Code:

Q2: 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 d, 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.

```
-Q2-
DROP TABLE route_details;
CREATE TABLE route_details(route_id INT(10) UNIQUE ,flight_num INT(10) CHECK(flight_num >1000),
origin_airport VARCHAR(225),destination_airport VARCHAR(225),aircraft_id VARCHAR(225),
distance_miles INT(10) CHECK(distance_miles>0) );
```

Q3: 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.

```
-Q3-
SELECT C.first_name FROM passengers_on_flights_csv P LEFT JOIN customer_csv C
ON(C.customer_id=P.customer_id) WHERE `route_id` BETWEEN 1 AND 25;
```

Q4: Write a query to identify the number of passengers and total revenue in business class from the ticket_details table.

```
-Q4-
SELECT COUNT(customer_id)AS No_of_Customers, SUM(`Price_per_ticket`)AS Total_Price
FROM `ticket_details_csv` WHERE`class_id`='Bussiness';
```

Q5: Write a query to display the full name of the customer by extracting the first name and last name from the customer table.

```
-Q5-
SELECT CONCAT(first_name,last_name) AS Full_name FROM customer_csv
```

Q6: Write a query to extract the customers who have registered and booked a ticket. Use data from the customer and ticket_details tables.

```
-Q6-
SELECT DISTINCT(C.customer_id) FROM ticket_details_csv T LEFT JOIN customer_csv C
ON (C.customer_id = T.customer_id) WHERE T.customer_id IS NOT NULL;
```

Q7: 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.

```
-Q7-
SELECT CONCAT(C.first_name,C.last_name) AS Full_name FROM customer_csv C LEFT JOIN ticket_details_csv T
ON(C.customer_id = T.customer_id) WHERE T.brand='Emirates' ORDER BY C.customer_id,T.brand;
```

Q8: 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.

```
-Q8-
SELECT COUNT(customer_id) AS Total_Customers FROM passengers_on_flights_csv
GROUP BY class_id HAVING class_id="Economy Plus";
```

Q9: Write a query to identify whether the revenue has crossed 10000 using the IF clause on the ticket_details table.

```
-Q9-
SELECT IF(SUM(Price_per_ticket)>10000,"Yes Revenue has Crossed 10000",
"no Revenue has Crossed not 10000") AS Total_Revenue FROM `ticket_details_csv`
```

Q10: Write a query to create and grant access to a new user to perform operations on a database.

```
-Q10-
USE `air_cargo_project`;
GRANT ALL ON *.* TO 'user'@'localhost';
#FLUSH PRIVILEGES;
```

Q11: Write a query to find the maximum ticket price for each class using window functions on the ticket_details table.

```
-Q11-
SELECT customer_id,  class_id , MAX(Price_per_ticket) OVER(PARTITION BY class_id)
 AS Max_Price FROM ticket_details_csv;
```

Q12: Write a query to extract the passengers whose route ID is 4 by improving the speed and performance of the passengers_on_fiights table.

```
-Q12-
SELECT customer_id FROM `passengers_on_flights_csv` WHERE route_id=4;
```

Q13: For the route ID 4, write a query to view the execution plan of the passengers_on_flights table.

```
-Q13-
SELECT * FROM `passengers_on_flights_csv` WHERE route_id=4;
```

Q14: Write a query to calculate the total price of all tickets booked by a customer across different aircraft IDs using rollup function.

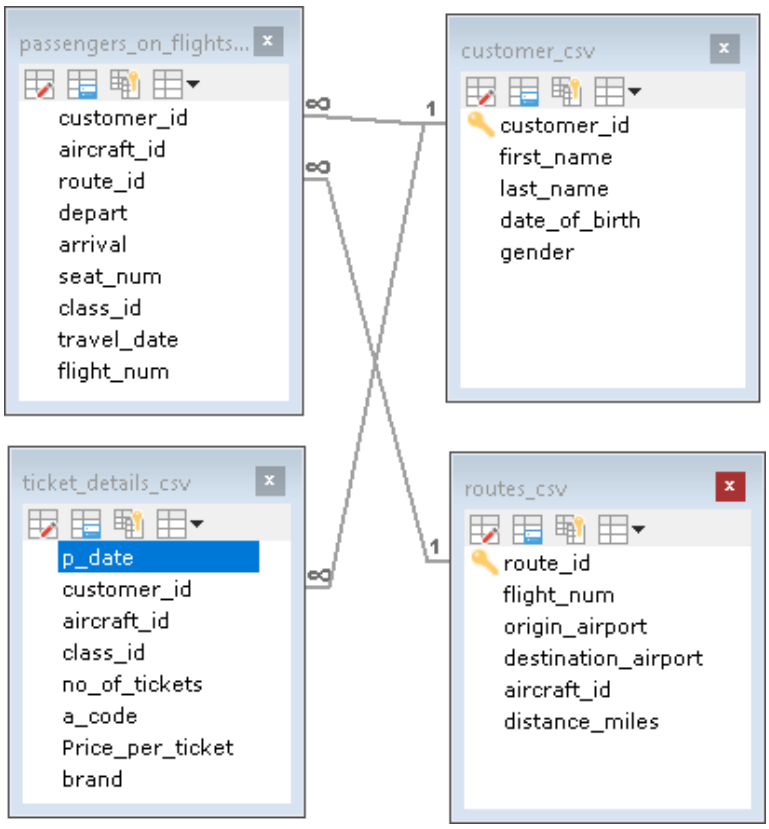
```
-Q14-
SELECT customer_id,aircraft_id,SUM(Price_per_ticket)AS Total_sales
FROM ticket_details_csv GROUP BY customer_id,aircraft_id WITH ROLLUP;
```

Q15: Write a query to create a view with only business class customers along with the brand of airlines.

```
-Q15-
DROP VIEW Bussiness_Class;
CREATE VIEW Bussiness_Class AS
SELECT customer_id,brand FROM `ticket_details_csv` WHERE class_id='Bussiness';
SELECT * FROM Bussiness_Class;
```

Output:

Q1: Create an ER diagram for the given airlines database.



