




Filter objects

Administration 

No object selected

```

1  -- Air Cargo Analysis Project
2
3  -- DESCRIPTION:
4  /* Air Cargo is an aviation company that provides air transportation services for passengers and freight. Air Cargo uses its aircraft to provide
5  different services with the help of partnerships or alliances with other airlines. The company wants to prepare reports on regular passengers, busiest routes,
6  ticket sales details, and other scenarios to improve the ease of travel and booking for customers.*/
7
8  -- Project Objective:
9  /* 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
10 aircraft required and prepare an analysis to determine the ticket sales details. This will ensure that the company improves its operability and becomes more
11 customer-centric and a favorable choice for air travel. */
12
13 -- Following operations should be performed:
14
15 -- Task 01: Create an ER diagram for the given airlines database.
16 • CREATE DATABASE air_cargo;
17 • USE air_cargo;
18
19 /* Task 02: Write a query to create route_details table using suitable data types for the fields, such as route_id, flight_num, origin_airport,
20 destination airport, aircraft id, and distance miles. Implement the check constraint for the flight number and unique constraint for the route id

```

Output

 Action Output

#	Time	Action	Message	Duration / Fetch
✓ 1	18:11:22	CREATE DATABASE air_cargo	1 row(s) affected	0.062 sec
✓ 2	18:11:29	USE air_cargo	0 row(s) affected	0.000 sec



SCHEMAS

Filter objects

air_cargo

Tables

Views

Stored Pr

Functions

demo

demo1

emp_data

employee

employee_d

employee_d

Administration

Information

Schema:

air_cargo

Schema:

air_cargo

Schema:

air_cargo

Schema:

air_cargo

Schema:

air_cargo

Schema:

air_cargo

Schema:

air_cargo

Schema:

air_cargo

Schema:

air_cargo

Schema:

air_cargo

Schema:

air_cargo



Limit to 2000 rows

```
11 customer-centric and a favorable choice for air travel. */
12
13 -- Following operations should be performed:
14
15 -- Task 01: Create an ER diagram for the given airlines database.
16 • CREATE DATABASE air_cargo;
17 • USE air_cargo;
18 • SELECT*FROM air_cargo.customer;
19 • SELECT*FROM air_cargo.passengers_on_flights;
20 • SELECT*FROM air_cargo.routes;
21 • SELECT*FROM air_cargo.ticket_details;
22
23
```

Result Grid Filter Rows: Export: Wrap Cell Content:

	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

customer 5 ×

Output

Action Output

#	Time	Action	Message	Duration / Fetch
✓ 34	18:31:07	DESCRIBE air_cargo.routes	6 row(s) returned	0.000 sec / 0.000 sec
✓ 35	18:31:24	DESCRIBE air_cargo.ticket_details	8 row(s) returned	0.000 sec / 0.000 sec
✓ 36	18:31:47	SELECT*FROM air_cargo.customer LIMIT 0, 2000	50 row(s) returned	0.000 sec / 0.000 sec



SCHEMAS

Filter objects

- air_cargo
 - Tables
 - Views
 - Stored Pr
 - Functions
 - demo
 - demo1
 - emp_data
 - employee
 - employee_d
 - employee_d
- Administration
- Information



Limit to 2000 rows

```
11 customer-centric and a favorable choice for air travel. */
12
13 -- Following operations should be performed:
14
15 -- Task 01: Create an ER diagram for the given airlines database.
16 • CREATE DATABASE air_cargo;
17 • USE air_cargo;
18 • SELECT*FROM air_cargo.customer;
19 • SELECT*FROM air_cargo.passengers_on_flights;
20 • SELECT*FROM air_cargo.routes;
21 • SELECT*FROM air_cargo.ticket_details;
22
23
```

