Air Cargo Analysis

MADE BY- ASHISH CHAMEL

COURSE- Data Acquisition and Manipulation using SQL

DATE OF SUBMISSON- 08-03-2025

Pre-requisites

-Changing the date format to – "YYYY-MM-DD" IN THE EXCEL

- Here what I did is changed Column D in text format then I extracted the text using =TEXT(D2,"yyyy-mm-dd")
- Now copy the values from E column and paste special select values into D

	A	B	C	D	E	F
1	customer_id	first_name	last_name	date_of_birth		gender
2	1	Julie	Sam	32520	=TEXT(D2,"yyyy-mm-dd")	
3	2	Steve	Ryan	30409	1983-04-03	M
4	3	Morris	Lois	34312	1993-12-09	M
5	4	Cathenna	Emily	28382	1977-09-14	F
6	5	Aaron	Kim	33287	1991-02-18	M
7	6	Alexander	Scot	31090	1985-02-12	M
8	7	Anderson	Stewart	33614	1992-01-11	M

- Delete the column E

	A	B	C	D	E
1	customer_id	first_name	last_name	date_of_birth	gender
2	1	Julie	Sam	1989-01-12	F
3	2	Steve	Ryan	1983-04-03	M
4	3	Morris	Lois	1993-12-09	M
5	4	Cathenna	Emily	1977-09-14	F
6	5	Aaron	Kim	1991-02-18	M
7	6	Alexander	Scot	1985-02-12	M
8	7	Anderson	Stewart	1992-01-11	M

- Do the similar for other files excel file

Action 1

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;


CREATE TABLE customer (

   customer_id INT PRIMARY KEY,

   first_name VARCHAR(50),

   last_name VARCHAR(50),

   date_of_birth DATE,

   gender CHAR(1)

);


LOAD DATA INFILE 'C:/ProgramData/MySQL/MySQL Server 8.0/Uploads/customer.csv'

INTO TABLE customer

FIELDS TERMINATED BY ','

LINES TERMINATED BY '\n'

IGNORE 1 ROWS;
```

Result Grid | Filter Rows: | Edit:

customer_id	first_name	last_name	date_of_birth	gender
1	Julie	Sam	1989-01-12	F
2	Steve	Ryan	1983-04-03	M
3	Morris	Lois	1993-12-09	M
4	Cathenna	Emily	1977-09-14	F
5	Aaron	Kim	1991-02-18	M
6	Alexander	Scot	1985-02-12	M
7	Anderson	Stewart	1992-01-11	M
8	Floyd	Ted	1993-02-21	M
9	Leo	Travis	1994-03-22	M
10	Melvin	Tracy	1995-04-23	M
11	Roger	Walson	1996-05-24	M
12	Shirley	Wally	1997-06-25	F
13	Solomon	Walter	1998-07-26	M
14	Carol	Vernon	1999-08-27	F
15	Linda	William	1986-09-28	F
16	Chirstine	Willis	1987-10-06	F
17	Catherine	Shad	1988-11-09	F
18	Gloria	Richie	1989-12-04	F
19	Joyce	Paul	1990-06-02	F
20	Sara	Oliver	1991-01-01	F
21	Chirsty	Josh	2004-01-10	M
22	Pheny	Eri	1999-01-29	M
23	Erwin	Tosh	1994-02-03	M
24	Calvin	Willis	1994-02-15	M
25	Moss	Morris	2011-02-18	M
26	Bryan	Collin	2011-02-28	M

```
CREATE TABLE routes (
  route_id INT PRIMARY KEY,
  flight_num INT,
  origin_airport VARCHAR(10),
  destination_airport VARCHAR(10),
  aircraft_id VARCHAR(20),
  distance_miles INT
);

