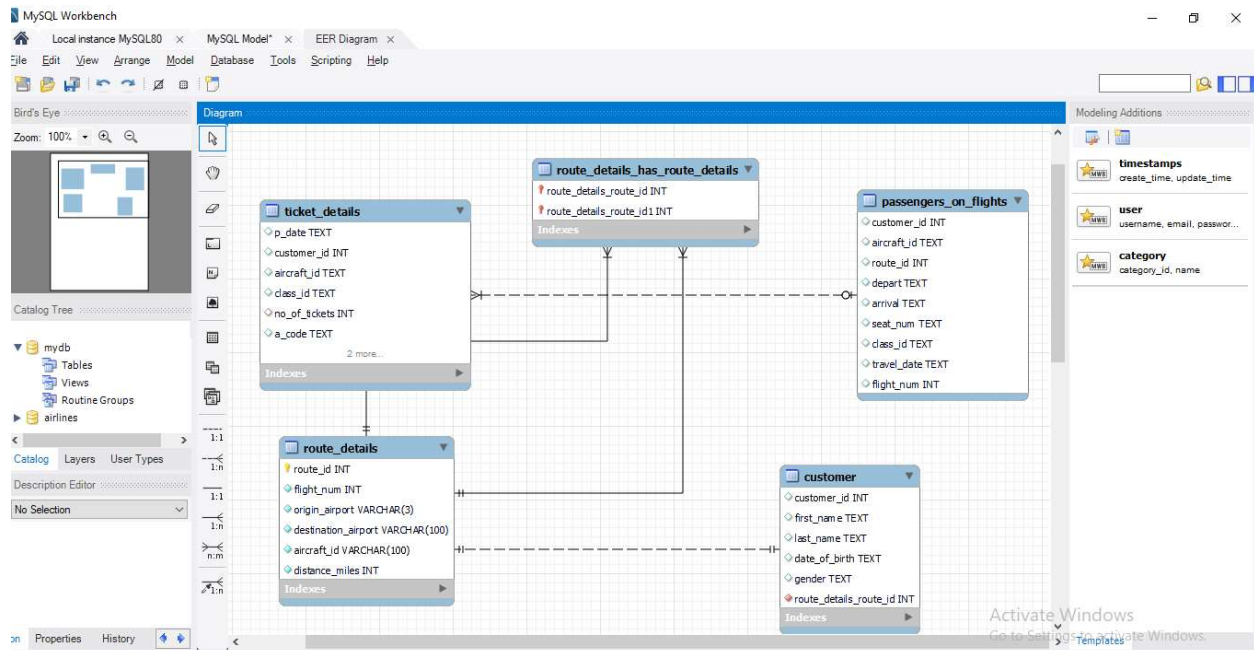


# Air Cargo Analysis.

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.

SCHEMAS

Filter objects

airlines

- Tables
  - customer
  - passengers\_on\_flight
  - route\_details
  - ticket\_details
- Views
- Stored Procedures
- Functions

employee\_db

- Tables
- Views
- Stored Procedures
- Functions

new\_schema

- Tables
- Views

Administration Schemas

Information

```
1
2 use airlines;
3
4 describe route_details;
```

Result Grid

Field	Type	Null	Key	Default	Extra
route_id	int	NO	PRI	NULL	
flight_num	int	NO		NULL	
origin_airport	varchar(3)	NO		NULL	
destination_airport	varchar(100)	NO		NULL	
aircraft_id	varchar(100)	NO		NULL	

Result 12 x

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.

The screenshot shows a database query editor with a schema tree on the left and a query window on the right. The schema tree includes 'airlines' (Tables: customer, passengers\_on\_flight, route\_details, ticket\_details; Views: Views; Stored Procedures: Stored Procedures; Functions: Functions), 'employee\_db' (Tables: Tables; Views: Views; Stored Procedures: Stored Procedures; Functions: Functions), and 'new\_schema' (Tables: Tables; Views: Views). The query window contains the following SQL code:

```
1
2 use airlines;
3
4 describe route_details;
5
6 select * from passengers_on_flights where route_id between 1 and 25;
```

The result grid shows the following data:

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	Business	02-07-2018	1122
5	ERJ142	18	ANI	BGR	02E	Economy	06-05-2020	1128
4	767-301ER	5	LAX	JFK	02FC	First Class	06-04-2020	1115

4. Write a query to identify the number of passengers and total revenue in business class from the ticket\_details table.

