

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

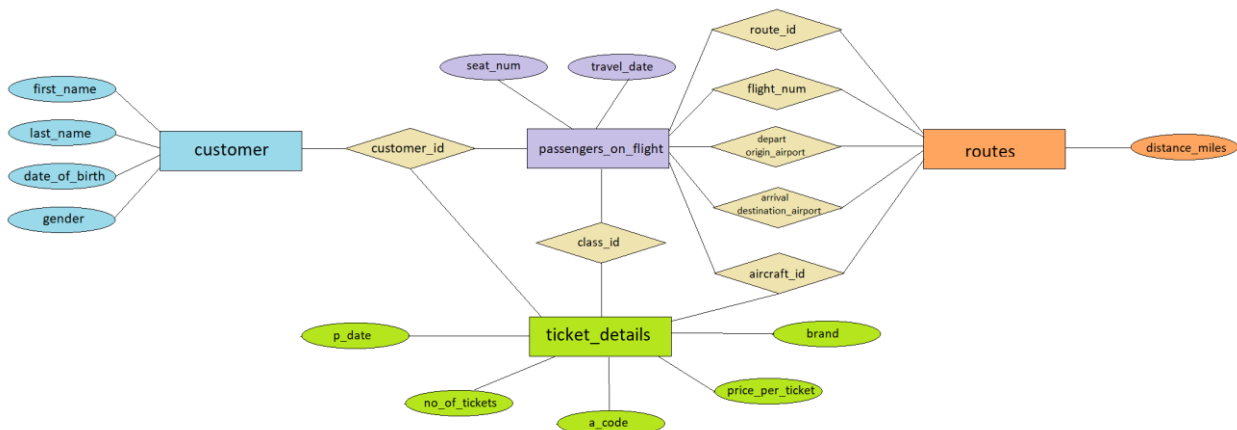
- class_id – ID of travel class
- no_of_tickets – Number of tickets purchased
- a_code – Code of each airport
- price_per_ticket – Price of a ticket
- brand – Aviation service provider for each aircraft

routes: Contains information about the route details

- Route_id – Route ID of from and to location
- Flight_num – Specific flight 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.



2. 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 DATABASE aircargo;

- All tables imported

SELECT * FROM customer;

customer_id	first_name	last_name	date_of_birth	gender
1	Julie	Sam	12-01-1989	F
2	Steve	Ryan	03-04-1983	M
3	Morris	Lois	09-12-1993	M
4	Cathenna	Emily	14-09-1977	F
5	Aaron	Kim	18-02-1991	M
6	Alexander	Scot	12-02-1985	M
7	Anderson	Stewart	11-01-1992	M
8	Floyd	Ted	21-02-1993	M
9	Leo	Travis	22-03-1994	M
10	Melvin	Tracy	23-04-1995	M

SELECT * FROM passengers_on_flights;

customer_id	aircraft_id	route_id	depart	arrival	seat_num	class_id	travel_date	flight_num
2	A321	34	CRW	COD	01B	Bussiness	26-01-2019	1117
2	767-301ER	4	JFK	LAX	01E	Economy	02-09-2018	1114
1	ERJ142	9	DEN	LAX	01EP	Economy Plus	26-12-2019	1119
1	CRJ900	30	BUR	STT	01FC	First Class	04-11-2018	1140
5	767-301ER	12	ABI	ADK	02B	Bussiness	02-07-2018	1122
5	ERJ142	18	ANI	BGR	02E	Economy	06-05-2020	1128
8	A321	38	CST	DAL	02EP	Economy Plus	09-08-2020	1148
4	767-301ER	5	LAX	JFX	02FC	First Class	06-04-2020	1115
7	767-301ER	20	BOI	03B	Bussiness	08-07-2020	1130	
5	ERJ142	22	BGR	BTT	03F	Economy	31-05-2020	1132

SELECT * FROM ticket_details;

p_date	customer_id	aircraft_id	class_id	no_of_tickets	a_code	Price_per_ticket	brand
26-12-2018	27	767-301ER	Economy	1	DAL	130	Emirates
02-02-2020	22	ERJ142	Economy Plus	1	AGB	220	Jet Airways
03-03-2020	21	CRJ900	Bussiness	1	BOH	490	Bristish Airways
04-04-2020	4	767-301ER	First Class	1	AGB	390	Emirates
05-05-2020	5	ERJ142	Economy	1	CTM	120	Jet Airways
07-07-2020	7	767-301ER	Bussiness	1	BFS	430	Emirates
08-08-2020	8	A321	Economy Plus	1	DAL	275	Qatar Airways
09-09-2020	9	767-301ER	First Class	1	BOH	380	Emirates
10-10-2020	10	A321	Economy	1	MCO	135	Qatar Airways
11-11-2020	11	767-301ER	Bussiness	1	AGR	465	Emirates

