

Course-end Project 1 (SQL)

Air Cargo Analysis

By:

Kairos Siddhartha Kaizen

Email: kairos@kaizen.tel

Whatsapp: <https://wa.me/6587745558>

Websites (both under construction):

- <https://kaizen.tel>
- <https://hermesAI.net>

Tel: +6587745558

For:

BCG Rise 2.0

Wave 9, Group 8, DA Track

Contents

Project Description.....	3
Preliminary Activities	5
Question 1.....	6
Question 2.....	8
Question 3.....	9
Question 4.....	10
Question 5.....	11
Question 6.....	12
Question 7.....	13
Question 8.....	14
Question 9.....	15
Question 10.....	16
Question 11.....	17
Question 12.....	19
Question 13.....	21

Project Description

Air Cargo Analysis .

Course-end Project 1

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

Preliminary Activities

Database setup.

```
CREATE DATABASE air_cargo;
```

```
SHOW DATABASES;
```

```
USE air_cargo;
```

“Table Data Import Wizard” for customer.csv, passengers_on_flights.csv, routes.csv, and ticket_details.csv .

Fixing 2 typos: "Bussiness" to "Business", and "Bristish Airways" to "British Airways".

```
UPDATE ticket_details
```

```
    SET class_id = 'Business'
```

```
    WHERE class_id = 'Bussiness';
```

```
UPDATE passengers_on_flights
```

```
    SET class_id = 'Business'
```

```
    WHERE class_id = 'Bussiness';
```

```
UPDATE ticket_details
```

```
    SET brand = 'British Airways'
```

```
    WHERE brand = 'Bristish Airways';
```

Question 1

Create an ER diagram for the given airlines database.

```
ALTER TABLE routes
```

```
    ADD PRIMARY KEY (route_id);
```

```
ALTER TABLE customer
```

```
    ADD PRIMARY KEY (customer_id);
```

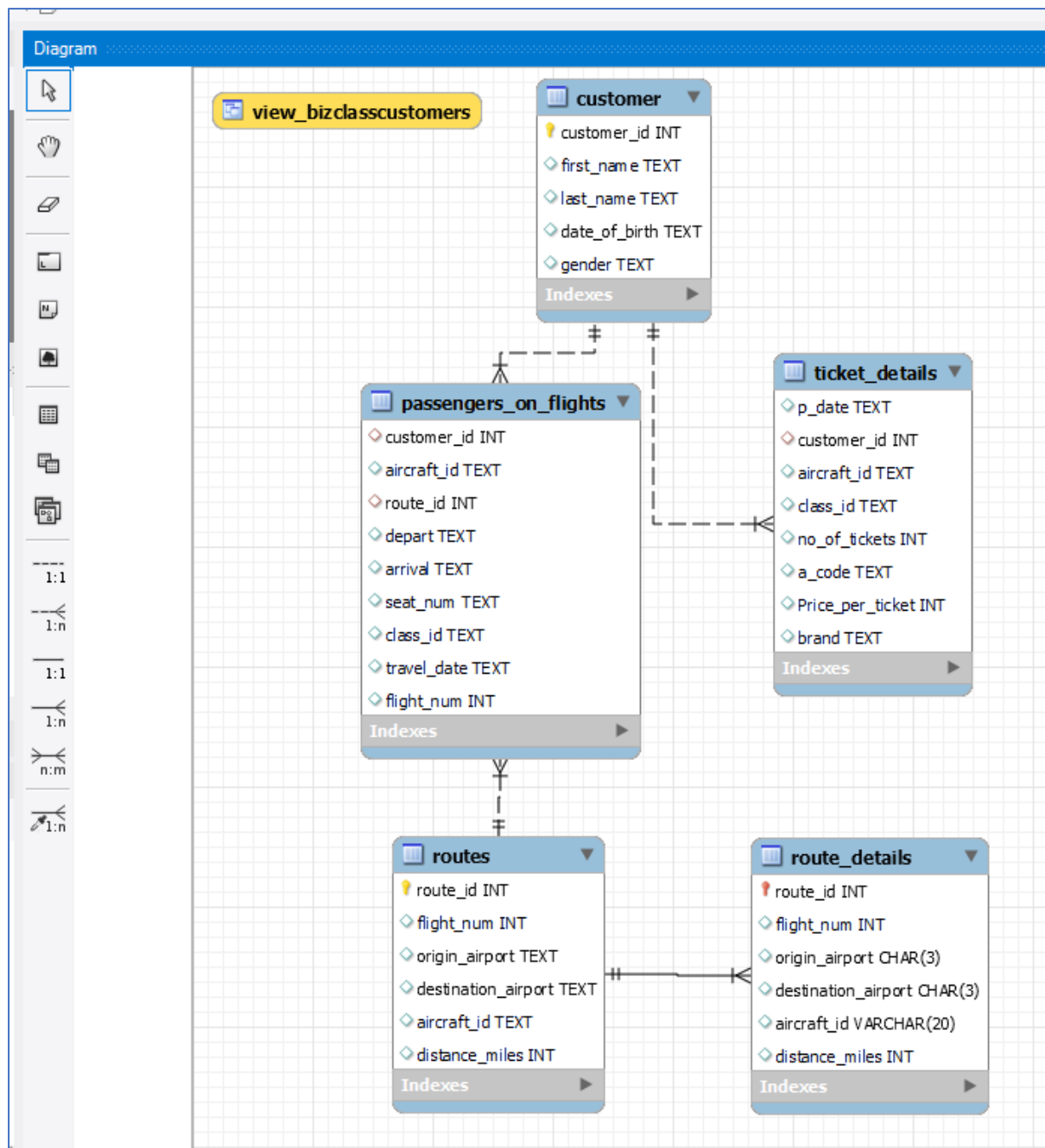
```
ALTER TABLE passengers_on_flights
```

```
    ADD FOREIGN KEY (customer_id) REFERENCES customer(customer_id),
```

```
    ADD FOREIGN KEY (route_id) REFERENCES routes(route_id);
```

```
ALTER TABLE ticket_details
```

```
    ADD FOREIGN KEY (customer_id) REFERENCES customer(customer_id);
```