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	A321	34	CRW	COD	01B	Bussiness	2019-01-26	1117
	2	767-301ER	4	JFK	LAX	01E	Economy	2018-09-02	1114
	1	ERJ142	9	DEN	LAX	01EP	Economy Plus	2019-12-26	1119
	1	CRJ900	30	BUR	STT	01FC	First Class	2018-11-04	1140
	5	767-301ER	12	ABI	ADK	02B	Bussiness	2018-07-02	1122

passengers_on_flights 6 ×

Read Only

Output

Action Output

#	Time	Action	Message	Duration / Fetch
✓ 35	18:31:24	DESCRIBE air_cargo.ticket_details	8 row(s) returned	0.000 sec / 0.000 sec
✓ 36	18:31:47	SELECT*FROM air_cargo.customer LIMIT 0, 2000	50 row(s) returned	0.000 sec / 0.000 sec
✓ 37	18:32:51	SELECT*FROM air_cargo.passengers_on_flights LIMIT 0, 2000	50 row(s) returned	0.000 sec / 0.000 sec



Navigator Air Cargo Analysis Project* ×

SCHEMAS

Filter objects

- air_cargo
 - Tables
 - Views
 - Stored Pr
 - Functions
 - demo
 - demo1
 - emp_data
 - employee
 - employee_d
 - employee_d
- Administration
- Information

Limit to 2000 rows

```
11 customer-centric and a favorable choice for air travel. */
12
13 -- Following operations should be performed:
14
15 -- Task 01: Create an ER diagram for the given airlines database.
16 • CREATE DATABASE air_cargo;
17 • USE air_cargo;
18 • SELECT*FROM air_cargo.customer;
19 • SELECT*FROM air_cargo.passengers_on_flights;
20 • SELECT*FROM air_cargo.routes;
21 • SELECT*FROM air_cargo.ticket_details;
22
23
```

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

routes 7 ×

Read Only

Output

Action Output

#	Time	Action	Message	Duration / Fetch
✓ 36	18:31:47	SELECT*FROM air_cargo.customer LIMIT 0, 2000	50 row(s) returned	0.000 sec / 0.000 sec
✓ 37	18:32:51	SELECT*FROM air_cargo.passengers_on_flights LIMIT 0, 2000	50 row(s) returned	0.000 sec / 0.000 sec
✓ 38	18:33:23	SELECT*FROM air_cargo.routes LIMIT 0, 2000	49 row(s) returned	0.000 sec / 0.000 sec



SCHEMAS

Filter objects

air_cargo

Tables

Views

Stored Pr

Functions

demo

demo1

emp_data

employee

employee_d

employee_d

Administration

Information

Schema:

air_cargo



Limit to 2000 rows

```
12
13  -- Following operations should be performed:
14
15  -- Task 01: Create an ER diagram for the given airlines database.
16 • CREATE DATABASE air_cargo;
17 • USE air_cargo;
18 • SELECT*FROM air_cargo.customer;
19 • SELECT*FROM air_cargo.passengers_on_flights;
20 • SELECT*FROM air_cargo.routes;
21 • SELECT*FROM air_cargo.ticket_details;
22
23
24
```

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

	p_date	customer_id	aircraft_id	class_id	no_of_tickets	a_code	Price_per_ticket	brand
▶	2018-12-26	27	767-301ER	Economy	1	DAL	130	Emirates
	2020-02-02	22	ERJ142	Economy Plus	1	AGB	220	Jet Airways
	2020-03-03	21	CRJ900	Bussiness	1	BOH	490	British Airways
	2020-04-04	4	767-301ER	First Class	1	AGB	390	Emirates
	2020-05-05	5	ERJ142	Economy	1	CTM	120	Jet Airways