LOAD DATA INFILE 'C:/ProgramData/MySQL/MySQL Server 8.0/Uploads/routes.csv'
INTO TABLE routes
FIELDS TERMINATED BY ','
LINES TERMINATED BY '\n'
IGNORE 1 ROWS;
```

Result Grid | Filter Rows: | Edit: | Export/Import:

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
7	1117	LAX	ORD	A321	1745
8	1118	ORD	EWR	A321	719
9	1119	DEN	LAX	ERJ142	862
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
15	1125	CAK	ANI	767-301ER	2000
16	1126	ALB	APN	A321	1700
17	1127	APN	BLV	767-301ER	1900
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
22	1132	BGR	BJI	ERJ142	1242
23	1133	BLV	BFL	767-301ER	2354
24	1134	BJI	BQN	A321	1575
25	1135	RDM	BJI	A321	2425
26	1136	BET	BTM	ERJ142	1311
27	1137	BOI	CLD	A321	578
28	1138	BOS	CDC	767-301ER	246
29	1139	BKG	CRW	767-301ER	909
30	1140	BUR	STT	CRJ900	780
31	1141	BTM	CHA	ERJ142	660

routes 8 ×

Output

```
CREATE TABLE passengers_on_flights (
   customer_id INT,
   aircraft_id VARCHAR(20),
   route_id INT,
   depart VARCHAR(10),
   arrival VARCHAR(10),
   seat_num VARCHAR(10),
   class_id VARCHAR(20),
   travel_date DATE,
   flight_num INT,
   PRIMARY KEY (customer_id, flight_num),
   FOREIGN KEY (customer_id) REFERENCES customer(customer_id),
   FOREIGN KEY (route_id) REFERENCES routes(route_id)
);

LOAD DATA INFILE 'C:/ProgramData/MySQL/MySQL Server 8.0/Uploads/passengers_on_flights.csv'
INTO TABLE passengers_on_flights
FIELDS TERMINATED BY ','
LINES TERMINATED BY '\n'
```

IGNORE 1 ROWS;

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

customer_id	aircraft_id	route_id	depart	arrival	seat_num	class_id	travel_date	flight_num
1	ERJ142	9	DEN	LAX	01EP	Economy Plus	2019-12-26	1119
1	CRJ900	30	BUR	STT	01FC	First Class	2018-11-04	1140
2	767-301ER	4	JFK	LAX	01E	Economy	2018-09-02	1114
2	A321	34	CRW	COD	01B	Business	2019-01-26	1117
4	767-301ER	4	JFK	LAX	03FC	First Class	2020-04-30	1114
4	767-301ER	5	LAX	JFX	02FC	First Class	2020-04-06	1115
5	767-301ER	12	ABI	ADK	02B	Business	2018-07-02	1122
5	ERJ142	18	ANI	BGR	02E	Economy	2020-05-06	1128
5	ERJ142	22	BGR	BJI	03E	Economy	2020-05-31	1132
7	767-301ER	20	AVL	BOI	03B	Business	2020-07-08	1130
8	A321	38	CST	DAL	02EP	Economy Plus	2020-08-09	1148
8	A321	43	CBM	BOI	04E	Economy	2018-05-02	1153
9	767-301ER	15	CAK	ANI	04FC	First Class	2020-09-10	1125
9	CRJ900	33	CDC	CST	05FC	First Class	2018-02-01	1143
10	A321	10	HNL	DEN	05E	Economy	2020-10-11	1120
11	767-301ER	4	JFK	LAX	05B	Business	2020-11-09	1114
11	767-301ER	5	LAX	JFX	04B	Business	2020-11-12	1115
11	ERJ142	31	BTM	CHA	03EP	Economy Plus	2018-08-02	1141
13	A321	13	ADK	BQN	06FC	First Class	2019-01-05	1123
14	ERJ142	35	STT	CDB	06E	Economy	2019-04-02	1145
14	767-301ER	42	CSG	BOS	07E	Economy	2020-01-25	1152
15	A321	14	BQN	CAK	06B	Business	2018-11-02	1124
16	CRJ900	39	COD	SCC	07FC	First Class	2019-05-04	1149
17	A321	13	ABI	ADK	04EP	Economy Plus	2019-06-03	1123
18	767-301ER	1	EWR	HNL	13FC	First Class	2018-04-01	1111
18	767-301ER	46	CDV	HNL	08E	Economy	2019-07-07	1156
19	CRJ900	30	BUR	STT	06EP	Economy Plus	2020-12-17	1140
19	767-301ER	32	CLD	CHI	09E	Economy	2018-02-07	1142
19	CRJ900	47	DAL	LAX	05EP	Economy Plus	2021-01-13	1157
20	CRJ900	36	CHA	COU	08FC	First Class	2019-08-26	1146

passengers_on_flights 9

Output