The screenshot shows a database query editor with a schema tree on the left and a query window on the right. The schema tree is the same as in the previous screenshot. The query window contains the following SQL code:

```
1
2
3
4 select count(customer_id) as number_of_passengers, sum(price_per_ticket) as Total_Revenue
5 from ticket_details where class_id="Business";
```

The result grid shows the following data:

number_of_passengers	Total_Revenue
13	6034

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

The screenshot shows a database query editor with a schema tree on the left and a query window on the right. The schema tree is the same as in the previous screenshots. The query window contains the following SQL code:

```
1
2
3
4 select concat(first_name,last_name) as FullName from customer;
```

The result grid shows the following data:

FullName
JulieSam
SteveRyan
MorrisLois
CatherinaEmily
AaronKim

6. Write a query to extract the customers who have registered and booked a ticket. Use data from the customer and ticket\_details tables.

The screenshot shows a SQL query editor with the following query:

```
select customer.first_name, customer.last_name from customer left
join ticket_details on customer.customer_id=ticket_details.customer_id;
```

The result grid displays the following data:

first_name	last_name
Julie	Sam
Julie	Sam
Steve	Ryan
Steve	Ryan
Morris	Lois

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.

The screenshot shows a SQL query editor with the following query:

```
select customer.first_name, customer.last_name from customer
left join ticket_details on customer.customer_id=ticket_details.customer_id
where ticket_details.brand="Emirates";
```

The result grid displays the following data:

first_name	last_name
Steve	Ryan
Cathenna	Emily
Cathenna	Emily
Aaron	Kim
Anderson	Stewart

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.

The screenshot shows a SQL query editor with the following query:

```
select customer_id, class_id from passengers_on_flights where class_id="Economy Plus";
select customer_id from passengers_on_flights where class_id="Economy Plus";
select customer_id, count(class_id) from passengers_on_flights where class_id="Economy Plus"
group by customer_id having count(class_id)<3;
```

The result grid displays the following data:

customer_id	count(class_id)
1	1
8	1
11	1
17	1
19	2

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

Navigator: Query 1

SCHEMAS

Filter objects

airlines

- Tables
  - customer
  - passengers\_on\_flight
  - route\_details
  - ticket\_details
- Views
- Stored Procedures
- Functions

employee\_db

- Tables
- Views
- Stored Procedures
- Functions

new\_schema

- Tables
- Views

Administration Schemas

Query 1

```

1
2
3
4 select * from ticket_details;
5 select sum(price_per_ticket) as total_revenue from ticket_details;
6 select if(15639>10000,"Revenue has crossed 10000","Revenue has not crossed 10000");

```

Result Grid

if(15639>10000,"Revenue has crossed 10000","Revenue has not crossed 10000")
Revenue has crossed 10000

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

Navigator: Query 1

SCHEMAS

Filter objects

airlines

- Tables
  - customer
  - passengers\_on\_flight
  - route\_details
  - ticket\_details
- Views
- Stored Procedures
- Functions

employee\_db

- Tables
- Views
- Stored Procedures
- Functions

new\_schema

- Tables
- Views

Administration Schemas

Query 1

```

1 create user "shivani";
2 grant select on customer to shivani;
3 grant select on passengers_on_flight to shivani;
4 grant select on route_details to shivani;
5 grant select on ticket_details to shivani;

```

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

Navigator: Query 1

SCHEMAS

Filter objects

airlines

- Tables
  - customer
  - passengers\_on\_flight
  - route\_details
  - ticket\_details
- Views
- Stored Procedures
- Functions

employee\_db

- Tables
- Views
- Stored Procedures
- Functions

new\_schema

- Tables
- Views

Administration Schemas

Query 1

```

1
2
3
4 select class_id,max(Price_per_ticket) over (Partition by class_id)
5 as Max_ticketpriceforeach_class from ticket_details;
6 select distinct(class_id),max(Price_per_ticket) over (Partition by class_id)
7 as Max_ticketpriceforeach_class from ticket_details;

```

Result Grid

class_id	Max_ticketpriceforeach_class
Business	510
Economy	190
Economy Plus	295
First Class	395

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.

