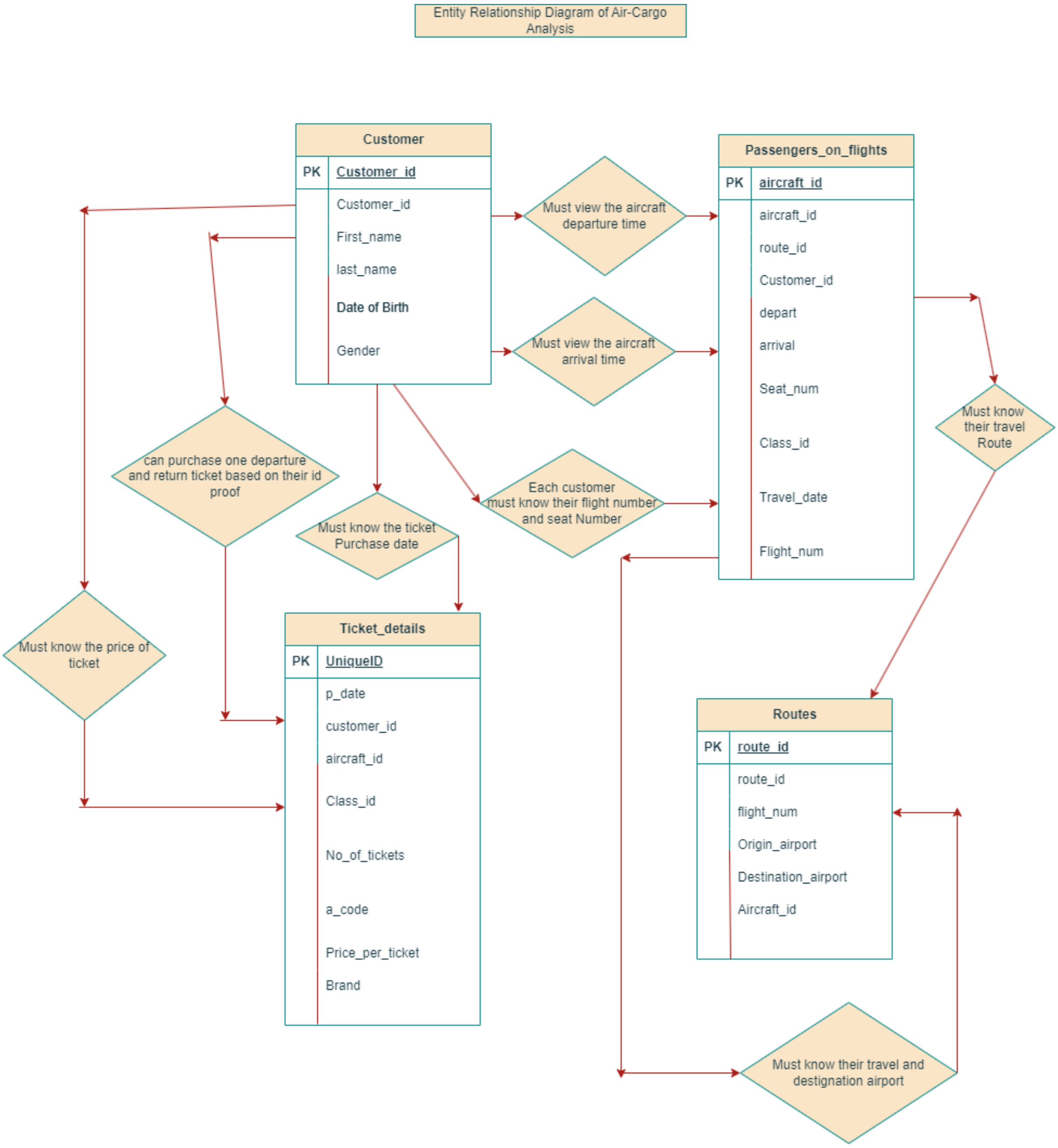
Air-Cargo Analysis

Q1) Create an E-R Diagram for the given database airline

Ans) The E-R Diagram for the given database airline is mentioned below:



Entity Relationship Diagram of Air-Cargo Analysis
Customer
PK
Customer_id
Customer_id
First_name
last_name
Date of Birth
Gender
Must view the aircraft departure time
Must view the aircraft arrival time
Passengers_on_flights
PK
aircraft_id
aircraft_id
route_id
Customer_id
depart
arrival
Seat_num
Class_id
Travel_date
Flight_num
Must know their travel Route
can purchase one departure and return ticket based on their id proof
Must know the ticket Purchase date
Each customer must know their flight number and seat Number
Must know the price of ticket
Ticket_details
PK
UniqueID
p_date
customer_id
aircraft_id
Class_id
No_of_tickets
a_code
Price_per_ticket
Brand
Routes
PK
route_id
route_id
flight_num
Origin_airport
Destination_airport
Aircraft_id
Must know their travel and destignation airport

Q2)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.

Ans)create database airlines;

```
use airlines;

create table route_details(route_id int unique,flight_num bigint
check(flight_num>=1111),origin_airport varchar(100),destination_airport

varchar(100),aircraft_id varchar(90),distance_miles bigint check(distance_miles>0));

desc route_details;
```

Q3)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.

