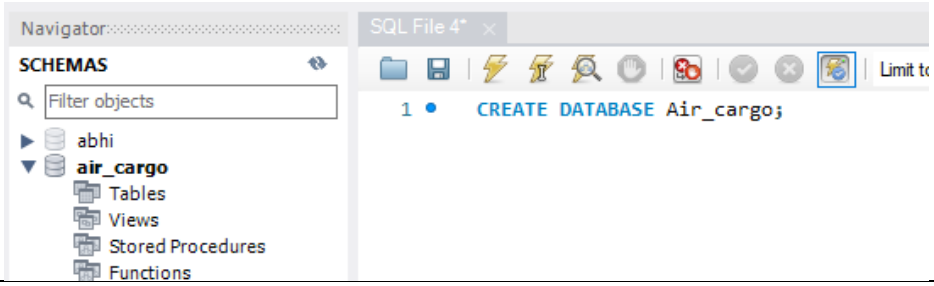
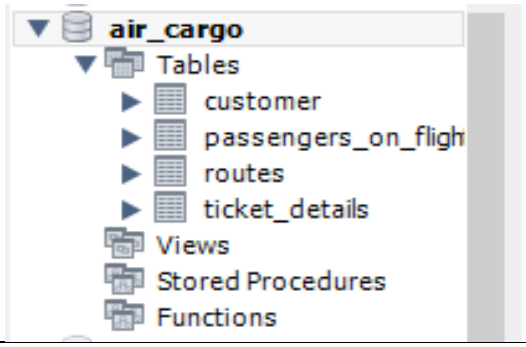
AIR CARGO ANALYSIS - SQL

