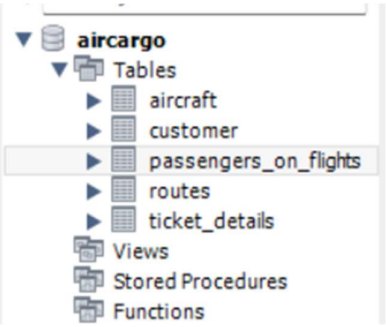
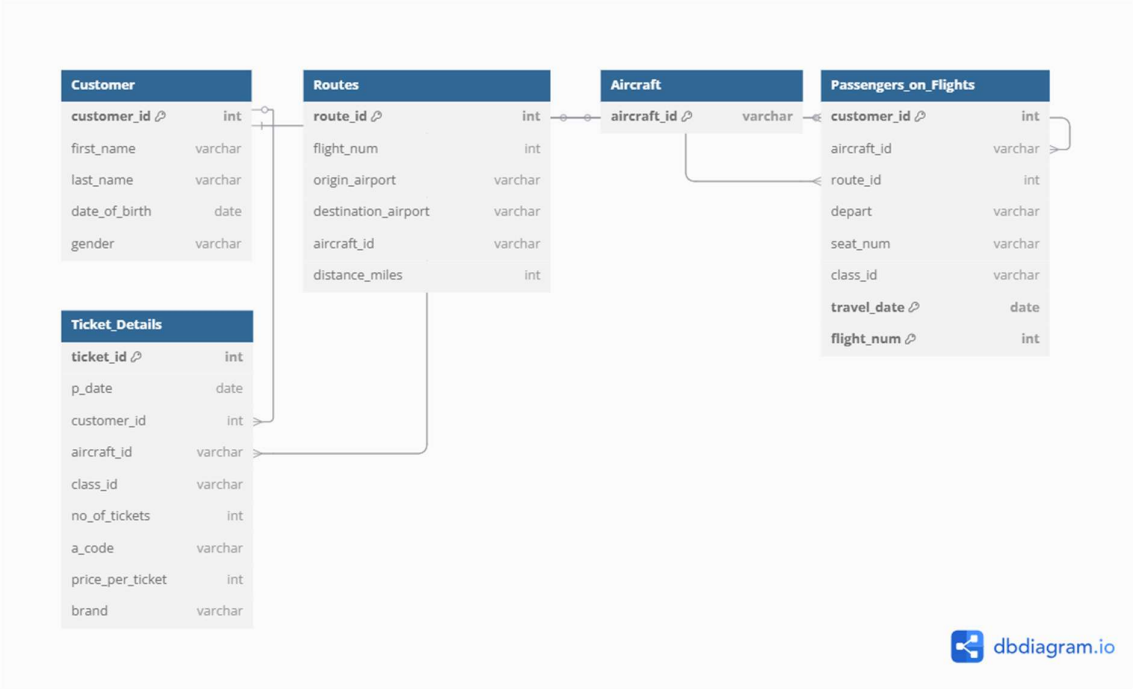
Task 1st **Create a database named AirCargo and import ticket_details.csv, routes.csv, passengers_on_flights.csv, and customer.csv from the given resources into it**

```
create database AirCargo;
use AirCargo;
```



Task 2nd **Create an ER diagram for the given airlines' database.**



Task 3rd Write a query to display all the passengers who have traveled on routes 01 to 25 from the passengers_on_flights table.

```
Select *
from passengers_on_flights
where route_id between 01 and 25;
```

Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell Content:

customer_id	aircraft_id	route_id	depart	seat_num	class_id	travel_date	flight_num	arrival
1	ERJ142	9	DEN	01EP	Economy Plus	2019-12-26	1119	LAX
2	767-301ER	4	JFK	01E	Economy	2018-09-02	1114	LAX
4	767-301ER	5	LAX	02FC	First Class	2020-04-06	1115	JFX
4	767-301ER	4	JFK	03FC	First Class	2020-04-30	1114	LAX
5	767-301ER	12	ABI	02B	Business	2018-07-02	1122	ADK
5	ERJ142	18	ANI	02E	Economy	2020-05-06	1128	BGR
5	ERJ142	22	BGR	03E	Economy	2020-05-31	1132	BJI
7	767-301ER	20	AVL	03B	Business	2020-07-08	1130	BOI
9	767-301ER	15	CAK	04FC	First Class	2020-09-10	1125	ANI
10	A321	10	HNL	05E	Economy	2020-10-11	1120	DEN
11	767-301ER	4	JFK	05B	Business	2020-11-09	1114	LAX
11	767-301ER	5	LAX	04B	Business	2020-11-12	1115	JFX
13	A321	13	ADK	06FC	First Class	2019-01-05	1123	BQN
15	A321	14	BQN	06B	Business	2018-11-02	1124	CAK
17	A321	13	ABI	04EP	Economy Plus	2019-06-03	1123	ADK
18	767-301ER	1	EWR	13FC	First Class	2018-04-01	1111	HNL
22	ERJ142	22	BGR	07EP	Economy Plus	2020-02-09	1132	BJI
24	A321	14	BQN	08B	Business	2019-07-22	1124	CAK
25	767-301ER	23	BLV	09B	Business	2019-03-07	1133	BFL
29	ERJ142	9	DEN	11B	Business	2018-05-03	1119	LAX
31	767-301ER	20	AVL	13E	Economy	2018-12-31	1130	BOI
44	767-301ER	15	CAK	11FC	First Class	2020-10-06	1125	ANI
46	A321	8	ORD	12FC	First Class	2011-07-08	1118	EWR
46	A321	25	RDM	14E	Economy	2020-11-25	1135	BJI
49	767-301ER	15	CAK	13B	Business	2020-08-19	1125	ANI
50	A321	21	BFL	10EP	Economy Plus	2020-08-15	1131	BET
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