Question 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 TABLE route_details (  
    route_id INT,  
    flight_num INT UNIQUE CHECK (flight_num >= 1000 AND flight_num <= 9999),  
    origin_airport CHAR(3),  
    destination_airport CHAR(3),  
    aircraft_id VARCHAR(20),  
    distance_miles INT CHECK (distance_miles > 0),  
    PRIMARY KEY (route_id),  
    FOREIGN KEY (route_id) REFERENCES routes(route_id)  
);
```

Right-click, "Alter Table":

		Table Name: <input type="text" value="route_details"/>	Schema: air_cargo							
Charset/Collation: <input type="text" value="utf8mb4"/> <input type="text" value="utf8mb4_0900_ai_ci"/>		Engine: <input type="text" value="InnoDB"/>								
Comments: <input type="text"/>										
Column Name	Datatype	PK	NN	UQ	B	UN	ZF	AI	G	Default/Expression
 route_id	INT	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="text"/>
 flight_num	INT	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NULL
 origin_airport	CHAR(3)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NULL
 destination_airport	CHAR(3)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NULL
 aircraft_id	VARCHAR(20)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NULL
 distance_miles	INT	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NULL

Question 3

Write a query to display all the passengers (customers) who have traveled in routes 01 to 25. Take data from the passengers_on_flights table.

```
SELECT DISTINCT p.customer_id, c.first_name, c.last_name, p.route_id
FROM passengers_on_flights p
LEFT JOIN customer c
ON p.customer_id = c.customer_id
WHERE p.route_id BETWEEN 1 AND 25
ORDER BY p.customer_id;
```

(26 rows returned)

Result Grid				
Filter Rows:				
	customer_id	first_name	last_name	route_id
▶	1	Julie	Sam	9
	2	Steve	Ryan	4
	4	Cathenna	Emily	5
	4	Cathenna	Emily	4
	5	Aaron	Kim	12
	5	Aaron	Kim	18
	5	Aaron	Kim	22
	7	Anderson	Stewart	20
	9	Leo	Travis	15
	10	Melvin	Tracy	10
	11	Roger	Walson	5
	11	Roger	Walson	4
	13	Solomon	Walter	13
	15	Linda	William	14
	17	Catherine	Shad	13
	18	Gloria	Richie	1
	22	Pheny	Eri	22
	24	Calvin	Willis	14
	25	Moss	Morris	23
	29	Watson	Ronald	9
	31	James	Robert	20
	44	Pat	Pat	15

Question 4

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

