- ```
  create database air_cargo;
  ```
- ```
  use air_cargo;
  ```

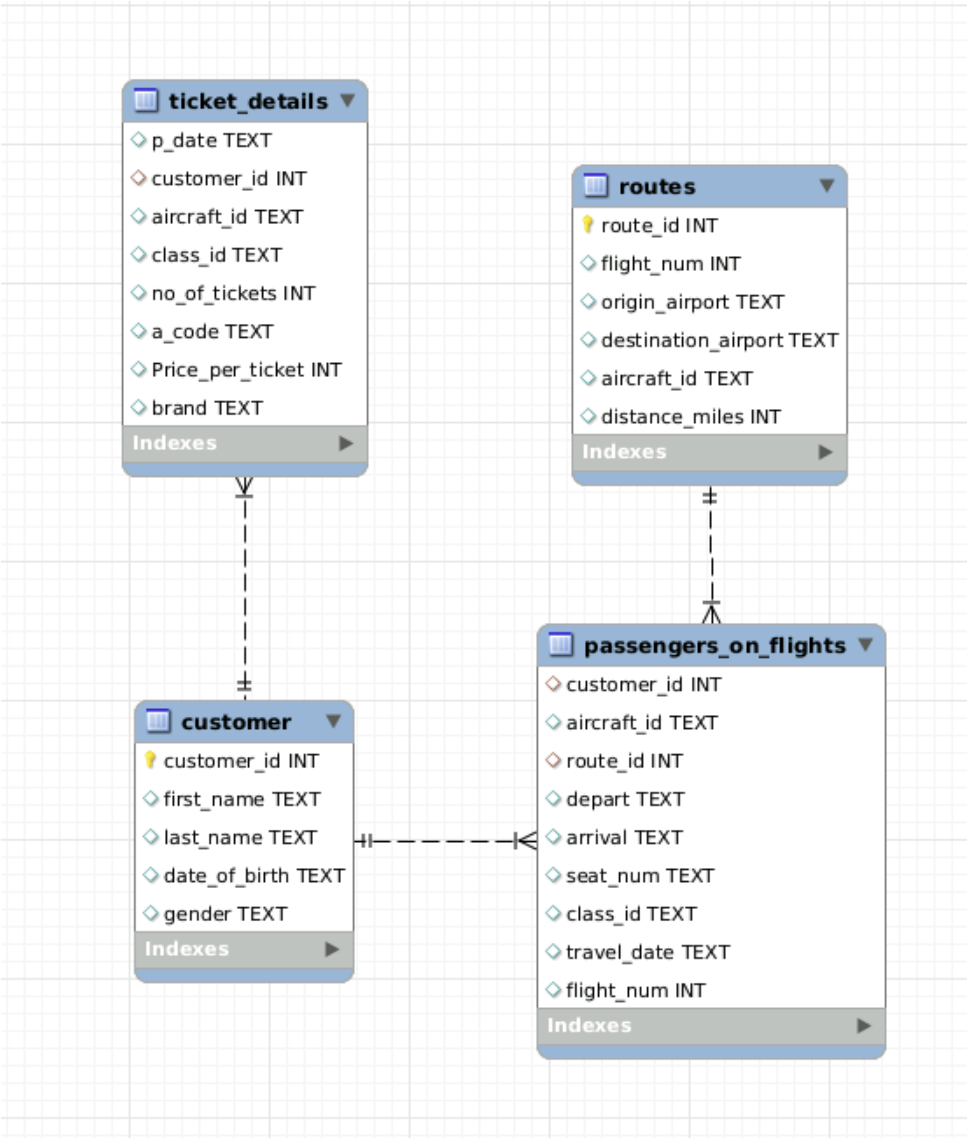
```
-- Tables have been imported using the table data import wizard
```

- ```
  alter table routes
  add constraint flt_num_chk check(flight_num is not null);
  ```

- ```
  alter table routes
  add constraint routes_unq unique(route_id);
  ```

- ```
  alter table routes
  add constraint distance_chk check(distance_miles>0);
  ```

Create an ER diagram for the given airlines database.


  ```

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.

#	first_name	last_name	route_id
1	Steve	Ryan	4
2	Julie	Sam	9
3	Aaron	Kim	12
4	Aaron	Kim	18
5	Cathenna	Emily	5
6	Anderson	Stewart	20
7	Aaron	Kim	22
8	Cathenna	Emily	4
9	Roger	Walson	5
10	Catherine	Shad	13
11	Leo	Travis	15
12	Roger	Walson	4
13	Melvin	Tracy	10
14	Linda	William	14
15	Solomon	Walter	13
16	Pheny	Eri	22
17	Calvin	Willis	14
18	Moss	Morris	23
19	Rose	Arthur	21
20	Watson	Ronald	9
21	Bily	Brian	15
22	Louis	Douglas	8
23	Russell	Peter	15
24	James	Robert	20
25	Gloria	Richie	1
26	Louis	Douglas	25

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

#	no_of_passenger	total_revenu
1	11	6034

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

#	first_name	last_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

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

#	first_name	last_name	customer_id
1	Julie	Sam	1
2	Steve	Ryan	2
3	Cathenna	Emily	4
4	Aaron	Kim	5
5	Anderson	Stewart	7
6	Floyd	Ted	8
7	Leo	Travis	9
8	Melvin	Tracy	10
9	Roger	Walson	11
10	Solomon	Walter	13
11	Carol	Vernon	14
12	Linda	William	15
13	Chirstine	Willis	16
14	Catherine	Shad	17
15	Gloria	Richie	18
16	Joyce	Paul	19
17	Sara	Oliver	20
18	Chirsty	Josh	21
19	Pheny	Eri	22
20	Calvin	Willis	24
21	Moss	Morris	25
22	Cherly	Vernon	27
23	Du plesis	Chris	28
24	Watson	Ronald	29
25	James	Robert	31
26	Chirstoper	Sean	32

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.