ticket_details 8 ×

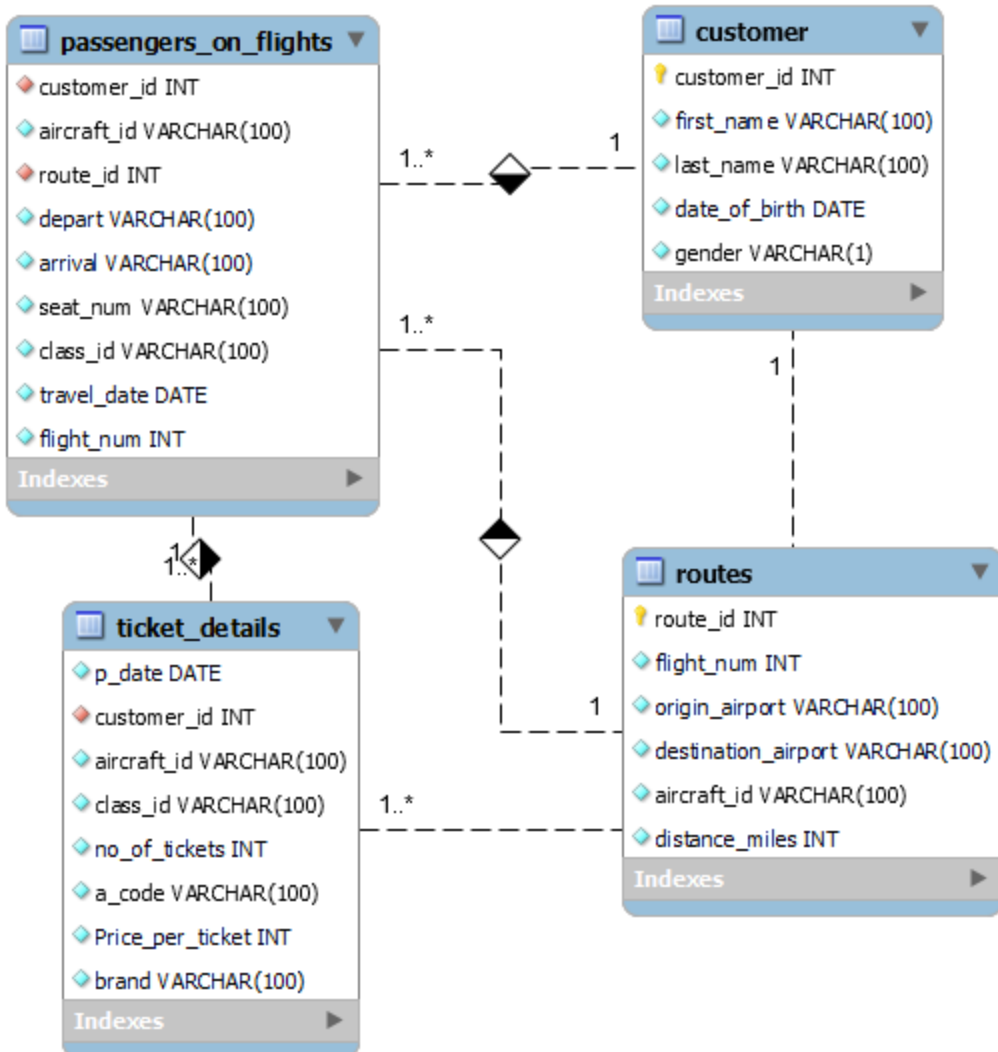
Read Only

Output

Action Output

#	Time	Action	Message	Duration / Fetch
✓ 37	18:32:51	SELECT*FROM air_cargo.passengers_on_flights LIMIT 0, 2000	50 row(s) returned	0.000 sec / 0.000 sec
✓ 38	18:33:23	SELECT*FROM air_cargo.routes LIMIT 0, 2000	49 row(s) returned	0.000 sec / 0.000 sec
✓ 39	18:33:54	SELECT*FROM air_cargo.ticket_details LIMIT 0, 2000	50 row(s) returned	0.000 sec / 0.000 sec