Ans)use airlines;

```
select * from passengers_on_flights where route_id between 1 and 25;
```

Q4)Write a query to identify the number of passengers and

total revenue in business class from the ticket_details table.

Ans)use airlines;

```
select count(customer_id) as number_of_passengers,sum(price_per_ticket) as Total_Revenue

from ticket_details where class_id="Bussiness";
```

Q5)Write a query to display the full name of the customer

by extracting the first name and last name from the customer table.

Ans)use airlines;

```
select concat(first_name,last_name) as FullName from customer;
```

Q6)Write a query to extract the customers who have registered and booked a ticket.

Use data from the customer and ticket_details tables.

Ans)use airlines;

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

Q7)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.

Ans)use airlines;

```
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";
```

Q8)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.

Ans)use airlines;

```
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;
```

Q9)Write a query to identify whether the revenue has crossed 10000 using the IF clause on the ticket_details table.

Ans)use airlines;

```
select * from ticket_details;
select sum(price_per_ticket) as total_revenue from ticket_details;
select if(15639>10000,"Revenue has crossed 10000","Revenue has not crossed 10000");
```

Q10)Write a query to create and grant access to a new user to perform operations on a database.

Ans)use airlines;

```
create user "shivani";
grant select on customer to shivani;
grant select on passengers_on_flights to shivani;
grant select on route_details to shivani;
grant select on ticket_details to shivani;
```

Q11)Write a query to find the maximum ticket price for each class using window functions on the ticket_details table.

```
Ans)use airlines;
select class_id,max(Price_per_ticket) over (Partition  by class_id)
 as Max_ticketpriceforeach_class from ticket_details;
select distinct(class_id),max(Price_per_ticket) over (Partition  by class_id)
 as Max_ticketpriceforeach_class from ticket_details;
```

Q12)Write a query to extract the passengers whose route ID is 4 by improving the speed and performance of the passengers_on_flights table.

```
Ans)use airlines;
select customer_id from passengers_on_flights where route_id=4;
select first_name,last_name from customer where customer_id in(2,4,11);
```

Q13)For the route ID 4,write a query to view the execution plan of the passengers_on_flights table.

```
Ans)use airlines;
select * from passengers_on_flights;
create view myflights
as select * from passengers_on_flights where route_id=4;
select * from myflights;
```

Q14)Write a query to calculate the total price of all tickets booked by a customer across different aircraft IDs using rollup function.

```
Ans)use airlines;
select aircraft_id,customer_id,sum(price_per_ticket) as TotalPrice
from ticket_details group by aircraft_id,customer_id with rollup;
```

Q15)Write a query to create a view with only business class customers along with the brand of airlines.

```
Ans)use airlines;
select * from ticket_details;
create view airlinesbrand as
select customer_id,class_id,brand,price_per_ticket
from ticket_details where class_id="Bussiness";
select * from airlinesbrand;
```

Q16)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.

```
Ans)use airlines;
Delimiter $$
drop procedure passengerdetails;
create procedure passengerdetails()

begin
select customer.first_name,customer.last_name,passengers_on_flights.route_id
from customer left join passengers_on_flights on
customer.customer_id=passengers_on_flights.customer_id;
end;

call passengerdetails();
```

Q17)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.

```
Ans)use airlines;
select * from routes;
Delimiter $$
create procedure myroutes()
begin
select * from routes where distance_miles>2000;
end $$

call myroutes();
```

Q18)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.

```
Ans)use airlines;
drop procedure if exists flightanalysis;
delimiter $$
create procedure flightanalysis()
begin
select * from routes;
select flight_num,distance_miles as shortdistance from routes where distance_miles between 0 and 2000;
select flight_num,distance_miles as intermediatedistance from routes where distance_miles between 2000 and 6500;
select flight_num,distance_miles as LongDistance from routes where distance_miles >6500;

end $$
```

Q19)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.If the class is Business and Economy Plus, then complimentary services are given as Yes,

else it is No

```
Ans)use airlines;
drop procedure if exists ticketanalysis;

Delimiter $$
create procedure ticketanalysis()
begin
select p_date,customer_id,class_id from ticket_details;
select class_id as firstclass from ticket_details where class_id="Bussiness";
alter table ticket_details add column Compserv varchar(100);
update ticket_details set compserv="Yes" where class_id="Bussiness";
select distinct(class_id),compserv from ticket_details where class_id="Bussiness";
update ticket_details set compserv="Yes" where class_id="Economy Plus";
select distinct(class_id),compserv from ticket_details where class_id="Economy Plus";
update ticket_details set compserv="No" where class_id="Economy";
select distinct(class_id),compserv from ticket_details where class_id="Economy";
update ticket_details set compserv="No" where class_id="First Class";
select distinct(class_id),compserv from ticket_details where class_id="First Class";


end $$
```

Q20) Write a query to extract the first record of the customer whose last name ends with Scott

using a cursor from the customer table

```
Ans)use airlines;
select * from customer where last_name like '%Scott';
drop procedure if exists finalnames;

delimiter $$

create procedure finalnames()
begin

Declare c1 cursor for select last_name from customer where last_name like '%Scott';
open c1;
select * from customer where last_name like '%Scott';
end $$

call finalnames();
```

For detailed description of SQL code click on the below mentioned link

https://github.com/shivanipriya89/AirCargo-Analysis