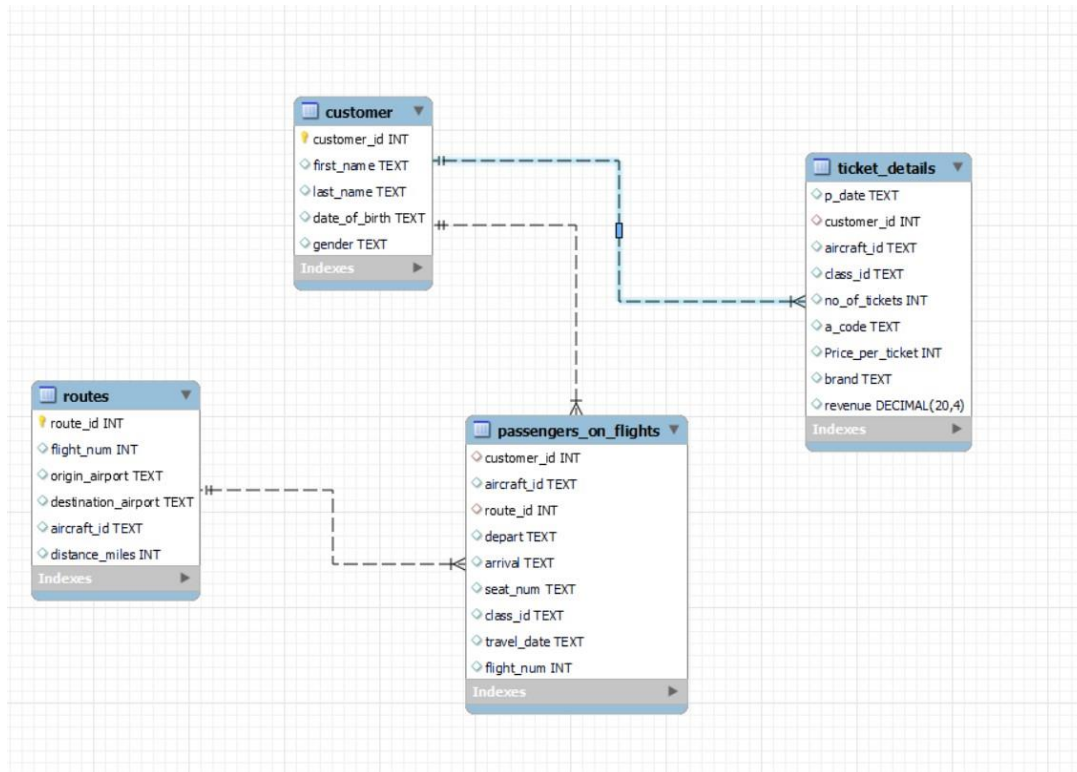


Q1



Raymond Ya

Q2

The screenshot displays a database management interface with three main sections: a Navigator on the left, a Query editor in the center, and an Output window at the bottom right.

**Navigator:** Shows a tree view of the database schema. Under the 'project' schema, the 'route\_details' table is highlighted. Other tables listed include 'customer', 'passengers\_on\_flights', 'routes', and 'ticket\_details'. Views, Stored Procedures, and Functions are also listed.

**Query 1:** The SQL query is as follows:

```
1 Create table route_details (  
2   route_id INT NOT NULL,  
3   flight_num INT NOT NULL,  
4   origin_airport varchar (255),  
5   destination_airport varchar (255),  
6   aircraft_id INT NOT NULL,  
7   distance_miles INT,  
8   Unique (route_id),  
9   check (distance_miles > 0));
```

**Table: route\_details**

**Columns:**

Column Name	Data Type	Constraints
route_id	int	pk
flight_num	int	
origin_airport	varchar	
destination_airport	varchar	
aircraft_id	int	
distance_miles	int	

**Output:** The output window shows the execution results of the query. It includes a table with columns for the action number, time, and the action performed.