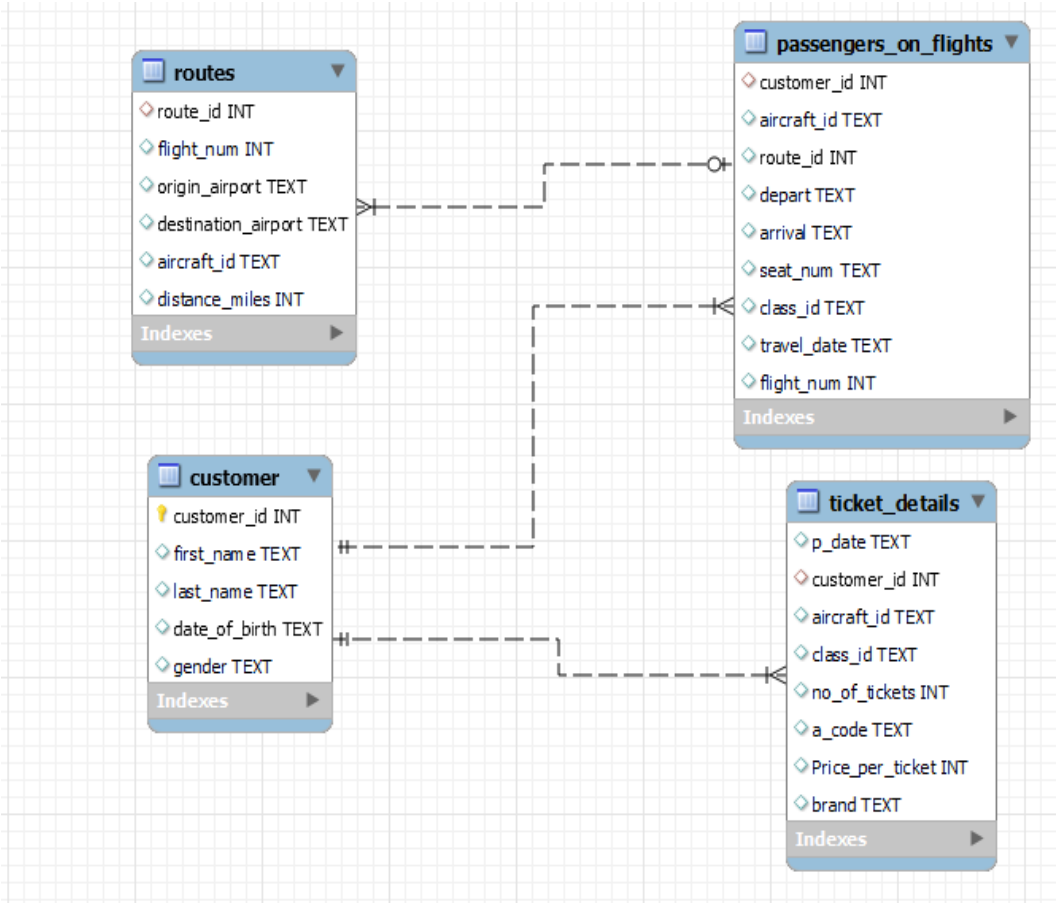
Created database Air_cargo –



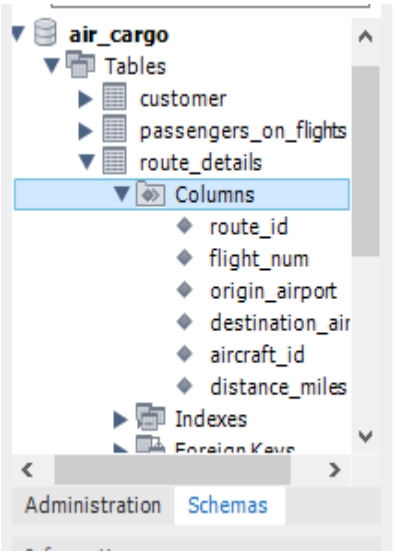
Imported available datasets in database –



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 TABLE route_details
    (route_id INT,
    flight_num INT,
    origin_airport varchar(15),
    destination_airport varchar(15),
    aircraft_id varchar(15),
    distance_miles INT,
    check(flight_num is NOT NULL),
    unique(route_id),
    check(distance_miles>0)
    );
```

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 *
FROM passengers_on_flights
WHERE route_id BETWEEN 1 and 25;

#alternate query

SELECT *
FROM passengers_on_flights
WHERE route_id >=1 and route_id <=25;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

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
1	ERJ142	9	DEN	LAX	01EP	Economy Plus	26-12-2019	1119
5	767-301ER	12	ABI	ADK	02B	Bussiness	02-07-2018	1122
5	ERJ142	18	ANI	BGR	02E	Economy	06-05-2020	1128
4	767-301ER	5	LAX	JFX	02FC	First Class	06-04-2020	1115
7	767-301ER	20	AVL	BOI	03B	Bussiness	08-07-2020	1130
5	ERJ142	22	BGR	BJI	03E	Economy	31-05-2020	1132
4	767-301ER	4	JFK	LAX	03FC	First Class	30-04-2020	1114
11	767-301ER	5	LAX	JFX	04B	Bussiness	12-11-2020	1115
17	A321	13	ABI	ADK	04EP	Economy Plus	03-06-2019	1123
9	767-301ER	15	CAK	ANI	04FC	First Class	10-09-2020	1125
11	767-301ER	4	JFK	LAX	05B	Bussiness	09-11-2020	1114

passengers_on_flights 2

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) AS pax_count, SUM(Price_per_ticket) AS Revenue, class_id
FROM ticket_details
WHERE class_id = "Bussiness";
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

pax_count	Revenue	class_id
13	6034	Bussiness

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 customer_name
FROM customer;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

customer_name
Julie Sam
Steve Ryan
Morris Lois
Cathenna Emily
Aaron Kim
Alexander Scot
Anderson Stewart
Floyd Ted
Leo Travis
Melvin Tracy

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 c.customer_id, concat(c.first_name, " ", c.last_name) AS customer_name,
   COUNT(t.no_of_tickets) AS tickets_booked
FROM customer c
INNER JOIN ticket_details t
ON c.customer_id = t.customer_id
GROUP BY c.customer_id, customer_name
ORDER BY tickets_booked DESC;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

customer_id	customer_name	tickets_booked
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

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 DISTINCT(c.customer_id), c.first_name, c.last_name, t.brand
FROM customer c
LEFT JOIN ticket_details t
ON c.customer_id = t.customer_id
WHERE brand = "Emirates"
ORDER BY c.customer_id;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

customer_id	first_name	last_name	brand
2	Steve	Ryan	Emirates
4	Cathenna	Emily	Emirates
5	Aaron	Kim	Emirates
7	Anderson	Stewart	Emirates
9	Leo	Travis	Emirates
11	Roger	Walson	Emirates
14	Carol	Vernon	Emirates
18	Gloria	Richie	Emirates
19	Joyce	Paul	Emirates
25	Moss	Morris	Emirates
27	Cherly	Vernon	Emirates
31	James	Robert	Emirates
44	Bily	Brian	Emirates
49	Russell	Peter	Emirates

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 DISTINCT(c.customer_id), c.first_name, c.last_name, p.class_id
FROM customer c
RIGHT JOIN passengers_on_flights p
ON c.customer_id = p.customer_id
GROUP BY 1,2,3,4
HAVING class_id = "Economy Plus";
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

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

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