63 22:06:59 Select * from passengers_on_flights where route_id between 01 and 25 LIMIT 0, 1000 26 row(s) returned 0.016 sec / 0.000 sec

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

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

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

total_passengers	total_revenue
13	6034

70 22:32:10 SELECT SUM(no_of_tickets) AS total_passengers, SUM(price_per_ticket * no_of_tickets) AS total_revenue F... 1 row(s) returned 0.016 sec / 0.000 sec

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

```
Select customer_id, concat(first_name,' ',last_name) AS Full_Name
from customer;
```

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	Pheny Eri
23	Erwin Tosh
24	Calvin Willis
25	Moss Morris
26	Bryan Collin
27	Cherly Vernon
28	Du plesis Chris
29	Watson Ronald
30	Donack Dukins
31	James Robert
32	Chirstoper Sean
33	Mark Ethan
34	Jacqueline Keith
35	Jeffrey Aaron
36	Kayla Patrick
37	Samuel Scott
38	Alexis Scott
39	Tyler Edward
40	Adam Paul
41	Kyle Mark
42	Roger Mattew
43	Joe Daniel
44	Bily Brian
45	Doris Walter
46	Louis Douglas
47	Sophia Carl
48	Wayne Noah
49	Russell Peter
50	Rose Arthur

72 23:32:40 Select customer_id, concat(first_name,' ',last_name) AS Full_Name from customer LIMIT 0, 1000 50 row(s) returned 0.000 sec / 0.000 sec

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

```
Select c.customer_id, c.first_name, c.last_name
from customer c
JOIN ticket_details t ON c.customer_id = t.customer_id;
```

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

customer_id	first_name	last_name
1	Julie	Sam
1	Julie	Sam
2	Steve	Ryan
2	Steve	Ryan
4	Cathenna	Emily
4	Cathenna	Emily
5	Aaron	Kim
5	Aaron	Kim
5	Aaron	Kim
7	Anderson	Stewart
8	Floyd	Ted
8	Floyd	Ted
9	Leo	Travis
9	Leo	Travis
10	Melvin	Tracy
11	Roger	Walson
11	Roger	Walson
11	Roger	Walson
13	Solomon	Walter
14	Carol	Vernon
14	Carol	Vernon
15	Linda	William
16	Chirstine	Willis
17	Catherine	Shad
18	Gloria	Richie
18	Gloria	Richie
19	Joyce	Paul
19	Joyce	Paul
19	Joyce	Paul
20	Sara	Oliver
20	Sara	Oliver
21	Chirsty	Josh
22	Pheny	Eri
24	Calvin	Willis
25	Moss	Morris
25	Moss	Morris

27	Cherly	Vernon
28	Du plesis	Chris
29	Watson	Ronald
29	Watson	Ronald
31	James	Robert
32	Chirstoper	Sean
33	Mark	Ethan
41	Kyle	Mark
44	Bily	Brian
46	Louis	Douglas
46	Louis	Douglas
47	Sophia	Carl
49	Russell	Peter
50	Rose	Arthur

Output

Action Output

#	Time	Action	Message	Duration / Fetch
80	10:57:18	Select * from passengers_on_flights where first_name like 'A%' LIMIT 0, 1000	Error Code: 1054. Unknown column 'first_name' in 'where clause'	0.000 sec
81	10:57:31	Select * from passengers_on_flights where first_name like '%A' LIMIT 0, 1000	Error Code: 1054. Unknown column 'first_name' in 'where clause'	0.000 sec
82	10:57:51	Select * from customer where first_name like '%A' LIMIT 0, 1000	6 row(s) returned	0.015 sec / 0.000 sec
83	10:58:01	Select * from customer where first_name like 'A%' LIMIT 0, 1000	5 row(s) returned	0.000 sec / 0.000 sec
84	10:58:37	Select * from ticket_details where customer_id = 5 LIMIT 0, 1000	3 row(s) returned	0.000 sec / 0.000 sec
85	11:02:21	Select c.customer_id, c.first_name, c.last_name from customer c JOIN ticket_details t ON c.customer_id = t.cu...	50 row(s) returned	0.000 sec / 0.000 sec

