraft\_id with rollup;

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

**SQL Query :**

create view bussiness\_class\_view as select customer\_id,class\_id,brand from air\_cargo.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.

**SQL Query :**

delimiter //

drop procedure pro\_aircargo1;

create procedure pro\_aircargo1(route1 int,route2 int)

begin

declare continue handler for sqlstate '42SO2'

select 'sql state handler-table not found' as msg;

declare continue handler for sqlexception

begin

get diagnostics condition 1 @sqlstate=returned\_sqlstate,@errno=mysql\_errno,@text=message\_text;

set @full\_error=concat("sql exception handler-error",@errno,"(",@sqlstate,"):",@text);

select @full\_error as msg;

end;

select customer\_id,route\_id from air\_cargo.passengers\_on\_flights where route\_id between route1 and route2;

end //

delimiter //

call pro\_aircargo1(1,15);

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.

**SQL Query :**

delimiter //

create procedure pro\_routes()

begin

select \* from air\_cargo.routes where distance\_miles>2000;

end //

delimiter //

call pro\_routes();

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.

**SQL Query :**

delimiter //

drop procedure if exists pro\_distance\_category;

create procedure pro\_distance\_category(in f\_no int,out category text)

begin

declare dm int default 1;

select distance\_miles into dm from air\_cargo.routes where flight\_num=f\_no ;

if dm >=0 and dm<=2000 then set category='SDT';

elseif dm >2000 and dm<= 6500 then set category='IDT';

elseif dm > 6500 then set category='LDT';

else set category='INVALID DISTANCE';

end if;

end //

delimiter //

call pro\_distance\_category(1153,@category);

select @category;

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*

**SQL Query :**

*DELIMITER //*

*DROP FUNCTION FUNC\_COMP\_SERVICE;*

*CREATE FUNCTION FUNC\_COMP\_SERVICE(CLASS\_ID TEXT)RETURNS TEXT DETERMINISTIC*

*BEGIN*

*DECLARE COMP\_SER TEXT;*

*IF CLASS\_ID='Bussiness' or CLASS\_ID='Economy Plus' THEN SET COMP\_SER='YES';*

*ELSE SET COMP\_SER='NO';*

*END IF;*

*RETURN (COMP\_SER);*

*END //*

*DELIMITER ;*

*SELECT P\_DATE,CUSTOMER\_ID,CLASS\_ID,FUNC\_COMP\_SERVICE(CLASS\_ID) AS COMPLIMENTRY\_SERVICE FROM AIR\_CARGO.TICKET\_DETAILS ;*

*DELIMITER //*

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

**SQL Query :**

delimiter //

drop procedure pro\_cust\_record;

create procedure pro\_cust\_record()

begin

declare a int;

declare b text;

declare c text;

declare d text;

declare e text;

declare cursor1 cursor for

select customer\_id,first\_name,last\_name,date\_of\_birth,gender

from air\_cargo.customer where last\_name='scott' ;

open cursor1;

fetch cursor1 into a,b,c,d,e;

select a as customer\_id,b as first\_name,c as last\_name,d as date\_of\_birth,e as gender;

close cursor1;

end;

delimiter //

call pro\_cust\_record();