#	Time	Action
8	11:30:15	Create table route_details ( route_id INT N
9	11:39:11	Create table route_details ( Unique (route_
10	11:39:19	DROP TABLE 'project'.route_details'

Raymond Ya

Q3

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

Result Grid								
Filter Rows:								
Export:   Wrap Cell Content:								
	customer_id	aircraft_id	route_id	depart	arrival	seat_num	class_id	travel_date
2		767-301ER	4	JFK	LAX	01E	Economy	02-09-2018
1		ERJ142	9	DEN	LAX	01EP	Economy Plus	26-12-2019
5		767-301ER	12	ABI	ADK	02B	Bussiness	02-07-2018
5		ERJ142	18	ANI	BGR	02E	Economy	06-05-2020
4		767-301ER	5	LAX	JFX	02FC	First Class	06-04-2020
7		767-301ER	20	AVL	BOI	03B	Bussiness	08-07-2020
5		ERJ142	22	BGR	BJI	03E	Economy	31-05-2020
4		767-301ER	4	JFK	LAX	03FC	First Class	30-04-2020
11		767-301ER	5	LAX	JFX	04B	Bussiness	12-11-2020
17		A321	13	ABI	ADK	04EP	Economy Plus	03-06-2019
9		767-301ER	15	CAK	ANI	04FC	First Class	10-09-2020
11		767-301ER	4	JFK	LAX	05B	Bussiness	09-11-2020
10		A321	10	HNL	DEN	05E	Economy	11-10-2020

Q4

```
1  Select sum(no_of_tickets * price_per_ticket), count(customer_id) from ticket_details where class_id='Bussiness';
```



Result Grid

	sum(no_of_tickets * price_per_ticket)	count(customer_id)
▶	6034	13

Raymond Ya

Q5

```
1  select first_name, last_name from customer;
```

Result Grid		
Filter Rows: <input type="text"/>		
Export: 		
Wrap Cell Content: 		
	first_name	last_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

Raymond Ya

Q6

```
1  Select * from customer c inner 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	date_of_birth	gender	p_date	customer_id	aircraft_id	class_id	no_of_tickets	a_code	Price_per_ticket	brand
▶	27	Cherly	Vernon	19-03-1992	F	26-12-2018	27	767-301ER	Economy	1	DAL	130	Emirates
	22	Phenly	Eri	29-01-1999	M	02-02-2020	22	ERJ142	Economy Plus	1	AGB	220	Jet Airways
	21	Chirsty	Josh	10-01-2004	M	03-03-2020	21	CRJ900	Bussiness	1	BOH	490	British Airwa
	4	Cathenna	Emily	14-09-1977	F	04-04-2020	4	767-301ER	First Class	1	AGB	390	Emirates
	5	Aaron	Kim	18-02-1991	M	05-05-2020	5	ERJ142	Economy	1	CTM	120	Jet Airways
	7	Anderson	Stewart	11-01-1992	M	07-07-2020	7	767-301ER	Bussiness	1	BFS	430	Emirates
	8	Floyd	Ted	21-02-1993	M	08-08-2020	8	A321	Economy Plus	1	DAL	275	Qatar Airway
	9	Leo	Travis	22-03-1994	M	09-09-2020	9	767-301ER	First Class	1	BOH	380	Emirates
	10	Melvin	Tracy	23-04-1995	M	10-10-2020	10	A321	Economy	1	MCO	135	Qatar Airway
	11	Roger	Walson	24-05-1996	M	11-11-2020	11	767-301ER	Bussiness	1	AGB	465	Emirates
	19	Joyce	Paul	02-06-1990	F	12-12-2020	19	CRJ900	Economy Plus	1	DEN	225	British Airwa
	13	Solomon	Walter	26-07-1998	M	01-01-2019	13	A321	First Class	1	YVR	395	Qatar Airway

Raymond Ya

Q7




```
1 select first_name, last_name from customer c left 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			