Task 7th 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 c.first_name, c.last_name
from customer as c
JOIN ticket_details t
ON c.customer_id= t.customer_id
where brand = 'Emirates';
```

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

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

Output

Action Output

#	Time	Action	Message	Duration / Fetch
88	13:43:03	select customer_id, brand from ticket_details where brand = 'Emirates' LIMIT 0, 1000	18 row(s) returned	0.000 sec / 0.000 sec
89	13:44:03	Select * from customer LIMIT 0, 1000	50 row(s) returned	0.000 sec / 0.000 sec
90	13:46:27	select * from ticket_details LIMIT 0, 1000	50 row(s) returned	0.000 sec / 0.000 sec
91	13:46:52	Select c.first_name, c.last_name from customer as c JOIN (select t.customer_id, t.brand from ticket_details t w...	18 row(s) returned	0.000 sec / 0.000 sec
92	13:49:14	SELECT c.first_name, c.last_name FROM customer c JOIN ticket_details t ON c.customer_id = t.customer_id ...	18 row(s) returned	0.000 sec / 0.000 sec
93	13:52:27	Select c.first_name, c.last_name from customer as c JOIN ticket_details t ON c.customer_id= t.customer_id wh...	18 row(s) returned	0.000 sec / 0.000 sec

Task 8th Write a query to identify the customers who have traveled by Economy Plus class using the sub-query on the passengers_on_flights table.

```
Select c.first_name, c.last_name
from customer as c
where c.customer_id IN
                        (Select customer_id
                        from passengers_on_flights
                        where class_id = 'Economy Plus');
```

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

first_name	last_name
Julie	Sam
Floyd	Ted
Roger	Walson
Catherine	Shad
Joyce	Paul
Pheny	Eri
Chirstoper	Sean
Sophia	Carl
Rose	Arthur

customer 54 × Read Only

Output

Action Output

#	Time	Action	Message	Duration / Fetch
99	14:03:14	Select c.first_name, c.last_name from customer as c JOIN (Select class_id from passengers_on_flights as p) p...	Error Code: 1054. Unknown column 'p.customer_id' in 'on clause'	0.000 sec
100	14:03:38	Select c.first_name, c.last_name from customer as c JOIN (Select class_id from passengers_on_flights as p ...	Error Code: 1054. Unknown column 'p.customer_id' in 'on clause'	0.000 sec
101	14:05:12	Select c.first_name, c.last_name from customer as c JOIN (Select class_id from passengers_on_flights as p ...	Error Code: 1054. Unknown column 'p.customer_id' in 'on clause'	0.000 sec
102	14:05:55	Select c.first_name, c.last_name from customer as c JOIN passengers_on_flights as p on c.customer_id = ...	10 row(s) returned	0.000 sec / 0.000 sec
103	15:59:17	Select c.first_name, c.last_name from customer as c where c.customer_id IN (Select customer_id ...	9 row(s) returned	0.000 sec / 0.000 sec
104	15:59:36	SELECT c.first_name, c.last_name FROM customer c WHERE c.customer_id IN (SELECT customer_id FR...	9 row(s) returned	0.000 sec / 0.000 sec

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

```
Select IF(
            sum(price_per_ticket * no_of_tickets) > 10000,
            'Revenue Crossed over 10000',
            'Revenue Not Croeesed over 10000'
        ) AS revenue_status
from ticket_details;
```

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

revenue_status
Revenue Crossed over 10000

107	16:10:03	Select IF(sum(price_per_ticket * no_of_tickets) > 10000,	'Revenue Crossed over 10000',	'Rev...	1 row(s) returned	0.000 sec / 0.000 sec

Task 10th Write a query to create and grant access to a new user to perform database operations

```
CREATE USER 'aircargo_user'@'localhost' IDENTIFIED BY 'Password123!';

GRANT ALL PRIVILEGES ON AirCargo.* TO 'aircargo_user'@'localhost';