```
SELECT class_id, COUNT(customer_id) AS total_passengers, SUM(no_of_tickets *  
Price_per_ticket) AS total_revenue  
  
FROM ticket_details  
  
WHERE class_id = 'Business';
```

Notes:

1. data cleansing has been previously done with SET/ WHERE command for "Bussiness" to "Business", and "Bristish Airways" to "British Airways".

(1 row returned)

Result Grid			
Filter Rows: <input type="text"/>			
	class_id	total_passengers	total_revenue
▶	Business	13	6034

Question 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 DISTINCT customer_id, CONCAT(first_name, " ", last_name) AS `Full Name`  
FROM customer  
ORDER BY customer_id;
```

(50 rows returned)

Result Grid			Filter Rows:
	customer_id	Full Name	
▶	1	Julie Sam	
	2	Steve Ryan	
	3	Morris Lois	
	4	Cathenna Emily	
	5	Aaron Kim	
	6	Alexander Scot	
	7	Anderson Stewart	
	8	Floyd Ted	
	9	Leo Travis	
	10	Melvin Tracy	
	11	Roger Walson	
	12	Shirley Wally	
	13	Solomon Walter	
	14	Carol Vernon	
	15	Linda William	
	16	Chirstine Willis	
	17	Catherine Shad	
	18	Gloria Richie	
	19	Joyce Paul	
	20	Sara Oliver	
	21	Chirsty Josh	
	22	Sharon Esi	

Result 86 ×

Question 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 t.customer_id, c.first_name, c.last_name, SUM(t.no_of_tickets) AS total_tickets
FROM ticket_details t
LEFT JOIN customer c
ON t.customer_id = c.customer_id
GROUP BY t.customer_id, c.first_name, c.last_name
ORDER BY total_tickets DESC;
```

Notes:

1. I have decided to do it this way, because the Project Objective instructs us to "focus on identifying the regular customers to provide offers", thus, finding out not only who bought the tickets, but actually how many in total, seems logical, so as to be able to offer highest-recurring customers more rewards and offers.

(33 rows returned)

Result Grid			 Filter Rows:	<input type="text"/>	Export
	customer_id	first_name	last_name	total_tickets	
▶	11	Roger	Walson	3	
	19	Joyce	Paul	3	
	5	Aaron	Kim	3	
	18	Gloria	Richie	2	
	4	Cathenna	Emily	2	
	25	Moss	Morris	2	
	8	Floyd	Ted	2	
	9	Leo	Travis	2	
	14	Carol	Vernon	2	
	20	Sara	Oliver	2	
	29	Watson	Ronald	2	
	1	Julie	Sam	2	
	2	Steve	Ryan	2	
	46	Louis	Douglas	2	
	16	Chirstine	Willis	1	
	17	Catherine	Shad	1	
	27	Cherly	Vernon	1	
	24	Calvin	Willis	1	
	22	Pheny	Eri	1	
	21	Chirsty	Josh	1	
	7	Anderson	Stewart	1	
	10	Melvin	Terry	1	

Question 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 t.customer_id, c.first_name, c.last_name, t.brand, COUNT(*) as brand_usage
FROM ticket_details t
LEFT JOIN customer c
ON t.customer_id = c.customer_id
GROUP BY t.customer_id, c.first_name, c.last_name, t.brand
ORDER BY brand, brand_usage DESC;
```

Notes:

1. I have decided to do it this way, because the Project Objective instructs us to "focus on identifying the regular customers to provide offers", thus, finding out which customers have used which brand the most, would allow decision-makers to be able to better offer rewards and promos to such customers.