The screenshot shows the SQL Server Enterprise Manager interface. The left pane displays the 'airlines' schema with tables 'customer', 'passengers\_on\_flight', 'route\_details', and 'ticket\_details'. The right pane shows a query window with the following SQL code:

```
1
2
3
4 • select customer_id from passengers_on_flights where route_id=4;
5 • select first_name,last_name from customer where customer_id in(2,4,11);
```

The 'Result Grid' at the bottom shows the output of the second query:

first_name	last_name
Steve	Ryan
Cathenna	Emily
Roger	Walson

13. For the route ID 4, write a query to view the execution plan of the passengers\_on\_flights table.

The screenshot shows the SQL Server Enterprise Manager interface. The left pane displays the 'airlines' schema with tables 'customer', 'passengers\_on\_flight', 'route\_details', and 'ticket\_details'. The right pane shows a query window with the following SQL code:

```
1
2
3
4 • select * from passengers_on_flights;
5 • create view myflights
6 • as select * from passengers_on_flights where route_id=4;
7 • select * from myflights;
```

The 'Result Grid' at the bottom shows the output of the third query:

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

The screenshot shows the SQL Server Enterprise Manager interface. The left pane displays the 'airlines' schema with tables 'customer', 'passengers\_on\_flight', 'route\_details', and 'ticket\_details'. The right pane shows a query window with the following SQL code:

```
1
2
3
4 • select aircraft_id,customer_id,sum(price_per_ticket) as TotalPrice
5 • from ticket_details group by aircraft_id,customer_id with rollup;
```

The 'Result Grid' at the bottom shows the output of the query:

aircraft_id	customer_id	TotalPrice
767-301ER	2	130
767-301ER	4	780
767-301ER	5	430
767-301ER	7	430
767-301ER	9	380



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

Query 1

```

1
2
3
4 select * from ticket_details;
5 create view airlinesbrand as
6 select customer_id,class_id,brand,price_per_ticket
7 from ticket_details where class_id="Business";
8 select * from airlinesbrand;

```

Result Grid

customer_id	class_id	brand	price_per_ticket
21	Business	British Airways	490
7	Business	Emirates	430
11	Business	Emirates	465
25	Business	Emirates	499
24	Business	Qatar Airways	480

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.

Query 1

```

1
2
3
4 Delimiter $$
5 drop procedure passengerdetails;
6 create procedure passengerdetails()
7
8 begin
9 select customer.first_name,customer.last_name,passengers_on_flights.route_id
10 from customer left join passengers_on_flights on
11 customer.customer_id=passengers_on_flights.customer_id;
12 end;
13
14 call passengerdetails(); /* Details of the passengers with route id */
15

```

Output

#	Time	Action	Message	Duration / Fetch
138	18:46:03	drop procedure passengerdetails; create procedure passengerdetails() begin select customer...	Error Code: 1305. PROCEDURE airlines.passengerdetails does not exist	0.000 sec

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.

Query 1

```

1 select * from route_details;
2 Delimiter $$
3 create procedure myroutes_1()
4 begin
5 select * from route_details where distance_miles>2000;
6 end $$
7
8 call myroutes_1();

```

Result Grid

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

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  $\geq 0$  AND  $\leq 2000$  miles, intermediate distance travel (IDT) for  $>2000$  AND  $\leq 6500$ , and long-distance travel (LDT) for  $>6500$ .

```

1 drop procedure if exists flightanalysis;
2 delimiter $$
3 create procedure flightanalysis()
4 begin
5     select * from route_details;
6     select flight_num,distance_miles as shortdistance from routes where distance_miles between 0 and 2000;
7     select flight_num,distance_miles as intermediatedistance from routes where distance_miles between 2000 and 6500;
8     select flight_num,distance_miles as LongDistance from routes where distance_miles >6500;
9 end
10 delimiter ;

```

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

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.

Condition:

- If the class is *Business* and *Economy Plus*, then complimentary services are given as Yes, else it is No

```

1 drop procedure if exists ticketanalysis;
2 delimiter $$
3 create procedure ticketanalysis()
4 begin
5     select p_date,customer_id,class_id from ticket_details;
6     select class_id as firstclass from ticket_details where class_id="Business";
7     alter table ticket_details add column Comperv varchar(100);
8     update ticket_details set Comperv="Yes" where class_id="Business";
9 end
10 delimiter ;

```

firstclass
Business
Business
Business
Business
Business
Business

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.

```

1 delimiter $$
2 create procedure finalnames()
3 begin
4     declare c1 cursor for select last_name from customer where last_name like '%Scott';
5     open c1;
6     select * from customer where last_name like '%Scott';
7 end
8 delimiter ;

```

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