```
SELECT SUM(no_of_tickets*Price_per_ticket) AS total_revenue,
IF (SUM(no_of_tickets*Price_per_ticket) > 10000, "YES", "NO") AS revenue_more_than_10000
FROM ticket_details;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

total_revenue	revenue_more_than_10000
15369	YES

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

```
CREATE USER 'ABHI'@'localhost' IDENTIFIED BY 'password';
GRANT ALL PRIVILEGES ON *.* TO 'ABHI'@'localhost' WITH GRANT OPTION;
```

Output

Action Output

#	Time	Action	Message
69	15:47:05	CREATE USER 'ABHI'@'localhost' IDENTIFIED BY 'password'	0 row(s) affected
70	15:47:06	GRANT ALL PRIVILEGES ON *.* TO 'ABHI'@'localhost' WITH GRANT OPTION	0 row(s) affected

11. Write a query to find the maximum ticket price for each class using window functions on the ticket_details table.

```
SELECT DISTINCT(class_id), MAX(Price_per_ticket) OVER(partition by class_id) max_ticket_price
FROM ticket_details
ORDER BY max_ticket_price DESC;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

class_id	max_ticket_price
Bussiness	510
First Class	395
Economy Plus	295
Economy	190

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.

```
CREATE INDEX route ON passengers_on_flights (route_id);
    SELECT customer_id, aircraft_id, depart, arrival
    FROM passengers_on_flights
    WHERE route_id = 4;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

customer_id	aircraft_id	depart	arrival
2	767-301ER	JFK	LAX
4	767-301ER	JFK	LAX
11	767-301ER	JFK	LAX

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

```
SELECT aircraft_id, route_id, depart, arrival, travel_date, flight_num
FROM passengers_on_flights
WHERE route_id = 4;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

aircraft_id	route_id	depart	arrival	travel_date	flight_num
767-301ER	4	JFK	LAX	02-09-2018	1114
767-301ER	4	JFK	LAX	30-04-2020	1114
767-301ER	4	JFK	LAX	09-11-2020	1114

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, aircraft_id, SUM(Price_per_ticket) AS total_ticket_Price
FROM ticket_details
GROUP BY customer_id, aircraft_id WITH ROLLUP;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

customer_id	aircraft_id	total_ticket_Price
1	CRJ900	320
1	ERJ142	250
1	NULL	570
2	767-301ER	130
2	A321	505
2	NULL	635
4	767-301ER	780
4	NULL	780
5	767-301ER	430

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

```
CREATE OR REPLACE VIEW Business_Class
AS
SELECT t.customer_id, c.first_name, c.last_name, t.class_id, t.brand
FROM ticket_details t
LEFT JOIN customer c
ON t.customer_id = c.customer_id
WHERE class_id="Bussiness"
ORDER BY t.customer_id;
SELECT * FROM air_cargo.business_class;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