Raymond Ya

Q8

```
1  Select * from passengers_on_flights group by customer_id having class_id='Economy Plus';
```

Result Grid    Filter Rows: <input type="text"/>   Export:  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	26-12-2019	1119
	8	A321	38	CST	DAL	02EP	Economy Plus	09-08-2020	1148
	11	ERJ142	31	BTM	CHA	03EP	Economy Plus	02-08-2018	1141
	17	A321	13	ABI	ADK	04EP	Economy Plus	03-06-2019	1123
	19	CRJ900	47	DAL	LAX	05EP	Economy Plus	13-01-2021	1157
	22	ERJ142	22	BGR	BJI	07EP	Economy Plus	09-02-2020	1132
	32	ERJ142	31	BTM	CHA	08EP	Economy Plus	04-03-2021	1141
	47	CRJ900	33	CDC	CST	09EP	Economy Plus	15-12-2020	1143
	50	A321	21	BFL	BET	10EP	Economy Plus	15-08-2020	1131



Q9

1

```
select if(sum(no_of_tickets*price_per_ticket)>10000,"Yes","No") as Result from ticket_details;
```

Result Grid

Filter Rows:

Export:

Wrap Cell Content:

Result

Yes

Raymond Ya

Q10

The screenshot shows the SQL Server Enterprise Manager interface. On the left, the 'SCHEMAS' pane displays a tree view with 'project' expanded, showing 'Tables' and 'Views'. The 'Views' folder is selected, and the 'business' view is highlighted. On the right, the SQL text area shows the definition of the 'business' view:

```
1 CREATE VIEW Business AS
2 SELECT p_date, customer_id, aircraft_id, class_id, brand
3 FROM ticket_details
4 WHERE class_id = 'Bussiness';
```

```
1 • select * from business;
```

Result Grid					
Filter Rows:		Export:		Wrap Cell Content:	
	p_date	customer_id	aircraft_id	class_id	brand
▶	03-03-2020	21	CRJ900	Bussiness	Bristish Airways
	07-07-2020	7	767-301ER	Bussiness	Emirates
	11-11-2020	11	767-301ER	Bussiness	Emirates
	03-03-2019	25	767-301ER	Bussiness	Emirates
	07-07-2019	24	A321	Bussiness	Qatar Airways
	22-10-2019	29	A321	Bussiness	Qatar Airways
	25-01-2019	2	A321	Bussiness	Qatar Airways
	01-04-2018	29	ERJ142	Bussiness	Jet Airways
	01-07-2018	5	767-301ER	Bussiness	Emirates
	01-11-2018	15	A321	Bussiness	Qatar Airways
	12-03-2020	33	CRJ900	Bussiness	Bristish Airways
	17-07-2020	49	767-301ER	Bussiness	Emirates
	08-11-2020	11	767-301ER	Bussiness	Emirates

# Q11

The screenshot displays a database management interface with two main windows. The top window shows the 'SCHEMAS' pane on the left, listing various database objects like 'route\_details', 'routes', 'ticket\_details', 'Views', 'Stored Procedures', 'Functions', 'sys', 'tutorial', 'Tables', and 'Views'. The main editor area contains the following SQL code:

```
1 delimiter &&
2 create procedure answer11()
3 begin
4 select * from routes where route_no >=route and route_no <= to_route;
5 end &&
6 delimiter ;
```

The bottom window shows the 'Administration' pane on the left, displaying the 'Table: routes' structure with columns: route\_id (int, PK), flight\_num (int), origin\_airport (text), destination\_airport (text), aircraft\_id (text), and distance\_miles (int). The main editor area contains the SQL code:

```
1 call answer11();
```