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

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.

#	first_name	last_name	customer_id
1	Julie	Sam	1
2	Floyd	Ted	8
3	Roger	Walson	11
4	Catherine	Shad	17
5	Joyce	Paul	19
6	Pheny	Eri	22
7	Chirstoper	Sean	32
8	Sophia	Carl	47
9	Rose	Arthur	50

Write a query to identify whether the revenue has crossed 10000 using the IF clause on the ticket_details table.

#	rev_crossed_1000	revenue
1	Yes	15369

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

```
-- Query to create a new user
create user 'new_user'@'localhost' identified by 'password';
grant select, insert, update, delete on air_cargo.* to 'new_user'@'localhost';
```

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

#	class_id	max_price_per_clas
1	Bussiness	510
2	Economy	190
3	Economy Plus	295
4	First Class	395

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

#	customer_i	aircraft_id	route_id	depart	arriva	seat_num	class_id	travel_dat	flight_nun
1	2	767-301ER	4	JFK	LAX	01E	Economy	02-09-2018	1114
2	4	767-301ER	4	JFK	LAX	03FC	First Class	30-04-2020	1114
3	11	767-301ER	4	JFK	LAX	05B	Bussiness	09-11-2020	1114

For the route ID 4, write a query to view the execution plan of the passengers_on_flights table.

#	id	select_typ	table	partition	type	possible_key	key	key_len	ref	rows	filtered	Extra
1	1	SIMPLE	passengers_on_flights	NULL	ref	idx_route_id	idx_route_id	5	const	3	100.00	NULL

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

#	customer_id	aircraft_id	total_price
1	1	CRJ900	320
2	1	ERJ142	250
3	1	NULL	570
4	2	767-301ER	130
5	2	A321	505
6	2	NULL	635
7	4	767-301ER	780
8	4	NULL	780
9	5	767-301ER	430
10	5	ERJ142	240
11	5	NULL	670
12	7	767-301ER	430
13	7	NULL	430
14	8	A321	465
15	8	NULL	465
16	9	767-301ER	380
17	9	CRJ900	390
18	9	NULL	770
19	10	A321	135
20	10	NULL	135
21	11	767-301ER	930
22	11	ERJ142	295
23	11	NULL	1225
24	13	A321	395
25	13	NULL	395
26	14	767-301ER	170

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

#	first_name	last_name	brand
1	Chirsty	Josh	Bristish Airways
2	Anderson	Stewart	Emirates
3	Roger	Walson	Emirates
4	Moss	Morris	Emirates
5	Calvin	Willis	Qatar Airways
6	Watson	Ronald	Qatar Airways
7	Steve	Ryan	Qatar Airways
8	Watson	Ronald	Jet Airways
9	Aaron	Kim	Emirates
10	Linda	William	Qatar Airways
11	Mark	Ethan	Bristish Airways
12	Russell	Peter	Emirates
13	Roger	Walson	Emirates

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.