Q2: 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 d, 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.

```
2 queries executed, 2 success, 0 errors, 1 warnings

Query: DROP TABLE route_details

0 row(s) affected

Execution Time : 0.357 sec
Transfer Time  : 1.045 sec
Total Time     : 1.403 sec
------------------------------------------------------------

Query: CREATE TABLE route_details(route_id INT(10) UNIQUE ,flight_num INT(10) CHECK(flight_num >1000),origin_airport
VARCHAR(225),desti...

0 row(s) affected, 3 warning(s)

Execution Time : 0.554 sec
Transfer Time  : 1.056 sec
Total Time     : 1.610 sec

Note: To see complete list of warning(s), enable Tools -> Preferences -> General -> Show Warning(s) in Messages Tab
```

Q3: 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.

first_name
Aaron
Cathenna
Anderson
Aaron
Cathenna
Roger
Catherine
Leo
Roger
Melvin
Linda
Solomon
Pheny
Calvin

```
SELECT C.first_name FROM passengers_on_flights_csv P LEFT JOIN customer_csv C ON(C.customer_id=P.customer_id) WHERE `route_id` BETWEEN 1 AND 2...
```

Q4: Write a query to identify the number of passengers and total revenue in business class from the ticket_details table.

No_of_Customers	Total_Price
13	6034

Q5: Write a query to display the full name of the customer by extracting the first name and last name from the customer table.

Full_name
JulieSam
SteveRyan
MorrisLois
CathennaEmily
AaronKim
AlexanderScot
AndersonStewart
FloydTed
LeoTravis
MelvinTracy
RogerWalson
ShirleyWally
SolomonWalter
CarolVernon

```
SELECT CONCAT(first_name,last_name) AS Full_name FROM customer_csv LIMIT 0, 1000
```

Q6: Write a query to extract the customers who have registered and booked a ticket. Use data from the customer and ticket_details tables.

customer_id
1
2
4
5
7
8
9
10
11
13
14
15
16
17

```
SELECT DISTINCT(C.customer_id) FROM ticket_details_csv T LEFT JOIN
```

Q7: 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.

Full_name
JulieSam
SteveRyan
MorrisLois
CathennaEmily
AaronKim
AlexanderScot
AndersonStewart
FloydTed
LeoTravis
MelvinTracy
RogerWalson
ShirleyWally
SolomonWalter
CarolVernon

```
SELECT CONCAT(first_name,last_name) AS Full_name FROM customer_csv LIMIT 0, 1000
```

Q8: 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.

Total_Customers
10

Q9: Write a query to identify whether the revenue has crossed 10000 using the IF clause on the ticket_details table.

Total_Revenue
Yes Revenue has Crossed 10000

Q10: Write a query to create and grant access to a new user to perform operations on a database.

```
Query: USE `air_cargo_project`

0 row(s) affected

Execution Time : 0.027 sec
Transfer Time  : 0 sec
Total Time     : 0.028 sec
-------------------------------------------------------------

Query: GRANT ALL ON *.* TO 'user'@'localhost'

Error Code: 1045
Access denied for user 'vasu'@'%' (using password: YES)

Execution Time : 0 sec
Transfer Time  : 0 sec
Total Time     : 0.044 sec
-------------------------------------------------------------
```

Q11: Write a query to find the maximum ticket price for each class using window functions on the ticket_details table.

customer_id	class_id	Max_Price
25	Bussiness	510
49	Bussiness	510
21	Bussiness	510
33	Bussiness	510
29	Bussiness	510
7	Bussiness	510
24	Bussiness	510
15	Bussiness	510
2	Bussiness	510
11	Bussiness	510
29	Bussiness	510
5	Bussiness	510

```
SELECT customer_id, class_id , MAX(Price_per_ticket) OVER(PARTITION BY class_id) as Max_Price
```

Q12: Write a query to extract the passengers whose route ID is 4 by improving the speed and performance of the passengers_on_fiights table.

customer_id
2
4
11

Q13: For the route ID 4, write a query to view the execution plan of the passengers_on_flights table.

customer_id	aircraft_id	route_id	depart	arrival	seat_num	class_id	travel_date	flight_num
2	767-301ER	4	JFK	LAX	01E	Economy	2018-09-02 00:00:00.000000	1114
4	767-301ER	4	JFK	LAX	03FC	First Class	2020-04-30 00:00:00.000000	1114
11	767-301ER	4	JFK	LAX	05B	Bussiness	2020-11-09 00:00:00.000000	1114

Q14: Write a query to calculate the total price of all tickets booked by a customer across different aircraft IDs using rollup function.

customer_id	aircraft_id	Total_sales
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
5	ERJ142	240
5	*(NULL)*	670
7	767-301ER	430

```
SELECT customer_id,aircraft_id,SUM(Price_per_ticket)AS Total_sales FROM ticket_
```

Q15: Write a query to create a view with only business class customers along with the brand of airlines.

```
1 queries executed, 1 success, 0 errors, 0 warnings

Query: CREATE VIEW Bussiness_Class AS SELECT customer_id,brand FROM `ticket_details_csv` WHERE class_id='Bussiness'

0 row(s) affected

Execution Time : 0.245 sec
Transfer Time  : 1.044 sec
Total Time     : 1.289 sec
```