(41 rows returned)

Result Grid			 Filter Rows: <input type="text"/>	Export: 	Wrap Ce
	customer_id	first_name	last_name	brand	brand_usage
▶	20	Sara	Oliver	British Airways	2
	19	Joyce	Paul	British Airways	2
	21	Chirsty	Josh	British Airways	1
	1	Julie	Sam	British Airways	1
	16	Chirstine	Willis	British Airways	1
	9	Leo	Travis	British Airways	1
	33	Mark	Ethan	British Airways	1
	47	Sophia	Carl	British Airways	1
	4	Cathenna	Emily	Emirates	2
	11	Roger	Walson	Emirates	2
	25	Moss	Morris	Emirates	2
	18	Gloria	Richie	Emirates	2
	27	Cherly	Vernon	Emirates	1
	7	Anderson	Stewart	Emirates	1
	9	Leo	Travis	Emirates	1
	2	Steve	Ryan	Emirates	1
	5	Aaron	Kim	Emirates	1
	14	Carol	Vernon	Emirates	1
	19	Joyce	Paul	Emirates	1
	31	James	Robert	Emirates	1
	49	Russell	Peter	Emirates	1
	44	Pat	Pat	Emirates	1

Question 8

Write a query to identify the customers who have traveled by Economy Plus class using Group By and Having clause on the passengers_on_flights table.

```
SELECT c.customer_id, c.first_name, c.last_name, p.class_id, COUNT(*) AS total_times
FROM passengers_on_flights p
LEFT JOIN customer c
ON p.customer_id = c.customer_id
WHERE p.class_id = 'Economy Plus'
GROUP BY c.customer_id, c.first_name, c.last_name
HAVING COUNT(p.class_id) > 0
ORDER BY total_times DESC;
```

Notes:

1. I have decided to do it this way, because the Project Objective instructs us to "focus on identifying the regular customers to provide offers", thus, finding out which customers have used which "Economy Plus" the most, would allow decision-makers to be able to better offer rewards and promos to such customers.

(9 rows returned)

	customer_id	first_name	last_name	class_id	total_times
▶	19	Joyce	Paul	Economy Plus	2
	1	Julie	Sam	Economy Plus	1
	8	Floyd	Ted	Economy Plus	1
	11	Roger	Walson	Economy Plus	1
	17	Catherine	Shad	Economy Plus	1
	22	Pheny	Eri	Economy Plus	1
	32	Chirstoper	Sean	Economy Plus	1
	47	Sophia	Carl	Economy Plus	1
	50	Rose	Arthur	Economy Plus	1

Question 9

Write a query to identify whether the revenue has crossed 10000 using the IF clause on the ticket_details table.

```
SELECT
```

```
    IF(
```

```
        (SELECT SUM(no_of_tickets * Price_per_ticket) FROM ticket_details) > 10000,
```

```
        'Yes', 'No'
```

```
    ) AS `Total Revenue > $10,000?`
```

```
;
```

(1 row returned)

Result Grid		Filter Rows:
	Total Revenue > \$10,000?	
▶	Yes	

Question 10

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

```
CREATE VIEW view_BizClassCustomers AS

    SELECT p.customer_id, c.first_name, c.last_name, p.class_id, t.brand, COUNT(*) AS
    Num_of_Flights

    FROM passengers_on_flights p

    LEFT JOIN ticket_details t

    ON p.customer_id = t.customer_id AND p.aircraft_id = t.aircraft_id AND p.class_id =
    t.class_id

    LEFT JOIN customer c

    ON p.customer_id = c.customer_id

    WHERE p.class_id = 'Business'

    GROUP BY p.customer_id, c.first_name, c.last_name, t.brand

    ORDER BY Num_of_Flights DESC

;
```

Notes:

1. I have decided to do it this way, because the Project Objective instructs us to "focus on identifying the regular customers to provide offers", thus, finding out which customers have used which "Business Class" the most, would allow decision-makers to be able to better offer rewards and promos to such customers.

```
SELECT * FROM view_BizClassCustomers;
```

(12 rows returned)

	customer_id	first_name	last_name	class_id	brand	Num_of_Flights
▶	11	Roger	Walson	Business	Emirates	4
	2	Steve	Ryan	Business	Qatar Airways	1
	5	Aaron	Kim	Business	Emirates	1
	7	Anderson	Stewart	Business	Emirates	1
	15	Linda	William	Business	Qatar Airways	1
	21	Chirsty	Josh	Business	British Airways	1
	24	Calvin	Willis	Business	Qatar Airways	1
	25	Moss	Morris	Business	Emirates	1
	29	Watson	Ronald	Business	Qatar Airways	1
	29	Watson	Ronald	Business	Jet Airways	1
	33	Mark	Ethan	Business	NULL	1
	49	Russell	Peter	Business	Emirates	1