FLUSH PRIVILEGES;
```

Output

Action Output

#	Time	Action	Message	Duration / Fetch
105	16:05:01	select * from ticket_details LIMIT 0, 1000	50 row(s) returned	0.000 sec / 0.000 sec
106	16:05:36	select sum(price_per_ticket) from ticket_details LIMIT 0, 1000	1 row(s) returned	0.000 sec / 0.000 sec
107	16:10:03	Select IF(sum(price_per_ticket * no_of_tickets) > 10000, 'Revenue Crossed over 10000', 'Rev...	1 row(s) returned	0.000 sec / 0.000 sec
108	16:18:52	CREATE USER 'aircargo_user'@'localhost' IDENTIFIED BY 'Password123!'	0 row(s) affected	0.109 sec
109	16:19:51	GRANT ALL PRIVILEGES ON AirCargo.* TO 'aircargo_user'@'localhost'	0 row(s) affected	0.000 sec
110	16:20:38	FLUSH PRIVILEGES	0 row(s) affected	0.015 sec

Welcome to MySQL Workbenc

Setup New Connection

Connection Name: AirCargo User — Type a name for the connection

Connection Method: Standard (TCP/IP) — Method to use to connect to the RDBMS

Parameters | SSL | Advanced

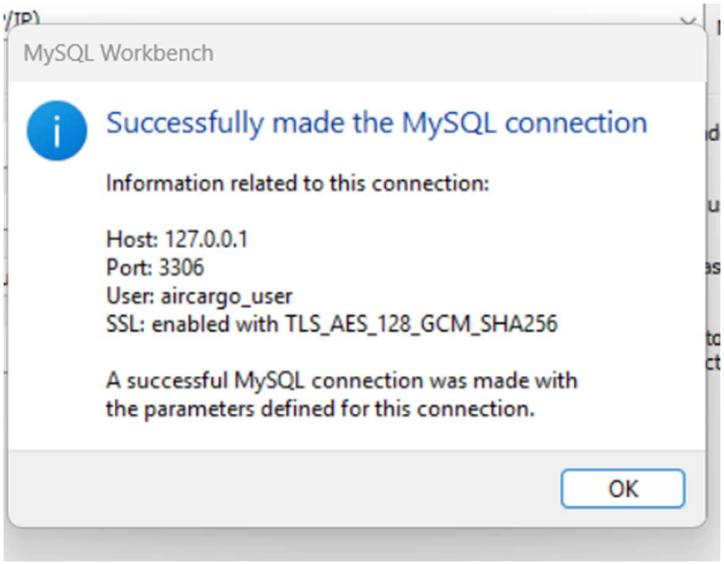
Hostname: 127.0.0.1 Port: 3306 — Name or IP address of the server host - and TCP/IP port.

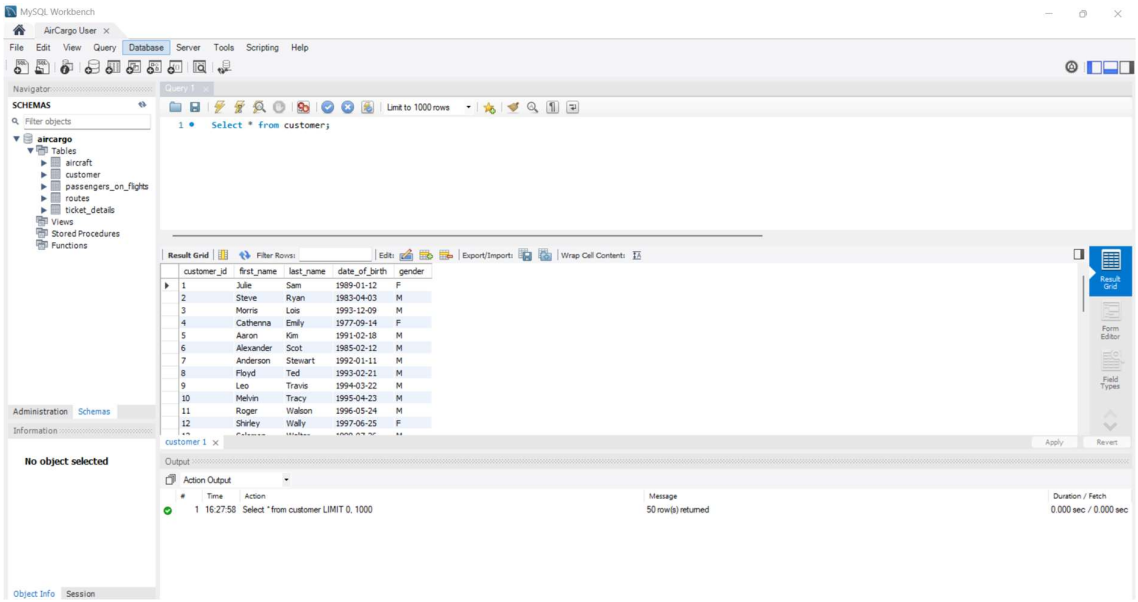
Username: aircargo_user — Name of the user to connect with.

Password: Store in Vault ... Clear — The user's password. Will be requested later if it's not set.

Default Schema: AirCargo — The schema to use as default schema. Leave blank to select it later.

Configure Server Management... Test Connection Cancel OK





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