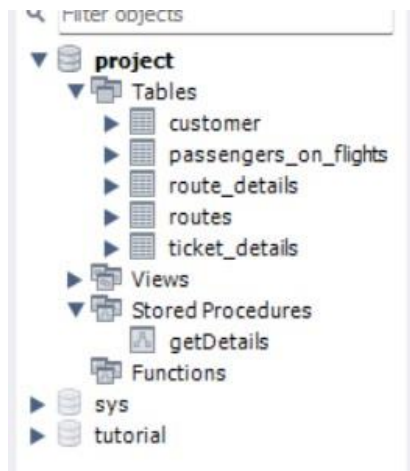
The 'Output' pane at the bottom shows the execution results:

#	Time	Action	Message
1	13:55:19	create procedure answer11() begin select * from routes where route_no >=route and route_no <= to_route; end	0 row(s) affected
2	13:56:19	call answer11()	Error Code: 1054. Unknown column 'route_no' in 'where clause'

On the right side of the bottom window, there is a button labeled 'Auto disable current toggle'.

Raymond Ya

Q12



```
1 delimiter &&
2 • create procedure getDetails()
3 begin
4 select * from routes where distance_miles>2000;
5 end &&
6 delimiter ;
```

A screenshot of a database IDE. The left pane shows the 'project' database structure. The top toolbar has icons for file operations, execution, and settings. The main editor shows the SQL command: `call getDetails();`. Below the editor, the 'Result Grid' displays the results of the stored procedure. The grid has columns: route\_id, flight\_num, origin\_airport, destination\_airport, aircraft\_id, and distance\_miles. The results are as follows:

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
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

The screenshot displays a database management interface with a left-hand sidebar and a main workspace. The sidebar, titled "SCHEMAS", contains a tree view of the database structure. Under the "project" schema, there are tables (customer, passengers\_on\_flights, route\_details, routes, ticket\_details), views, stored procedures (answer11, answer13, getDetails), and functions. The "Administration" tab is selected, and the "Schemas" sub-tab is active. Below the sidebar, a message states "No object selected".

The main workspace shows a SQL script being edited. The script is as follows:

```
1 delimiter &&
2 create procedure answer13()
3 begin
4     SELECT distance_miles,
5     CASE
6         WHEN distance_miles BETWEEN 0 and 2000 THEN 'SDT'
7         WHEN distance_miles > 6500 THEN 'LDT'
8         ELSE 'IDT'
9     END AS Quantity
10    FROM routes;
11 end &&
12 delimiter ;
13
```

Below the script editor, the "Output" pane shows the "Action Output" table, which records the execution of the script. The table has columns for "#", "Time", and "Action".

#	Time	Action
4	14:43:47	SELECT distance_miles, CASE WHEN distance_miles between 0 and 2000 THEN 'SDT' WHEN distanc..
5	14:45:38	create procedure answer13() begin SELECT distance_miles, CASE WHEN distance_miles between 0 and 2.
6	14:47:54	create procedure answer13() begin SELECT distance_miles, CASE WHEN distance_miles between 0 and 2.
7	14:49:58	SELECT distance_miles, CASE WHEN distance_miles BETWEEN 0 and 2000 THEN 'SDT' WHEN dista.
8	14:52:13	DROP PROCEDURE 'project'.answer13'
9	14:52:20	create procedure answer13() begin SELECT distance_miles, CASE WHEN distance_miles BETWEEN 0 an..

The screenshot shows a database management interface. On the left, the 'SCHEMAS' pane displays a tree view of the database structure. The 'project' database is expanded, showing tables (customer, passengers\_on\_flights, route\_details, routes, ticket\_details), views, stored procedures (answer11, answer13, getDetails), and functions. The 'sys' and 'tutorial' databases are also visible. Below the tree, the 'Administration' tab is active, and the 'Schemas' sub-tab is selected. The 'Information' section below shows 'No object selected'.

On the right, the 'call answer13();' query is executed. The 'Result Grid' displays the following data:

distance_miles	Quantity
4962	IDT
4962	IDT
3466	IDT
2475	IDT
2475	IDT
2556	IDT
1745	SDT
719	SDT
862	SDT
3365	IDT
4300	IDT
2232	IDT
2445	IDT
2000	SDT