Question 11

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 proc_GetPsgrDeetsViaRoutes(IN start_route INT, IN end_route INT)
BEGIN
    # Declare variable to count num of tables
    DECLARE numTables INT;

    # Check if table exists
    SELECT COUNT(*) INTO numTables
    FROM information_schema.tables
    WHERE table_schema = 'air_cargo'
    AND table_name = 'passengers_on_flights';

    # If table exists (count > 0), run query
    IF numTables > 0 THEN
        SELECT p.*, (SELECT COUNT(*) FROM passengers_on_flights WHERE route_id =
p.route_id) AS route_count
        FROM passengers_on_flights p
        WHERE p.route_id BETWEEN start_route AND end_route
        ORDER BY route_count DESC, p.flight_num;
    # If table doesn't exist, output error message
    ELSE
        SELECT 'Error: Table passengers_on_flights does not exist' AS ErrorMessage;
    END IF;
END$$

DELIMITER ;
```

Notes:

1. I have decided to do it this way, because the Project Objective instructs us to "focus on identifying the regular customers to provide offers", thus, finding out which customers

have used which "Business Class" the most, would allow decision-makers to be able to better offer rewards and promos to such customers.

2. In this case, decision-makers will be able to see that between routes 1 to 100, route 4 and 15 have the most number of flights, and thus, marketers will be able to target those flights with more ads/ promos/ etc.

CALL proc_GetPsgrDeetsViaRoutes(1, 100);

(50 rows returned)

Result Grid Filter Rows: Export: Wrap Cell Content:										
	customer_id	aircraft_id	route_id	depart	arrival	seat_num	class_id	travel_date	flight_num	route_count
▶	2	767-301ER	4	JFK	LAX	01E	Economy	2/9/2018	1114	3
	11	767-301ER	4	JFK	LAX	05B	Business	9/11/2020	1114	3
	4	767-301ER	4	JFK	LAX	03FC	First Class	30/4/2020	1114	3
	9	767-301ER	15	CAK	ANI	04FC	First Class	10/9/2020	1125	3
	44	767-301ER	15	CAK	ANI	11FC	First Class	6/10/2020	1125	3
	49	767-301ER	15	CAK	ANI	13B	Business	19/8/2020	1125	3
	11	767-301ER	5	LAX	JFX	04B	Business	12/11/2020	1115	2
	4	767-301ER	5	LAX	JFX	02FC	First Class	6/4/2020	1115	2
	2	A321	34	CRW	COD	01B	Business	26/1/2019	1117	2
	1	ERJ142	9	DEN	LAX	01EP	Economy Plus	26/12/2019	1119	2
	29	ERJ142	9	DEN	LAX	11B	Business	3/5/2018	1119	2
	13	A321	13	ADK	BQN	06FC	First Class	5/1/2019	1123	2
	17	A321	13	ABI	ADK	04EP	Economy Plus	3/6/2019	1123	2
	24	A321	14	BQN	CAK	08B	Business	22/7/2019	1124	2
	15	A321	14	BQN	CAK	06B	Business	2/11/2018	1124	2
	7	767-301ER	20	AVL	BOI	03B	Business	8/7/2020	1130	2
	31	767-301ER	20	AVL	BOI	13E	Economy	31/12/2018	1130	2
	5	ERJ142	22	BGR	BJI	03E	Economy	31/5/2020	1132	2
	22	ERJ142	22	BGR	BJI	07EP	Economy Plus	9/2/2020	1132	2
	1	CRJ900	30	BUR	STT	01FC	First Class	4/11/2018	1140	2
	19	CRJ900	30	BUR	STT	06EP	Economy Plus	17/12/2020	1140	2
	44	ERJ142	31	DTM	CLT	03EP	Economy Plus	2/6/2018	1141	2

Question 12

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.