```
SELECT
  class_id,
  price_per_ticket,
  MAX(price_per_ticket) OVER (PARTITION BY class_id) AS max_price_per_class
FROM ticket_details;
```

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

class_id	price_per_ticket	max_price_per_class
Business	499	510
Business	430	510
Business	490	510
Business	490	510
Business	510	510
Business	430	510
Business	480	510
Business	430	510
Business	505	510
Business	465	510
Business	410	510
Business	430	510
Business	465	510
Economy	130	190
Economy	130	190
Economy	170	190
Economy	120	190
Economy	120	190
Economy	130	190
Economy	135	190
Economy	120	190
Economy	135	190
Economy	190	190
Economy	150	190
Economy	170	190
Economy	100	190
Economy	190	190
Econom...	220	295
Econom...	225	295
Econom...	220	295
Econom...	250	295
Econom...	250	295
Econom...	275	295
Econom...	295	295
Econom...	275	295
Econom...	225	295
Econom...	225	295
First Class	390	395
First Class	380	395
First Class	395	395
First Class	390	395
First Class	320	395
First Class	365	395
First Class	375	395
First Class	315	395
First Class	390	395
First Class	395	395
First Class	380	395
First Class	395	395
First Class	395	395

10	17:57:49	Select class_id, price_per_ticket, MAX(price_per_ticket) over (partition by class_id) as max_price_per_class fr...	50 row(s) returned	0.000 sec / 0.000 sec
11	17:58:15	SELECT class_id, price_per_ticket, MAX(price_per_ticket) OVER (PARTITION BY class_id) AS max_price...	50 row(s) returned	0.015 sec / 0.000 sec

Task 12th Write a query to extract the passengers whose route ID is 4 by improving the speed and performance of the passengers_on_flights table using the index.

```
CREATE INDEX idx_route_id ON passengers_on_flights(route_id);

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

Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell Content:

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

Task 13th For 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;
```

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

id	select_type	table	partitions	type	possible_keys	key	key_len	ref	rows	filtered	Extra
1	SIMPLE	passengers_on_flights	NULL	ref	idx_route_id	idx_route_id	5	const	3	100.00	NULL

Task 14th Write a query to calculate the total price of all tickets booked by a customer across different aircraft IDs using the rollup function.