#	Time	Action	Message	Duration / Fetch
51	18:42:23	SET FOREIGN_KEY_CHECKS=0	0 row(s) affected	0.015 sec
52	18:42:24	SET GLOBAL FOREIGN_KEY_CHECKS=0	0 row(s) affected	0.031 sec
53	18:43:10	ALTER TABLE air_cargo.passengers_on_flights ADD FOREIGN KEY (customer_id) REFERENCES ...	0 row(s) affected Records: 0 Duplicates: 0 Warnings: 0	0.500 sec
54	18:43:16	DESCRIBE air_cargo.passengers_on_flights	9 row(s) returned	0.000 sec / 0.000 sec
55	18:43:35	Apply changes to passengers_on_flights		
56	18:45:26	Apply changes to routes	Changes applied	
57	18:46:02	ALTER TABLE air_cargo.passengers_on_flights ADD FOREIGN KEY (route_id) REFERENCES air_c...	0 row(s) affected Records: 0 Duplicates: 0 Warnings: 0	0.266 sec
58	18:46:16	DESCRIBE air_cargo.routes	6 row(s) returned	0.000 sec / 0.000 sec





```
36
37 /* Task 02: Write a query to create route_details table using suitable data types for the fields, such as route_id, flight_num, origin_airport,
38           destination_airport, aircraft_id, and distance_miles. Implement the check constraint for the flight number and unique constraint for the route_id
39           fields. Also, make sure that the distance miles field is greater than 0 */
40 • CREATE TABLE air_cargo.route_details
41 (
42   route_id INT UNIQUE,
43   flight_num INT check(flight_num >0),
44   origin_airport VARCHAR(50),
45   destination_airport VARCHAR(50),
46   aircraft_id INT,
47   distance_miles INT CHECK (distance_miles >0)
```

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

	Field	Type	Null	Key	Default	Extra
▶	route_id	int	YES	UNI	NULL	
	flight_num	int	YES		NULL	
	origin_airport	varchar(50)	YES		NULL	
	destination_airport	varchar(50)	YES		NULL	
	aircraft_id	int	YES		NULL	
	distance_miles	int	YES		NULL	

Result 13 ×



Output

Action Output

#	Time	Action	Message	Duration / Fetch
✖ 64	19:20:08	CREATE TABLE air_cargo.route_details (route_id INT UNIQUE, flight_num INT check(flight_num >0), origin_airpo...	Error Code: 1050. Table 'route_details' already exists	0.078 sec
✔ 65	19:20:21	DROP TABLE air_cargo.route_details	0 row(s) affected	0.219 sec

Query Completed



	Field	Type	Null	Key	Default	Extra
▶	route_id	int	YES	UNI	NULL	
	flight_num	int	YES		NULL	
	origin_airport	varchar(50)	YES		NULL	
	destination_airport	varchar(50)	YES		NULL	
	aircraft_id	int	YES		NULL	
	distance_miles	int	YES		NULL	

#	Time	Action	Message	Duration / Fetch
66	19:20:29	CREATE TABLE air_cargo.route_details (route_id INT UNIQUE, flight_num INT check(flight_num >0), origin_airport V...	0 row(s) affected	0.328 sec
67	19:20:44	DESCRIBE air_cargo.route_details	6 row(s) returned	0.000 sec / ...



Air Cargo Analysis Project ×



```
48 )
49 ENGINE=INNODB;
50 • DESCRIBE air_cargo.route_details;
51
52 -- Task 03: 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.
53 • SELECT*FROM passengers_on_flights WHERE route_id BETWEEN 1 AND 25 ORDER BY route_id DESC;
54
55 -- Task 04: Write a query to identify the number of passengers and total revenue in business class from the ticket_details table.
56 • SELECT COUNT(customer_id) AS number_of_passengers, SUM(Price_per_ticket) AS total_revenue_in_business FROM air_cargo.ticket_details WHERE class_id = 'Bussiness';
57
58
59
```

Result Grid Filter Rows: Export: Wrap Cell Contents:

	number_of_passengers	total_revenue_in_business
▶	13	6034



Result 15 ×

Read Only

Output

Action Output


#	Time	Action	Message	Duration / Fetch
✓ 68	19:25:17	SELECT*FROM passengers_on_flights WHERE route_id BETWEEN 1 AND 25 ORDER BY route_id DESC LIM...	26 row(s) returned	0.218 sec / 0.000 sec
✓ 69	19:29:04	SELECT COUNT(customer_id) AS number_of_passengers, SUM(Price_per_ticket) AS total_revenue_in_business ...	1 row(s) returned	0.328 sec / 0.000 sec

	full_name
▶	Julie,Sam
	Steve,Ryan
	Morris,Lois
	Cathenna,Emily
	Aaron,Kim
	Alexander.Scot

Result Grid

Form Editor

Read Only

 Action Output



Air Cargo Analysis Project x



Limit to 2000 rows

57

58 -- Task 05: Write a query to display the full name of the customer by extracting the first name and last name from the customer table.