```
call range_of_routes(20,50);
```

#	customer_id	aircraft_id	route_id	depart	arrival	seat_num	class_id	travel_date	flight_num
1	2	A321	34	CRW	COD	01B	Bussiness	26-01-2019	1117
2	1	CRJ900	30	BUR	STT	01FC	First Class	04-11-2018	1140
3	8	A321	38	CST	DAL	02EP	Economy Plus	09-08-2020	1148
4	7	767-301ER	20	AVL	BOI	03B	Bussiness	08-07-2020	1130
5	5	ERJ142	22	BGR	BJI	03E	Economy	31-05-2020	1132
6	11	ERJ142	31	BTM	CHA	03EP	Economy Plus	02-08-2018	1141
7	8	A321	43	CBM	BOI	04E	Economy	02-05-2018	1153
8	19	CRJ900	47	DAL	LAX	05EP	Economy Plus	13-01-2021	1157
9	9	CRJ900	33	CDC	CST	05FC	First Class	01-02-2018	1143
10	14	ERJ142	35	STT	CDB	06E	Economy	02-04-2019	1145
11	19	CRJ900	30	BUR	STT	06EP	Economy Plus	17-12-2020	1140
12	21	CRJ900	45	CCR	EWR	07B	Bussiness	07-03-2020	1155
13	14	767-301ER	42	CSG	BOS	07E	Economy	25-01-2020	1152
14	22	ERJ142	22	BGR	BJI	07EP	Economy Plus	09-02-2020	1132
15	16	CRJ900	39	COD	SCC	07FC	First Class	04-05-2019	1149
16	18	767-301ER	46	CDV	HNL	08E	Economy	07-07-2019	1156
17	32	ERJ142	31	BTM	CHA	08EP	Economy Plus	04-03-2021	1141
18	20	CRJ900	36	CHA	COU	08FC	First Class	26-09-2019	1146
19	25	767-301ER	23	BLV	BFL	09B	Bussiness	07-03-2019	1133
20	19	767-301ER	32	CLD	CHI	09E	Economy	07-02-2018	1142
21	47	CRJ900	33	CDC	CST	09EP	Economy Plus	15-12-2020	1143
22	20	CRJ900	39	COD	SCC	09FC	First Class	03-05-2020	1149
23	29	A321	38	CST	DAL	10B	Bussiness	21-11-2019	1148
24	25	767-301ER	44	COU	CAK	10E	Economy	21-10-2019	1154
25	50	A321	21	BFL	BET	10EP	Economy Plus	15-08-2020	1131
26	41	A321	34	CRW	COD	10FC	First Class	15-02-2019	1144

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.

#	route_id	flight_num	origin_airpor	destination_airpo	aircraft_id	distance_mile
1	1	1111	EWR	HNL	767-301ER	4962
2	2	1112	HNL	EWR	767-301ER	4962
3	3	1113	EWR	LHR	A321	3466
4	4	1114	JFK	LAX	767-301ER	2475
5	5	1115	LAX	JFK	767-301ER	2475
6	6	1116	HNL	LAX	767-301ER	2556
7	10	1120	HNL	DEN	A321	3365
8	12	1122	ABI	ADK	767-301ER	4300
9	13	1123	ADK	BQN	A321	2232
10	14	1124	BQN	CAK	A321	2445
11	18	1128	ANI	BGR	ERJ142	2450
12	19	1129	ATW	AVL	A321	2222
13	20	1130	AVL	BOI	767-301ER	3134
14	21	1131	BFL	BET	A321	2425
15	23	1133	BLV	BFL	767-301ER	2354
16	25	1135	RDM	BJI	A321	2425
17	34	1144	CRW	COD	A321	2452
18	35	1145	STT	CDB	ERJ142	2121
19	43	1153	CBM	BOI	A321	8989
20	44	1154	COU	CAK	767-301ER	7676
21	46	1156	CDV	HNL	767-301ER	8668
22	48	1158	SCC	DEN	A321	5645
23	49	1159	DEC	ABI	A321	4533
24	50	1160	DRT	ORD	A321	2445

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.

#	distance_mile	distance_category
1	4962	IDT
2	4962	IDT
3	3466	IDT
4	2475	IDT
5	2475	IDT
6	2556	IDT
7	1745	SDT
8	719	SDT
9	862	SDT
10	3365	IDT
11	4300	IDT
12	2232	IDT
13	2445	IDT
14	2000	SDT
15	1700	SDT
16	1900	SDT
17	2450	IDT
18	2222	IDT
19	3134	IDT
20	2425	IDT
21	1242	SDT
22	2354	IDT
23	1575	SDT
24	2425	IDT
25	1311	SDT
26	578	SDT

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*

#	p_date	customer_id	class_id	comp_servic
1	26-12-2018	27	Economy	No
2	02-02-2020	22	Economy Plus	Yes
3	03-03-2020	21	Bussiness	Yes
4	04-04-2020	4	First Class	No
5	05-05-2020	5	Economy	No
6	07-07-2020	7	Bussiness	Yes
7	08-08-2020	8	Economy Plus	Yes
8	09-09-2020	9	First Class	No
9	10-10-2020	10	Economy	No
10	11-11-2020	11	Bussiness	Yes
11	12-12-2020	19	Economy Plus	Yes
12	01-01-2019	13	First Class	No
13	02-02-2019	14	Economy	No
14	03-03-2019	25	Bussiness	Yes
15	04-04-2019	16	First Class	No
16	03-05-2019	17	Economy Plus	Yes
17	06-06-2019	18	Economy	No
18	07-07-2019	24	Bussiness	Yes
19	09-08-2019	20	First Class	No
20	21-09-2019	25	Economy	No
21	22-10-2019	29	Bussiness	Yes
22	23-11-2019	1	Economy Plus	Yes
23	24-12-2019	14	Economy	No
24	25-01-2019	2	Bussiness	Yes
25	01-01-2018	9	First Class	No
26	01-02-2018	19	Economy	No

Write a query to extract the first record of the customer whose last name ends with Scott using a cursor from the customer table.

#	customer_i	first_name	last_name	date_of_birt	gender
1	37	Samuel	Scott	28-01-2000	M
*	NULL	NULL	NULL	NULL	NULL