```
Select customer_id, aircraft_id, SUM(price_per_ticket * no_of_tickets) AS total_price
from ticket_details
group by customer_id, aircraft_id with rollup;
```

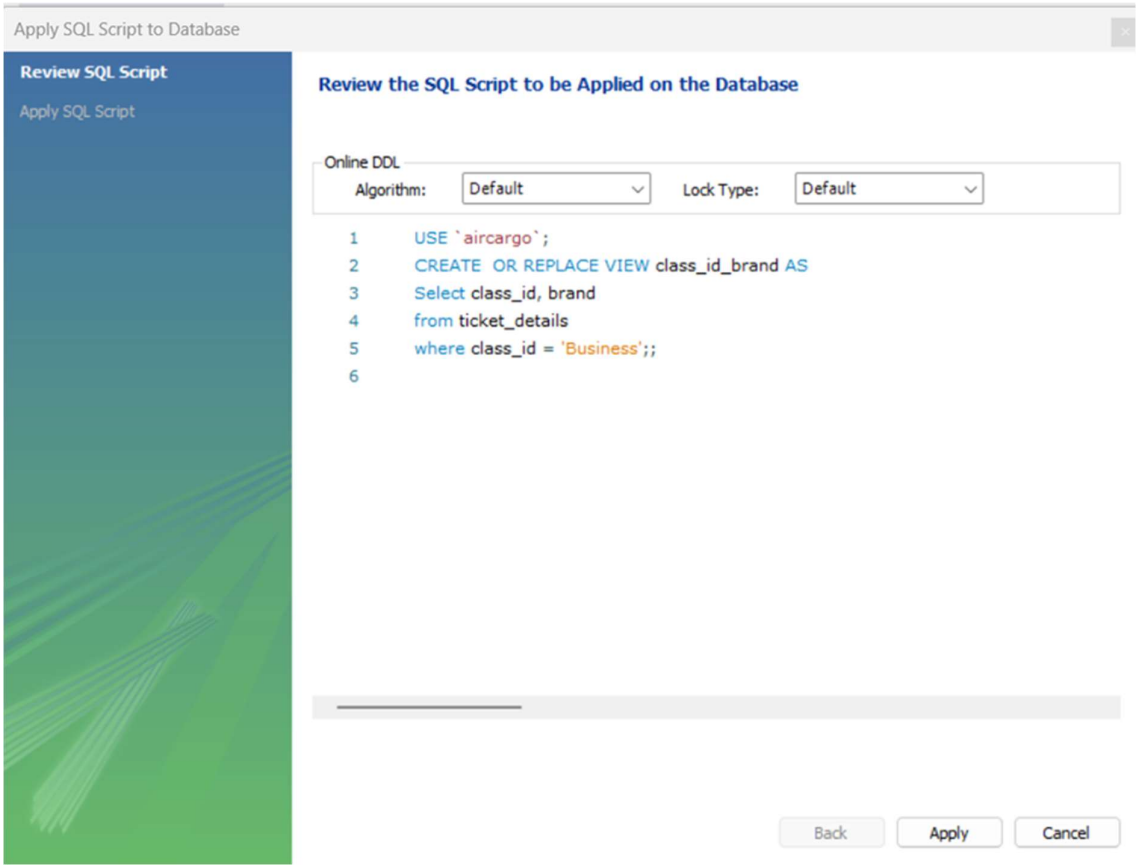
Result Grid | Filter Rows: | Export: | Wrap Cell Content:

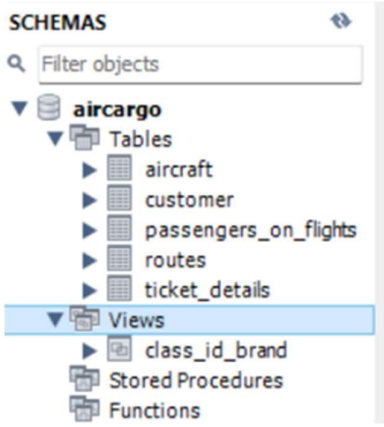
customer_id	aircraft_id	total_price
1	CRJ900	320
1	ERJ142	250
1	NULL	570
2	767-301ER	130
2	A321	505
2	NULL	635
4	767-301ER	780
4	NULL	780
5	767-301ER	430
5	ERJ142	240
5	NULL	670
7	767-301ER	430
7	NULL	430
8	A321	465
8	NULL	465
9	767-301ER	380
9	CRJ900	390
9	NULL	770
10	A321	135
10	NULL	135
11	767-301ER	930
11	ERJ142	295
11	NULL	1225
13	A321	395
13	NULL	395
14	767-301ER	170
14	ERJ142	120
14	NULL	290
15	A321	430
15	NULL	430
16	CRJ900	395
16	NULL	395
17	A321	250
17	NULL	250
18	767-301ER	565
18	NULL	565
19	767-301ER	100
19	CRJ900	450
19	NULL	550
20	CRJ900	680
20	NULL	680
21	CRJ900	490
21	NULL	490
22	ERJ142	220

22	NULL	220
24	A321	480
24	NULL	480
25	767-301ER	649
25	NULL	649
27	767-301ER	130
27	NULL	130
28	ERJ142	170
28	NULL	170
29	A321	410
29	ERJ142	510
29	NULL	920
31	767-301ER	130
31	NULL	130
32	ERJ142	220
32	NULL	220
33	CRJ900	490
33	NULL	490
41	A321	395
41	NULL	395
44	767-301ER	380
44	NULL	380
46	A321	530
46	NULL	530
47	CRJ900	225
47	NULL	225
49	767-301ER	430
49	NULL	430
50	A321	275
50	NULL	275
NULL	NULL	15369

42 19:14:32 Select customer_id, aircraft_id, SUM(price_per_ticket * no_of_tickets) AS total_price from ticket_details group ... 75 row(s) returned 0.000 sec / 0.000 sec

Task 15th Write a query to create a view with only business class customers and the airline brand.