59 • SELECT CONCAT(first_name, ",", last_name) AS full_name FROM customer;

60

61 -- Task 06: Write a query to extract the customers who have registered and booked a ticket. Use data from the customer and ticket_details tables.

62 • SELECT c.customer_id, t.no_of_tickets, t.class_id

63 FROM air_cargo.customer c JOIN ticket_details t

64 ON c.customer_id = t.customer_id

65 WHERE no_of_tickets > 0;

66

67 -- Task 07: 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.

68 -- Task 08: 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.

Result Grid Filter Rows: Export: Wrap Cell Content:

	customer_id	no_of_tickets	class_id
▶	27	1	Economy
	22	1	Economy Plus
	21	1	Bussiness
	4	1	First Class
	5	1	Economy
	7	1	Russiness

Result 17 x



Output

Action Output

#	Time	Action	Message	Duration / Fetch
✓ 70	19:31:11	SELECT CONCAT(first_name, ",", last_name) AS full_name FROM customer LIMIT 0, 2000	50 row(s) returned	0.188 sec / 0.000 sec
✓ 71	19:37:46	SELECT c.customer_id, t.no_of_tickets, t.class_id FROM air_cargo.customer c JOIN ticket_details t ON c.custom...	50 row(s) returned	0.000 sec / 0.000 sec

	first_name	last_name	customer_id	brand
▶	Cherly	Vernon	27	Emirates
	Cathenna	Emily	4	Emirates
	Anderson	Stewart	7	Emirates
	Leo	Travis	9	Emirates
	Roger	Walson	11	Emirates
	Moss	Morris	25	Emirates

Result Grid

Form Editor

Read Only

Action Output

	first_name	last_name	class_id
▶	Julie	Sam	Economy Plus
	Floyd	Ted	Economy Plus
	Roger	Walson	Economy Plus
	Catherine	Shad	Economy Plus
	Joyce	Paul	Economy Plus
	Phenv	Fri	Economy Plus

Result 19 Read Only

Action Output



Air Cargo Analysis Project ×



Limit to 2000 rows

```
73 -- Task 08: 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.
74 • SELECT c.first_name, c.last_name, p.class_id
75 FROM customer c JOIN passengers_on_flights p
76 ON c.customer_id = p.customer_id
77 GROUP BY c.first_name, p.class_id
78 HAVING p.class_id = "Economy Plus";
79
80 -- Task 09: Write a query to identify whether the revenue has crossed 10000 using the IF clause on the ticket_details table.
81 • SELECT SUM(Price_per_ticket) AS total_revenue,
82 IF(SUM(Price_per_ticket) > 10000, "Yes - revenue has crossed 10000", "NO - revenue has not crossed 10000") AS revenue_crossed_status FROM ticket_details;
83
84
```

Result Grid Filter Rows: Export: Wrap Cell Content: [IA](#)

	total_revenue	revenue_crossed_status
▶	15369	Yes - revenue has crossed 10000



Result 20 ×

! Read Only

Output








Action Output






#	Time	Action	Message	Duration / Fetch
✓ 73	19:45:04	SELECT c.first_name, c.last_name, p.class_id FROM customer c JOIN passengers_on_flights p ON c.customer_id...	9 row(s) returned	0.047 sec / 0.000 sec
✓ 74	19:49:34	SELECT SUM(Price_per_ticket) AS total_revenue, IF(SUM(Price_per_ticket) > 10000, "Yes - revenue has crosse...	1 row(s) returned	0.031 sec / 0.000 sec

MySQL Workbench

Practice

FileEditViewQueryDatabaseServerToolsScriptingHelp





Air Cargo Analysis Project

77GROUP BY c.first_name, p.class_id

78HAVING p.class_id = "Economy Plus";

79

80-- Task 09: Write a query to identify whether the revenue has crossed 10000 using the IF clause on the ticket_details table.