```
DELIMITER $$

CREATE PROCEDURE proc_GetRoutesOver2000Miles()
BEGIN
    # Declare variable to count num of tables
    DECLARE numTables INT;

    # Check if table exists
    SELECT COUNT(*) INTO numTables
    FROM information_schema.tables
    WHERE table_schema = 'air_cargo'
    AND table_name = 'routes';

    # If table exists (count > 0), run query
    IF numTables > 0 THEN
        SELECT *
        FROM routes
        WHERE distance_miles > 2000
        ORDER BY distance_miles DESC;
    # If table doesn't exist, output error message
    ELSE
        SELECT 'Error: Table routes does not exist' AS ErrorMessage;
    END IF;
END$$

DELIMITER ;
```

CALL proc_GetRoutesOver2000Miles();

(24 rows returned)

Result Grid						
		Filter Rows:			Export:	Wrap Cell Content: <input type="checkbox"/>
	route_id	flight_num	origin_airport	destination_airport	aircraft_id	distance_miles
▶	43	1153	CBM	BOI	A321	8989
	46	1156	CDV	HNL	767-301ER	8668
	44	1154	COU	CAK	767-301ER	7676
	48	1158	SCC	DEN	A321	5645
	1	1111	EWR	HNL	767-301ER	4962
	2	1112	HNL	EWR	767-301ER	4962
	49	1159	DEC	ABI	A321	4533
	12	1122	ABI	ADK	767-301ER	4300
	3	1113	EWR	LHR	A321	3466
	10	1120	HNL	DEN	A321	3365
	20	1130	AVL	BOI	767-301ER	3134
	6	1116	HNL	LAX	767-301ER	2556
	4	1114	JFK	LAX	767-301ER	2475
	5	1115	LAX	JFK	767-301ER	2475
	34	1144	CRW	COD	A321	2452
	18	1128	ANI	BGR	ERJ142	2450
	14	1124	BQN	CAK	A321	2445
	50	1160	DRT	ORD	A321	2445
	21	1131	BFL	BET	A321	2425
	25	1135	RDM	BJI	A321	2425
	23	1133	BLV	BFL	767-301ER	2354
	13	1123	ADK	BQN	A321	2333

Question 13

Write a query to create a stored procedure that groups the distance traveled 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 proc_CatFlightDistance()
BEGIN
    # Declare variable to count num of tables
    DECLARE numTables INT;

    # Check if table exists
    SELECT COUNT(*) INTO numTables
    FROM information_schema.tables
    WHERE table_schema = 'air_cargo'
    AND table_name = 'routes';

    # If table exists (count > 0), run query
    IF numTables > 0 THEN
        SELECT *,
        CASE
            WHEN distance_miles >= 0 AND distance_miles <= 2000 THEN 'SDT'
            WHEN distance_miles > 2000 AND distance_miles <= 6500 THEN 'IDT'
            WHEN distance_miles > 6500 THEN 'LDT'
        END AS DistanceCategory
        FROM routes
        ORDER BY distance_miles DESC;
    # If table doesn't exist, output error message
    ELSE
        SELECT 'Error: Table routes does not exist' AS ErrorMessage;
    END IF;
END$$

DELIMITER ;
```

CALL proc_CatFlightDistance();

(49 rows returned)

Result Grid		Filter Rows:	Export:	Wrap Cell Content:			
	route_id	flight_num	origin_airport	destination_airport	aircraft_id	distance_miles	DistanceCategory
▶	43	1153	CBM	BOI	A321	8989	LDT
	46	1156	CDV	HNL	767-301ER	8668	LDT
	44	1154	COU	CAK	767-301ER	7676	LDT
	48	1158	SCC	DEN	A321	5645	IDT
	1	1111	EWR	HNL	767-301ER	4962	IDT
	2	1112	HNL	EWR	767-301ER	4962	IDT
	49	1159	DEC	ABI	A321	4533	IDT
	12	1122	ABI	ADK	767-301ER	4300	IDT
	3	1113	EWR	LHR	A321	3466	IDT
	10	1120	HNL	DEN	A321	3365	IDT
	20	1130	AVL	BOI	767-301ER	3134	IDT
	6	1116	HNL	LAX	767-301ER	2556	IDT
	4	1114	JFK	LAX	767-301ER	2475	IDT
	5	1115	LAX	JFK	767-301ER	2475	IDT
	34	1144	CRW	COD	A321	2452	IDT
	18	1128	ANI	BGR	ERJ142	2450	IDT
	14	1124	BQN	CAK	A321	2445	IDT
	50	1160	DRT	ORD	A321	2445	IDT
	25	1135	RDM	BJI	A321	2425	IDT
	21	1131	BFL	BET	A321	2425	IDT
	23	1133	BLV	BFL	767-301ER	2354	IDT
	13	1123	ADK	BOI	A321	2323	IDT

END OF DOCUMENT.