```
CREATE TABLE ticket_details (
   p_date DATE,
   customer_id INT,
   aircraft_id VARCHAR(20),
   class_id VARCHAR(20),
   no_of_tickets INT,
   a_code VARCHAR(10),
   price_per_ticket DECIMAL(10,2),
   brand VARCHAR(50),
   FOREIGN KEY (customer_id) REFERENCES customer(customer_id)
);
LOAD DATA INFILE 'C:/ProgramData/MySQL/MySQL Server 8.0/Uploads/ticket_details.csv'
INTO TABLE ticket_details
FIELDS TERMINATED BY ','
LINES TERMINATED BY '\n'
IGNORE 1 ROWS;

ALTER TABLE ticket_details
CHANGE COLUMN a_code airport_code VARCHAR(3); -- Assuming a_code is VARCHAR(3)
```

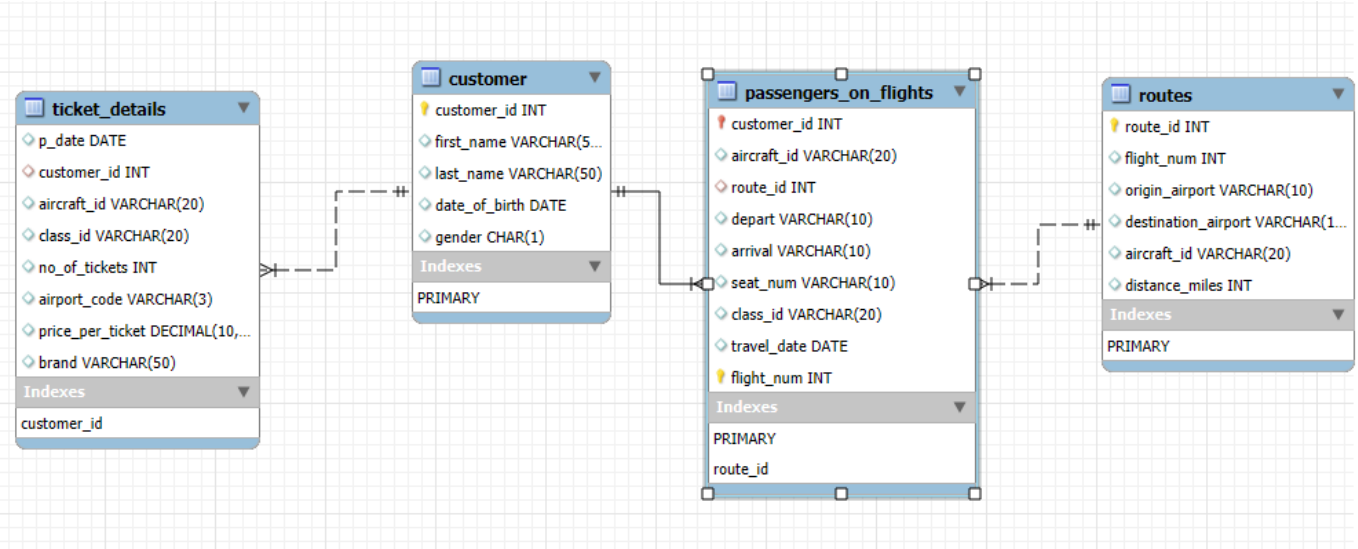
Result Grid | Filter Rows: | Export: | Wrap Cell Content:

p_date	customer_id	aircraft_id	class_id	no_of_tickets	airport_code	price_per_ticket	brand
2018-12-26	27	767-301ER	Economy	1	DAL	130.00	Emirates
2020-02-02	22	ERJ142	Economy Plus	1	AGB	220.00	Jet Airways
2020-03-03	21	CRJ900	Business	1	BOH	490.00	Bristish Airways
2020-04-04	4	767-301ER	First Class	1	AGB	390.00	Emirates
2020-05-05	5	ERJ142	Economy	1	CTM	120.00	Jet Airways
2020-07-07	7	767-301ER	Business	1	BFS	430.00	Emirates
2020-08-08	8	A321	Economy Plus	1	DAL	275.00	Qatar Airways
2020-09-09	9	767-301ER	First Class	1	BOH	380.00	Emirates
2020-10-10	10	A321	Economy	1	MCO	135.00	Qatar Airways
2020-11-11	11	767-301ER	Business	1	AGB	465.00	Emirates
2020-12-12	19	CRJ900	Economy Plus	1	DEN	225.00	Bristish Airways
2019-01-01	13	A321	First Class	1	YVR	395.00	Qatar Airways
2019-02-02	14	ERJ142	Economy	1	CTM	120.00	Jet Airways
2019-03-03	25	767-301ER	Business	1	BHX	499.00	Emirates
2019-04-04	16	CRJ900	First Class	1	YVR	395.00	Bristish Airways
2019-05-03	17	A321	Economy Plus	1	BFS	250.00	Qatar Airways
2019-06-06	18	767-301ER	Economy	1	YVR	190.00	Emirates
2019-07-07	24	A321	Business	1	CTM	480.00	Qatar Airways
2019-08-09	20	CRJ900	First Class	1	MCO	365.00	Bristish Airways
2019-09-21	25	767-301ER	Economy	1	BOH	150.00	Emirates