81• SELECT SUM(Price_per_ticket) AS total_revenue,

82IF(SUM(Price_per_ticket) > 10000, "Yes - revenue has crossed 10000", "NO - revenue has not crossed 10000") AS revenue_crossed_status FROM ticket_details;

83

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

85• GRANT

86ALL ON *.* TO 'root'@'localhost';

87-- Grant access to air_cargo dataset

88• GRANT

89ALL ON air_cargo TO 'root'@'localhost';

90

91

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

93-- Task 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.

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

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

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

Output

Action Output

#	Time	Action	Message	Duration / Fetch
82	19:55:04	GRANT ALL ON *.* TO 'root'@'localhost'	0 row(s) affected	0.125 sec
83	19:55:53	GRANT ALL ON air_cargo TO 'root'@'localhost'	0 row(s) affected	0.094 sec

Query Completed



Air Cargo Analysis Project ×



```
82 IF(SUM(Price_per_ticket) > 10000, "Yes - revenue has crossed 10000", "NO - revenue has not crossed 10000") AS revenue_crossed_status FROM ticket_details;
83
84 -- Task 10: Write a query to create and grant access to a new user to perform operations on a database.
85 • GRANT
86 ALL ON *.* TO 'root'@'localhost';
87 -- Grant access to air_cargo dataset
88 • GRANT
89 ALL ON air_cargo TO 'root'@'localhost';
90
91 -- Task 11: Write a query to find the maximum ticket price for each class using window functions on the ticket_details table
92 • SELECT class_id AS class, MAX(Price_per_ticket) AS maximum_ticket_price FROM air_cargo.ticket_details GROUP BY class_id ORDER BY class_id;
93
```

Result Grid Filter Rows: Export: Wrap Cell Content:

	class	maximum_ticket_price
►	Bussiness	510
	Economy	190
	Economy Plus	295
	First Class	395



Result 21 ×

Read Only


Output

Action Output

#	Time	Action	Message	Duration / Fetch
✓ 83	19:55:53	GRANT ALL ON air_cargo TO 'root'@'localhost'	0 row(s) affected	0.094 sec
✓ 84	19:59:23	SELECT class_id AS class, MAX(Price_per_ticket) AS maximum_ticket_price FROM air_cargo.ticket_details GRO...	4 row(s) returned	0.047 sec / 0.000 sec

	customer_id	route_id
▶	2	4
	4	4
	11	4

Read Only

 Action Output

	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	1114
	4	767-301ER	4	JFK	LAX	03FC	First Class	2020-04-30	1114
	11	767-301ER	4	JFK	LAX	05B	Bussiness	2020-11-09	1114

Read Only

#	Time	Action	Message	Duration / Fetch
91	20:06:22	create view execution_plan as select c.first_name, c.last_name , p.* from customer c inner join passengers_on_fli...	Error Code: 1050. Table 'execution_plan' already exists	0.000 sec
92	20:06:24	SELECT * FROM execution_plan LIMIT 0, 2000	3 row(s) returned	0.000 sec / 0.000 sec

Query Completed



Limit to 2000 rows



```
95 • SELECT customer_id, route_id FROM air_cargo.passengers_on_flights WHERE route_id = 4;
96
97 -- Task 13: For the route ID 4, write a query to view the execution plan of the passengers_on_flights table.
98 • SELECT * FROM air_cargo.passengers_on_flights WHERE route_id = 4;
99
100 -- Task 14: Write a query to calculate the total price of all tickets booked by a customer across different aircraft IDs using rollup function.
101 • SELECT customer_id, aircraft_id, SUM(Price_per_ticket) AS total_price_of_all_tickets FROM air_cargo.ticket_details GROUP BY customer_id with rollup;
102
103 -- Task 15: Write a query to create a view with only business class customers along with the brand of airlines.
104 • CREATE VIEW Bussiness_Class AS SELECT customer_id, brand FROM ticket_details WHERE class_id = 'Bussiness';
105 • SELECT*FROM Bussiness_Class;
106
```

