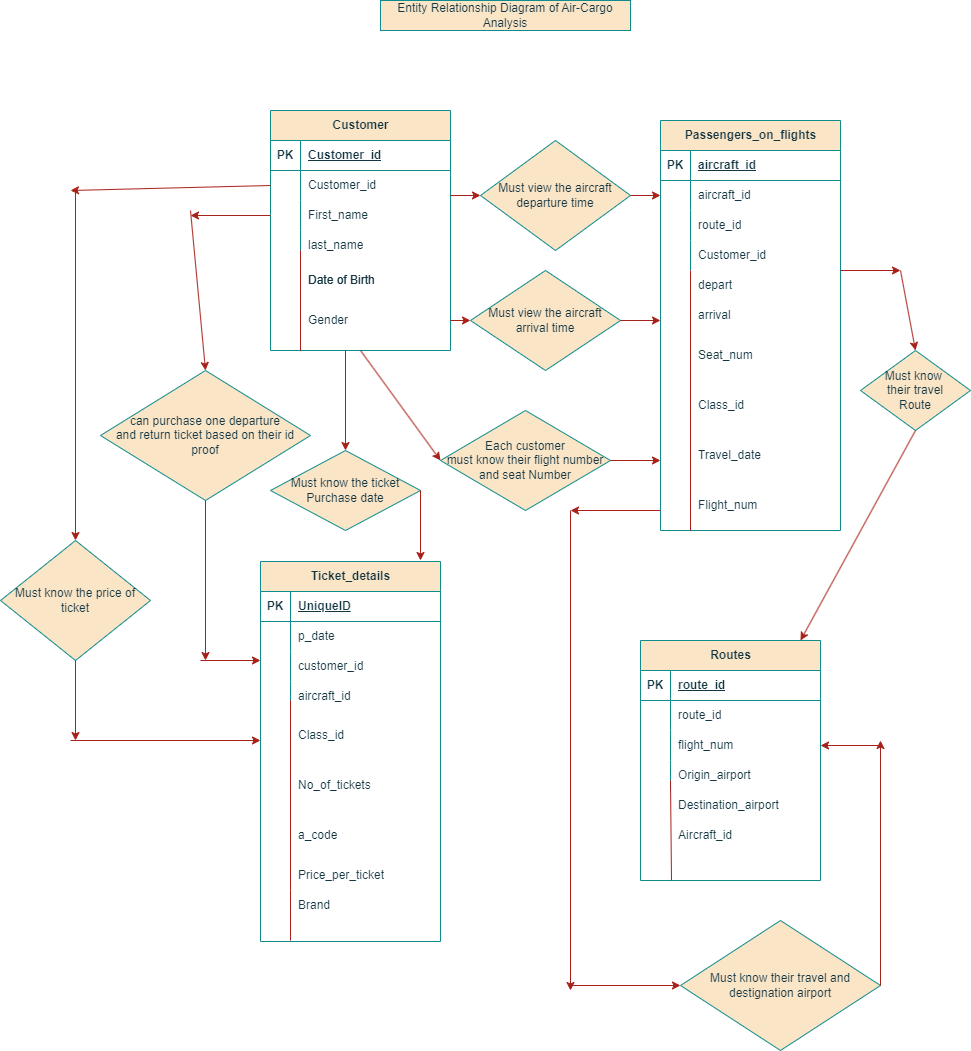
**Air-Cargo Analysis**

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

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



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>