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

business_class 37

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.

```
DROP PROCEDURE IF EXISTS passengersBetweenRoutes;
DELIMITER &&
CREATE PROCEDURE passengersBetweenRoutes(IN route_a INT, IN route_b INT)
BEGIN
DECLARE EXIT HANDLER FOR SQLEXCEPTION
  BEGIN
    GET DIAGNOSTICS CONDITION 1
    @sqlstate = RETURNED_SQLSTATE,
    @errno = MYSQL_ERRNO,
    @text = MESSAGE_TEXT;
    SET @full_error = CONCAT("SQLEXCEPTION Handler - ERROR ", @errno, " (", @sqlstate, "): ", @text);
    SELECT @full_error AS msg;
  END;
  SELECT * FROM passengers_on_flights WHERE route_id BETWEEN route_a AND route_b;
END &&
CALL passengersBetweenRoutes(10,20);
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

customer_id	aircraft_id	route_id	depart	arrival	seat_num	class_id	travel_date	flight_num
10	A321	10	HNL	DEN	05E	Economy	11-10-2020	1120
5	767-301ER	12	ABI	ADK	02B	Bussiness	02-07-2018	1122
17	A321	13	ABI	ADK	04EP	Economy Plus	03-06-2019	1123
13	A321	13	ADK	BQN	06FC	First Class	05-01-2019	1123
15	A321	14	BQN	CAK	06B	Bussiness	02-11-2018	1124
24	A321	14	BQN	CAK	08B	Bussiness	22-07-2019	1124

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 $$
CREATE PROCEDURE `distance` ()
BEGIN
SELECT * FROM routes
WHERE distance_miles > 2000
ORDER BY distance_miles;
END$$
CALL distance;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

route_id	flight_num	origin_airport	destination_airport	aircraft_id	distance_miles
35	1145	STT	CDB	ERJ142	2121
19	1129	ATW	AVL	A321	2222
13	1123	ADK	BQN	A321	2232
23	1133	BLV	BFL	767-301ER	2354
21	1131	BFL	BET	A321	2425
25	1135	RDM	BJI	A321	2425
14	1124	BQN	CAK	A321	2445
50	1160	DRT	ORD	A321	2445
18	1128	ANI	BGR	ERJ142	2450
34	1144	CRW	COD	A321	2452

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 $$
CREATE PROCEDURE `distance_group`(IN flight_number INT)
BEGIN
SELECT *,
    CASE
    WHEN distance_miles>=0 AND distance_miles<=2000 THEN "SDT"
    WHEN distance_miles>2000 AND distance_miles<=6500 THEN "IDT"
    ELSE "LDT"
END as category
FROM routes
WHERE flight_num = flight_number;
END$$
CALL distance_group(1115);
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

route_id	flight_num	origin_airport	destination_airport	aircraft_id	distance_miles	category
5	1115	LAX	JFK	767-301ER	2475	IDT

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.
    <u>Condition</u>:
    If the class is Business and Economy Plus, then complimentary services are given as Yes, else it is No

```
DELIMITER $$
CREATE PROCEDURE `complimentary_service`()
BEGIN
SELECT p_date, customer_id, class_id,
    CASE
    WHEN class_id = "Bussiness" OR class_id = "Economy Plus" THEN "YES"
    ELSE "NO"
    END as Complimentary_Service
FROM ticket_details
ORDER BY customer_id;
END$$

CALL complimentary_service ();
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

p_date	customer_id	class_id	Complimentary_Service
01-10-2018	1	First Class	NO
23-11-2019	1	Economy Plus	YES
01-09-2018	2	Economy	NO
25-01-2019	2	Bussiness	YES
29-04-2020	4	First Class	NO
04-04-2020	4	First Class	NO
05-05-2020	5	Economy	NO
01-07-2018	5	Bussiness	YES
30-05-2020	5	Economy	NO
07-07-2020	7	Bussiness	YES

Result 49

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.

```
DELIMITER $$
CREATE PROCEDURE scottAsLastName()
BEGIN DECLARE a VARCHAR(255);
DECLARE b VARCHAR(255);
DECLARE cursor_1 CURSOR FOR SELECT first_name, last_name FROM customer
WHERE last_name = 'Scott';
OPEN cursor_1;
REPEAT FETCH cursor_1 INTO a,b;
UNTIL b = 0 END REPEAT;
SELECT a as first_name, b as last_name;
CLOSE cursor_1;
END;
$$ DELIMITER ;
call scottAsLastName();
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

first_name	last_name
Samuel	Scott

** END **