2019-10-22	29	A321	Business	1	PEK	410.00	Qatar Airways
2019-11-23	1	ERJ142	Economy Plus	1	BFS	250.00	Jet Airways
2019-12-24	14	767-301ER	Economy	1	BHX	170.00	Emirates
2019-01-25	2	A321	Business	1	YVR	505.00	Qatar Airways
2018-01-01	9	CRJ900	First Class	1	AGB	390.00	British Airways
2018-02-01	19	767-301ER	Economy	1	AGB	100.00	Emirates
2018-03-01	18	767-301ER	First Class	1	BFS	375.00	Emirates
2018-04-01	29	ERJ142	Business	1	EME	510.00	Jet Airways
2018-05-01	8	A321	Economy	1	YVR	190.00	Qatar Airways
2018-06-01	20	CRJ900	First Class	1	PEK	315.00	Bristish Airways

ticket_details 10

ACTION-2

Create an ER diagram for the given airlines' database.



ACTION-3

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 1 AND 25;

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

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

ACTION-4

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

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

Result Grid | Filter Rows:

num_passengers	total_revenue
13	6034.00

ACTION-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

ACTION-6

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

```
SELECT DISTINCT c.customer_id, c.first_name, c.last_name

FROM customer c

JOIN ticket_details t ON c.customer_id = t.customer_id;
```

customer_id	first_name	last_name
1	Julie	Sam
2	Steve	Ryan
4	Cathenna	Emily
5	Aaron	Kim
7	Anderson	Stewart
8	Floyd	Ted
9	Leo	Travis
10	Melvin	Tracy
11	Roger	Walson
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
24	Calvin	Willis
25	Moss	Morris
27	Cherly	Vernon
28	Du plesis	Chris
29	Watson	Ronald
31	James	Robert
32	Chirstoper	Sean
33	Mark	Ethan
41	Kyle	Mark
44	Bily	Brian
46	Louis	Douglas
47	Sophia	Carl

ACTION-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 c.first_name, c.last_name

FROM customer c

JOIN ticket_details t ON c.customer_id = t.customer_id

WHERE t.brand = 'Emirates';
```

Result Grid | Filter Rows:

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

ACTION-8

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.customer_id, c.first_name, c.last_name

FROM customer c

JOIN passengers_on_flights p

ON c.customer_id = p.customer_id

WHERE p.class_id = 'Economy Plus';
```

Result Grid | Filter Rows:

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

ACTION- 9

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 Exceeded', 'Below Threshold') AS revenue_status

FROM ticket_details;
```

Result Grid | Filter Rows:

revenue_status
Revenue Exceeded

ACTION-10

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

```
CREATE USER 'analyst'@'localhost' IDENTIFIED BY 'password';

GRANT SELECT, INSERT, UPDATE ON AirCargo.* TO 'analyst'@'localhost';
```

234	15:33:29	CREATE USER 'analyst'@'localhost' IDENTIFIED BY 'password'
235	15:33:29	GRANT SELECT, INSERT, UPDATE ON AirCargo.* TO 'analyst'@'localhost'

Verify user existence

SELECT User FROM mysql.user WHERE User = 'analyst' AND Host = 'localhost';

Result Grid

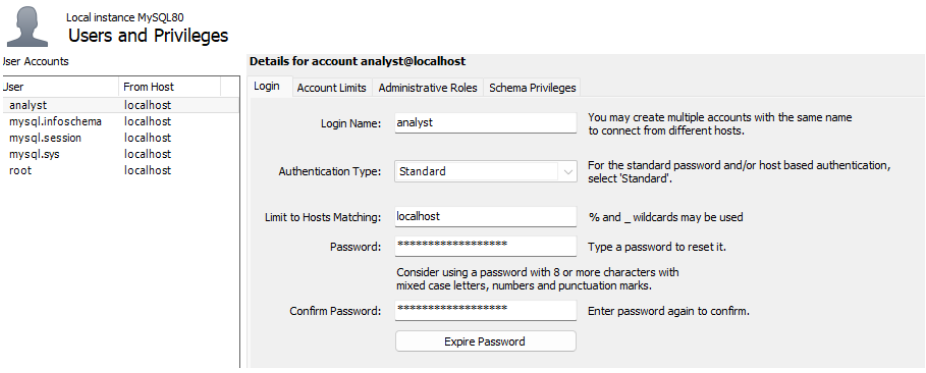
User
analyst

Verify user grants

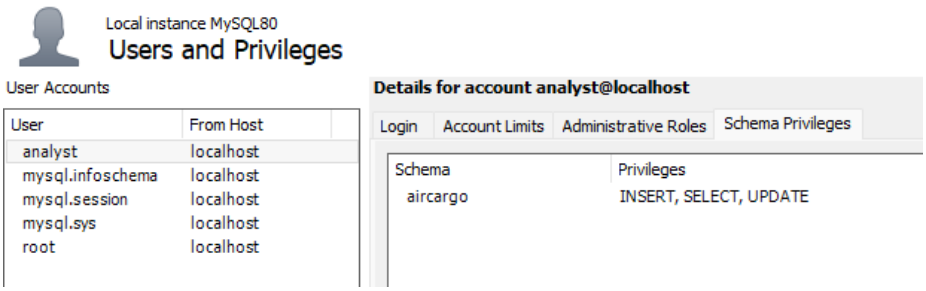
SHOW GRANTS FOR 'analyst'@'localhost';

Grants for analyst@localhost
GRANT USAGE ON *.* TO `analyst`@`localhost`
GRANT SELECT, INSERT, UPDATE ON `aircargo...

View User Details:



Verify Grants (Schema Privileges



ACTION-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) AS max_price

FROM ticket_details;

Result Grid | Filter Rows:

class_id	max_price
Business	510.00
Business	510.00
Business	510.00
Business	510.00
Business	510.00
Business	510.00
Business	510.00
Business	510.00
Business	510.00
Business	510.00
Business	510.00
Economy	190.00
Economy	190.00
Economy	190.00
Economy	190.00
Economy	190.00
Economy	190.00
Economy	190.00
Economy	190.00
Economy	190.00
Economy	190.00
Economy	190.00
Economy	190.00
Economy	190.00
Economy	190.00
Economy	190.00
Economy Plus	295.00
Economy Plus	295.00
Economy Plus	295.00
Economy Plus	295.00
Economy Plus	295.00

ACTION-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 using the index.