Result Grid Filter Rows: Export: Wrap Cell Content:

	customer_id	brand
▶	21	Bristish Airways
	7	Emirates
	11	Emirates
	25	Emirates
	24	Qatar Airways
	29	Qatar Airways

Bussiness_Class 28 ×






Output

Action Output

#	Time	Action	Message	Duration / Fetch
✓ 96	20:14:51	CREATE VIEW Bussiness_Class AS SELECT customer_id, brand FROM ticket_details WHERE class_id = 'Bussin...	0 row(s) affected	0.172 sec
✓ 97	20:15:05	SELECT*FROM Bussiness_Class LIMIT 0, 2000	13 row(s) returned	0.000 sec / 0.000 sec


```
106 |
107 | /* Task 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.
108 |      Also, return an error message if the table doesn't exist. */
109 | DELIMITER &&
110 | CREATE PROCEDURE get_total_passengers_()
111 | BEGIN
112 |     DECLARE total_passengers INT DEFAULT 0;
113 |     SELECT COUNT(*) INTO total_passengers FROM air_cargo.passengers_on_flights;
114 |     SELECT total_passengers;
115 | END &&
116 | DELIMITER ;
117 |
```

Result Grid  Filter Rows: Export:  Wrap Cell Content: 

	customer_id	brand
▶	21	British Airways
	7	Emirates
	11	Emirates
	25	Emirates
	24	Qatar Airways
	29	Qatar Airways

Business_Class 29 Read Only

Action Output

#	Time	Action	Message	Duration / Fetch
97	20:15:05	SELECT*FROM Bussiness_Class LIMIT 0, 2000	13 row(s) returned	0.000 sec / 0.000 sec
98	20:22:30	SELECT*FROM Bussiness_Class LIMIT 0, 2000	13 row(s) returned	0.000 sec / 0.000 sec

	Db	Name	Type	Definer	Modified	Created	Security_type	Comment	character_set_client	collation_connection	Database Collation
▶	employee	Get_Employee_Exp	PROCEDURE	root@localhost	2022-10-30 14:48:18	2022-10-30 14:48:18	DEFINER		utf8mb4	utf8mb4_0900_ai_ci	utf8mb4_0900_ai_ci
	employee_db	abcd	PROCEDURE	root@localhost	2022-10-10 18:47:15	2022-10-10 18:47:15	DEFINER		utf8mb4	utf8mb4_0900_ai_ci	utf8mb4_0900_ai_ci
	employee_db	Employee_of_Auto	PROCEDURE	root@localhost	2022-10-10 17:02:34	2022-10-10 17:02:34	DEFINER		utf8mb4	utf8mb4_0900_ai_ci	utf8mb4_0900_ai_ci
	employee_db	EvenLoop	PROCEDURE	root@localhost	2022-10-10 18:19:47	2022-10-10 18:19:47	DEFINER		utf8mb4	utf8mb4_0900_ai_ci	utf8mb4_0900_ai_ci
	employee_db	EvenRepeatLoop	PROCEDURE	root@localhost	2022-10-10 18:13:23	2022-10-10 18:13:23	DEFINER		utf8mb4	utf8mb4_0900_ai_ci	utf8mb4_0900_ai_ci

Read Only

Action Output

	route_id	flight_num	origin_airport	destination_airport	aircraft_id	distance_miles
▶	1	1111	EWB	HNL	767-301ER	4962
	2	1112	HNL	EWB	767-301ER	4962
	3	1113	EWB	LHR	A321	3466
	4	1114	JFK	LAX	767-301ER	2475
	5	1115	LAX	JFK	767-301ER	2475
	6	1116	HNL	LAX	767-301ER	2556