```
Select * from class_id_brand;
```

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

class_id	brand
Business	Bristish Airways
Business	Emirates
Business	Emirates
Business	Emirates
Business	Qatar Airways
Business	Qatar Airways
Business	Qatar Airways
Business	Jet Airways
Business	Emirates
Business	Qatar Airways
Business	Bristish Airways
Business	Emirates
Business	Emirates

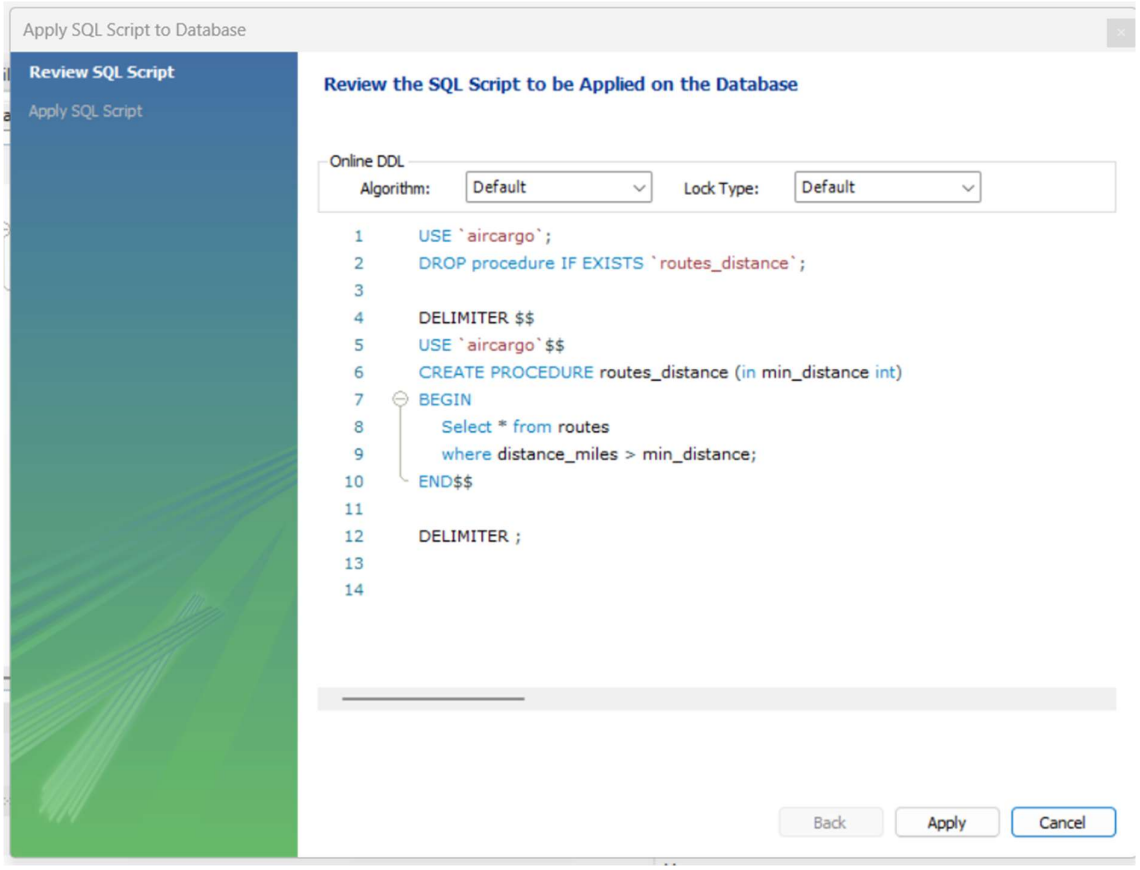
Output

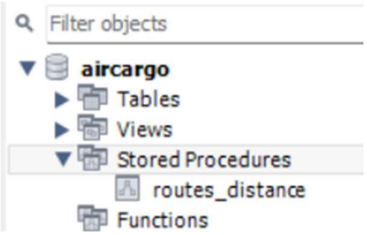
Action Output

#	Time	Action	Message	Duration / Fetch
43	20:36:12	Select * from ticket_details LIMIT 0, 1000	50 row(s) returned	0.000 sec / 0.000 sec
44	17:44:46	Select * from ticket_details LIMIT 0, 1000	50 row(s) returned	0.016 sec / 0.000 sec
45	17:45:11	Select class_id, brand from ticket_details LIMIT 0, 1000	50 row(s) returned	0.000 sec / 0.000 sec
46	17:47:24	Select class_id, brand from ticket_details where class_id = 'Business' LIMIT 0, 1000	13 row(s) returned	0.000 sec / 0.000 sec
47	17:48:09	Apply changes to class_id_brand	Changes applied	
48	17:50:25	Select * from class_id_brand LIMIT 0, 1000	13 row(s) returned	0.000 sec / 0.000 sec

Task 16th 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.

```
CREATE PROCEDURE routes_distance (in min_distance int)
BEGIN
    Select * from routes
    where distance_miles > min_distance;
END
```






```
call routes_distance(2000);
```

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

route_id	flight_num	origin_airport	destination_airport	aircraft_id	distance_miles
1	1111	EWR	HNL	767-301ER	4962
2	1112	HNL	EWR	767-301ER	4962
3	1113	EWR	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

Result 4 | Read Only

Output

Action Output

#	Time	Action	Message	Duration / Fetch
2	11:22:42	Select * from routes where distance_miles > 2000 LIMIT 0, 1000	24 row(s) returned	0.016 sec / 0.000 sec
3	11:51:55	Apply changes to routes_distance	Changes applied	
4	11:53:34	call routes_distance(2000)	24 row(s) returned	0.000 sec / 0.000 sec
5	11:53:50	call routes_distance(3000)	11 row(s) returned	0.000 sec / 0.000 sec
6	11:53:59	call routes_distance(4000)	8 row(s) returned	0.016 sec / 0.000 sec
7	11:54:09	call routes_distance(2000)	24 row(s) returned	0.016 sec / 0.000 sec

Task 17th Using GROUP BY, determine the total number of tickets purchased by each customer and the total price paid.