CREATE INDEX idx_route ON passengers_on_flights(route_id);

238	15:44:27	CREATE INDEX idx_route ON passengers_on_flights(route_id)

Verifying the Index's Existence-

SHOW INDEX FROM passengers_on_flights;

Table	Non_unique	Key_name	Seq_in_index	Column_name	Collation	Cardinality	Sub_part	Packed	Null	Index_type	Comment	Index_comment	Visible	Expression
passengers_on_flights	0	PRIMARY	1	customer_id	A	33	NULL	NULL		BTREE			YES	NULL
passengers_on_flights	0	PRIMARY	2	flight_num	A	50	NULL	NULL		BTREE			YES	NULL
passengers_on_flights	1	idx_route	1	route_id	A	32	NULL	NULL	YES	BTREE			YES	NULL

ACTION-13

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	idx_route	5	const	3	100.00	NULL

ACTION- 14.

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, SUM(price_per_ticket * no_of_tickets) AS total_spent

FROM ticket_details

GROUP BY customer_id WITH ROLLUP;

Result Grid | Filter Rows:

customer_id	total_spent
1	570.00
2	635.00
4	780.00
5	670.00
7	430.00
8	465.00
9	770.00
10	135.00
11	1225.00
13	395.00
14	290.00
15	430.00
16	395.00
17	250.00
18	565.00
19	550.00
20	680.00
21	490.00
22	220.00
24	480.00
25	649.00
27	130.00
28	170.00
29	920.00
31	130.00
32	220.00

ACTION-15

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

CREATE VIEW business_class_customers AS

SELECT customer_id, brand FROM ticket_details WHERE class_id = 'Business';

SELECT * FROM business_class_customers;

customer_id	brand
21	Bristish Airways
7	Emirates
11	Emirates
25	Emirates
24	Qatar Airways
29	Qatar Airways
2	Qatar Airways
29	Jet Airways
5	Emirates
15	Qatar Airways
33	Bristish Airways
49	Emirates
11	Emirates

ACTION-16

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 GetLongRoutes()

BEGIN

 SELECT * FROM routes WHERE distance_miles > 2000;

END //

DELIMITER ;

CALL GetLongRoutes();
```

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

ACTION-17

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

```
SELECT customer_id, SUM(no_of_tickets) AS total_tickets, SUM(price_per_ticket * no_of_tickets) AS total_price

FROM ticket_details

GROUP BY customer_id;
```

customer_id	total_tickets	total_price
1	2	570.00
2	2	635.00
4	2	780.00
5	3	670.00
7	1	430.00
8	2	465.00
9	2	770.00
10	1	135.00
11	3	1225.00
13	1	395.00
14	2	290.00
15	1	430.00
16	1	395.00
17	1	250.00
18	2	565.00
19	3	550.00
20	2	680.00
21	1	490.00
22	1	220.00
24	1	480.00
25	2	649.00
27	1	130.00
28	1	170.00
29	2	920.00
31	1	130.00
32	1	220.00

ACTION-18

Calculate the average distance and average number of passengers per aircraft, considering only those routes with more than one departure date.

```
SELECT r.aircraft_id, AVG(distance_miles) AS avg_distance, COUNT(DISTINCT customer_id) AS avg_passengers

FROM routes r

JOIN passengers_on_flights p ON r.route_id = p.route_id

GROUP BY r.aircraft_id

HAVING COUNT(DISTINCT travel_date) > 1

LIMIT 0, 1000;
```

Result Grid | Filter Rows: | Exp

aircraft_id	avg_distance	avg_passengers
767-301ER	3467.0556	14
A321	2646.3846	11
CRJ900	1107.8889	7
ERJ142	1353.1000	9