Read Only

Action Output

MySQL Workbench

Practice

FileEditViewQueryDatabaseServerToolsScriptingHelp

SQL

SQL

Limit to 2000 rows

Air Cargo Analysis Project

126

127

128

129

130

131

132

133

134

135

136

137

/* Task 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 FUNCTION groups_distance(dist int)

RETURNS VARCHAR(10) DETERMINISTIC

BEGIN

DECLARE distance_categories CHAR(3);

IF dist BETWEEN 0 AND 2000 THEN SET distance_categories = 'SDT';

ELSEIF dist BETWEEN 2001 AND 6500 THEN SET distance_categories = 'IDT';

ELSEIF dist > 6500 THEN SET distance_categories = 'LDT';

END IF;

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

Result 31

Output

Action Output

#	Time	Action	Message	Duration / Fetch
✓ 104	20:26:49	CREATE PROCEDURE distance_miles() BEGIN SELECT*FROM routes WHERE distance_miles > 2000; END	0 row(s) affected	0.141 sec
✓ 105	20:27:06	CALL distance_miles(); /* Task 18: Write a query to create a stored procedure that groups the distance travelled b...	24 row(s) returned	0.171 sec / 0.000 sec

Query Completed



Air Cargo Analysis Project x



Limit to 2000 rows

```
139 END $$
140 DELIMITER $$;
141
142 /* Task 19: Write a query to extract ticket purchase date, customer ID, class ID and specify if the complimentary services are provided for the specific
143 class using a stored function in stored procedure on the ticket_details table.
144 Condition: If the class is Business and Economy Plus, then complimentary services are given as Yes, else it is No */
145 SELECT p_date, customer_id, class_id,
146 CASE
147 WHEN class_id = 'Business'
148 OR class_id = 'Economy Plus' THEN 'YES' ELSE 'NO'
149 END AS Complimentary_Service
150 FROM ticket_details ORDER BY customer_id;
```

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

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

Result 32 x



Read Only

Output

Action Output

#	Time	Action	Message	Duration / Fetch
❌ 112	20:42:59	SELECT* FROM distance_categories; /* Task 19: Write a query to extract ticket purchase date, customer ID, cl...	Error Code: 1146. Table 'air_cargo.distance_categories' doesn't exist	0.000 sec
✅ 113	20:48:32	SELECT p_date, customer_id, class_id, CASE WHEN class_id = 'Business' OR class_id = 'Economy Plus' THEN ...	50 row(s) returned	0.000 sec / 0.000 sec



Air Cargo Analysis Project* x



Limit to 2000 rows

```
142  /* Task 19: Write a query to extract ticket purchase date, customer ID, class ID and specify if the complimentary services are provided for the specific
143          class using a stored function in stored procedure on the ticket_details table.
144          Condition: If the class is Business and Economy Plus, then complimentary services are given as Yes, else it is No */
145  SELECT p_date, customer_id, class_id,
146  CASE
147  WHEN class_id = 'Business'
148  OR class_id = 'Economy Plus' THEN 'YES' ELSE 'NO'
149  END AS Complimentary_Service
150  FROM ticket_details ORDER BY customer_id;
151
152  -- Task 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.
153  SELECT*FROM customer WHERE last_name = 'Scott';
```

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

	customer_id	first_name	last_name	date_of_birth	gender
▶	37	Samuel	Scott	2000-01-28	M
	38	Alexis	Scott	2001-10-31	M

Result Grid

Result 33 Result 34 x

Read Only

Output

Action Output

#	Time	Action	Message	Duration / Fetch
✓ 114	20:49:32	SELECT p_date, customer_id, class_id, CASE WHEN class_id = 'Business' OR class_id = 'Economy Plus' THEN ...	50 row(s) returned	0.000 sec / 0.000 sec
✓ 115	20:49:32	SELECT p_date, customer_id, class_id, CASE WHEN class_id = 'Business' OR class_id = 'Economy Plus' THEN ...	2 row(s) returned	- / 0.000 sec

Query Completed