```
Select c.customer_id, concat(c.first_name, ' ', last_name) as Customer_Name,
sum(t.no_of_tickets) as total_tickets,
sum(t.no_of_tickets * t.price_per_ticket) as total_price
from ticket_details as t
JOIN customer as c
on t.customer_id = c.customer_id
group by
c.customer_id, concat(c.first_name, ' ', last_name);
```

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

customer_id	Customer_Name	total_tickets	total_price
27	Cherly Vernon	1	130
22	Pheny Eri	1	220
21	Chirsty Josh	1	490
4	Cathenna Emily	2	780
5	Aaron Kim	3	670
7	Anderson Stewart	1	430
8	Floyd Ted	2	465
9	Leo Travis	2	770
10	Melvin Tracy	1	135
11	Roger Walson	3	1225
19	Joyce Paul	3	550
13	Solomon Walter	1	395
14	Carol Vernon	2	290
25	Moss Morris	2	649
16	Chirstine Willis	1	395
17	Catherine Shad	1	250
18	Gloria Richie	2	565
24	Calvin Willis	1	480
20	Sara Oliver	2	680
29	Watson Ronald	2	920
1	Julie Sam	2	570
2	Steve Ryan	2	635
15	Linda William	1	430
28	Du plesis Chris	1	170
31	James Robert	1	130
32	Chirstoper Sean	1	220
33	Mark Ethan	1	490
49	Russell Peter	1	430
50	Rose Arthur	1	275
44	Bily Brian	1	380
46	Louis Douglas	2	530
47	Sophia Carl	1	225
41	Kyle Mark	1	395

Result 17 | Read Only

Output

Action Output

#	Time	Action	Message	Duration / Fetch
20	12:14:28	Select customer_id, concat(first_name, ' ', last_name) as Customer_Name from customer LIMIT 0, 1000	50 row(s) returned	0.015 sec / 0.000 sec
21	12:18:28	Select c.customer_id, c.concat(first_name, ' ', last_name) as Customer_Name, sum(t.no_of_tickets) as total_tic...	Error Code: 1305. FUNCTION c.concat does not exist	0.000 sec
22	12:19:37	Select c.customer_id, concat(c.first_name, ' ', last_name) as Customer_Name, sum(t.no_of_tickets) as total_tic...	33 row(s) returned	0.000 sec / 0.000 sec
23	12:20:22	Select c.customer_id, concat(c.first_name, ' ', last_name) as Customer_Name, sum(t.no_of_tickets) as total_tic...	33 row(s) returned	0.000 sec / 0.000 sec
24	12:20:46	Select c.customer_id, concat(c.first_name, ' ', last_name) as Customer_Name, sum(t.no_of_tickets) as total_tic...	33 row(s) returned	0.000 sec / 0.000 sec
25	12:21:23	Select c.customer_id, concat(c.first_name, ' ', last_name) as Customer_Name, sum(t.no_of_tickets) as total_tic...	33 row(s) returned	0.000 sec / 0.000 sec

Task 18th Calculate the average number of passengers per flight route.

```
select route_id, COUNT(*) / COUNT(DISTINCT travel_date) AS avg_passengers_per_day
FROM passengers_on_flights
GROUP BY route_id;
```

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

route_id	avg_passengers_per_day
1	1.0000
4	1.0000
5	1.0000
8	1.0000
9	1.0000
10	1.0000
12	1.0000
13	1.0000
14	1.0000
15	1.0000
18	1.0000
20	1.0000
21	1.0000
22	1.0000
23	1.0000
25	1.0000
26	1.0000
30	1.0000
31	1.0000
32	1.0000
33	1.0000
34	1.0000
35	1.0000
36	1.0000
38	1.0000
39	1.0000
42	1.0000
43	1.0000
44	1.0000
45	1.0000
46	1.0000
47	1.0000

Result 26 | Read Only

Output

Action Output

#	Time	Action	Message	Duration / Fetch
29	12:30:57	Select * from passengers_on_flights LIMIT 0, 1000	50 row(s) returned	0.000 sec / 0.000 sec
30	12:32:49	SELECT route_id, COUNT(*) / COUNT(DISTINCT travel_date) AS avg_passengers_per_day FROM passe...	32 row(s) returned	0.000 sec / 0.000 sec
31	12:33:19	select route_id, COUNT(*) / COUNT(DISTINCT travel_date) AS avg_passengers_per_day FROM passengers...	32 row(s) returned	0.000 sec / 0.000 sec
32	12:33:41	SELECT SUM(passenger_count) / COUNT(*) AS overall_avg_passengers_per_route FROM (SELECT rout...	1 row(s) returned	0.000 sec / 0.000 sec
33	12:35:54	select route_id, COUNT(*) / COUNT(DISTINCT travel_date) AS avg_passengers_per_day FROM passengers...	32 row(s) returned	0.000 sec / 0.000 sec
34	12:37:25	select route_id, COUNT(*) / COUNT(DISTINCT travel_date) AS avg_passengers_per_day FROM passengers...	32 row(s) returned	0.000 sec / 0.000 sec