SELECT * FROM routes;

route_id	flight_num	origin_airport	destination_airport	aircraft_id	distance_miles
1	1111	EWB	HNL	767-301ER	4962
2	1112	HNL	EWB	767-301ER	4962
3	1113	EWB	LHR	A321	3466
4	1114	JFK	LAX	767-301ER	2475
5	1115	LAX	JFK	767-301ER	2475
6	1116	HNL	LAX	767-301ER	2556
7	1117	LAX	ORD	A321	1745
8	1118	ORD	EWB	A321	719
9	1119	DEN	LAX	ERJ142	862
10	1120	HNL	DEN	A321	3365

3. 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 customer_id FROM passengers_on_flights  
WHERE route_id BETWEEN 01 AND 25;
```

	customer_id
▶	2
	1
	5
	5
	4
	7
	5
	4
	11
	17
	9
	11
	10
	15
	13
	22
	24
	25
	50
	29
	44
	46
	49
	31
	18
	46

4. Write a query to identify the number of passengers and total revenue in business class from the ticket_details table.

```
SELECT COUNT(customer_id), class_id, (no_of_tickets*price_per_ticket) FROM ticket_details  
WHERE class_id = "Bussiness";
```

COUNT(customer_id)	class_id	(no_of_tickets*price_per_ticket)
13	Bussiness	490

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

```
SELECT CONCAT (first_name, " ", last_name) AS full_name  
FROM customer;
```

full_name
Julie Sam
Steve Ryan
Morris Lois
Cathenna Emily
Aaron Kim
Alexander Scot
Anderson Stewart
Floyd Ted
Leo Travis
Melvin Tracy
Roger Walson
Shirley Wally
Solomon Walter
Carol Vernon
Linda William
Chirstine Willis
Catherine Shad
Gloria Richie
Joyce Paul
Sara Oliver
Chirsty Josh
Pheny Eri
Erwin Tosh
Calvin Willis
Moss Morris
Bryan Collin
Cherly Vernon
Du plesis Chris
Watson Ronald
Donack Dukins
James Robert

6. Write a query to extract the customers who have registered and booked a ticket. Use data from the customer and ticket_details tables.

```
SELECT DISTINCT first_name, last_name FROM customer
INNER JOIN ticket_details
ON customer.customer_id = ticket_details.customer_id
WHERE ticket_details.no_of_tickets IS NOT NULL;
```

first_name	last_name
Julie	Sam
Steve	Ryan
Cathenna	Emily
Aaron	Kim
Anderson	Stewart
Floyd	Ted
Leo	Travis
Melvin	Tracy
Roger	Walson
Solomon	Walter
Carol	Vernon
Linda	William
Chirstine	Willis
Catherine	Shad
Gloria	Richie
Joyce	Paul
Sara	Oliver
Chirsty	Josh
Pheny	Eri
Calvin	Willis
Moss	Morris
Cherly	Vernon
Du plesis	Chris
Watson	Ronald
James	Robert
Chirstoper	Sean
Mark	Ethan
Kyle	Mark
Bily	Brian
Louis	Douglas
Sophia	Carl

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.

```
SELECT first_name, last_name FROM customer
RIGHT JOIN ticket_details
ON customer.customer_id = ticket_details.customer_id
WHERE ticket_details.brand = "Emirates";
```

first_name	last_name
Cherly	Vernon
Cathenna	Emily
Anderson	Stewart
Leo	Travis
Roger	Walson
Moss	Morris
Gloria	Richie
Moss	Morris
Carol	Vernon
Joyce	Paul
Gloria	Richie
Aaron	Kim
Steve	Ryan
James	Robert
Cathenna	Emily
Russell	Peter
Bily	Brian
Roger	Walson

8. 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 * FROM passengers_on_flights
```

```
WHERE class_id = "Economy Plus";
```

- I was not able to identify the attribute to GROUP BY for

customer_id	aircraft_id	route_id	depart	arrival	seat_num	class_id	travel_date	flight_num
1	ERJ142	9	DEN	LAX	01EP	Economy Plus	26-12-2019	1119
8	A321	38	CST	DAL	02EP	Economy Plus	09-08-2020	1148
11	ERJ142	31	BTM	CHA	03EP	Economy Plus	02-08-2018	1141
17	A321	13	ABI	ADK	04EP	Economy Plus	03-06-2019	1123
19	CRJ900	47	DAL	LAX	05EP	Economy Plus	13-01-2021	1157
19	CRJ900	30	BUR	STT	06EP	Economy Plus	17-12-2020	1140
22	ERJ142	22	BGR	BJI	07EP	Economy Plus	09-02-2020	1132
32	ERJ142	31	BTM	CHA	08EP	Economy Plus	04-03-2021	1141
47	CRJ900	33	CDC	CST	09EP	Economy Plus	15-12-2020	1143
50	A321	21	BFL	BET	10EP	Economy Plus	15-08-2020	1131

9. 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) FROM ticket_details) > 10000, "Yes", "No");
```

- I could not make the following work inside the IF statement

```
SELECT sum(no_of_tickets * price_per_ticket) FROM ticket_details;
```

	sum(no_of_tickets * price_per_ticket)
►	15369

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

```
GRANT privilege ON privilege_level TO new_user;
```

11. 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)
OVER (PARTITION BY class_id)
FROM ticket_details
GROUP BY class_id;
```

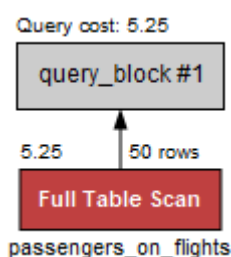
	class_id	MAX(price_per_ticket) OVER (PARTITION BY class_id)
▶	Bussiness	490
	Economy	130
	Economy Plus	220
	First Class	390

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.

```
SELECT * FROM passengers_on_flights
WHERE route_id = 4;
```

	customer_id	aircraft_id	route_id	depart	arrival	seat_num	class_id	travel_date	flight_num
▶	2	767-301ER	4	JFK	LAX	01E	Economy	02-09-2018	1114
	4	767-301ER	4	JFK	LAX	03FC	First Class	30-04-2020	1114
	11	767-301ER	4	JFK	LAX	05B	Bussiness	09-11-2020	1114

13. For the route ID 4, write a query to view the execution plan of the passengers_on_flights table.



```
EXPLAIN SELECT * FROM passengers_on_flights
WHERE route_id = 4;
```

	id	select_type	table	partitions	type	possible_keys	key	key_len	ref	rows	filtered	Extra
▶	1	SIMPLE	passengers_on_flights	HULL	ALL	HULL	HULL	HULL	HULL	50	10.00	Using where

14. 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, sum(price_per_ticket) FROM ticket_details
```


GROUP BY customer_id WITH ROLLUP;

	customer_id	sum(price_per_ticket)
▶	1	570
	2	635
	4	780
	5	670
	7	430
	8	465
	9	770
	10	135
	11	1225
	13	395
	14	290
	15	430
	16	395
	17	250
	18	565
	19	550
	20	680
	21	490
	22	220
	24	480
	25	649
	27	130
	28	170
	29	920
	31	130
	32	220
	33	490
	41	395
	44	380
	46	530
	47	225
	49	430
	50	275
	NULL	15369

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

```
CREATE VIEW class_brand AS
```

```
SELECT class_id, brand FROM ticket_details;
```

16. 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 routes_range(route1 int, route2 int)
```

```
BEGIN
```

```
SELECT * FROM routes
```

```
WHERE routes_range BETWEEN route1 AND route2;
```

```
END $$
```

17. 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 route_greater_2000()
```

```
BEGIN
```

```
SELECT * FROM routes WHERE distance_miles > 2000;
```

```
END$$
```

```
CALL route_greater_2000();
```

route_id	flight_num	origin_airport	destination_airport	aircraft_id	distance_miles
1	1111	EWB	HNL	767-301ER	4962
2	1112	HNL	EWB	767-301ER	4962
3	1113	EWB	LHR	A321	3466
4	1114	JFK	LAX	767-301ER	2475
5	1115	LAX	JFK	767-301ER	2475
6	1116	HNL	LAX	767-301ER	2556
10	1120	HNL	DEN	A321	3365
12	1122	ABI	ADK	767-301ER	4300
13	1123	ADK	BQN	A321	2232
14	1124	BQN	CAK	A321	2445
18	1128	ANI	BGR	ERJ142	2450
19	1129	ATW	AVL	A321	2222
20	1130	AVL	BOI	767-301ER	3134
21	1131	BFL	BET	A321	2425
23	1133	BLV	BFL	767-301ER	2354
25	1135	RDM	BJI	A321	2425
34	1144	CRW	COD	A321	2452
35	1145	STT	CDB	ERJ142	2121
43	1153	CBM	BOI	A321	8989
44	1154	COU	CAK	767-301ER	7676
46	1156	CDV	HNL	767-301ER	8668
48	1158	SCC	DEN	A321	5645
49	1159	DEC	ABI	A321	4533
50	1160	DRT	ORD	A321	2445

18. 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 group_by_dist(IN route_dist int, OUT dist_group varchar(45))
```

```
BEGIN
```

```
SELECT distance_miles FROM routes
```

```

WHERE distance_miles = route_dist;

IF distance_miles BETWEEN 0 AND 2000 THEN

SET dist_group = "short distance travel";

ELSEIF distance_miles BETWEEN 2001 AND 6500 THEN

SET dist_group = "intermediate distance travel";

ELSEIF distance_miles > 6500 THEN

SET dist_group = "short distance travel";

ELSE

SET dist_group = "invalid distance";

END IF;

END $$

```

19. 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 PROCEDURE complimentary_services(IN class varchar(45), OUT compl_serv varchar(45))

BEGIN

SELECT p_date, customer_id, class_id FROM ticket_details

WHERE class_id = class;

IF class_id = "Business" OR class_id = "Economy Plus" THEN

SET compl_serv = "Yes";

ELSE

SET compl_serv = "No";

END IF;

END $$

```

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.

```

SELECT * FROM customer

```

WHERE last_name = "Scott"

LIMIT 1;

	customer_id	first_name	last_name	date_of_birth	gender
▶	37	Samuel	Scott	28-01-2000	M