Air Cargo Analysis Source code

```
create database  Air_Cargo_Analysis;
use Air_Cargo_Analysis;
```

#1.Create an ER diagram for the given airlines database.

```
# ER diagram
```

/*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 unique constraint for the route_id fields. Also, make sure that the distance miles field is greater than 0.*/

```
create table routes (
route_id int not null unique,
flight_num varchar(40)  not null,
origin_airport varchar(40) not null,
destination_airport varchar(40) not null,
aircraft_id varchar(40) not null,
 distance_miles int not null);

alter table routes add Unique(route_id);

select*from Customer;

select*from passengers_on_flights;

select*from ticket_details;

select*from routes;
```

/*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.*/

```
select c.first_name,concat(first_name,' ',last_name) as Full_Name FROM passengers_on_flights p
left join customer c on(C.customer_id=P.customer_id)
where route_id between 1 and 25;
```

/*4.Write a query to identify the number of passengers and total revenue in business class from the ticket_details table.*/

```
select count(*) as number_of_customer,sum(price_per_ticket) from ticket_details
where class_id='Bussiness';
```

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

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

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

```
select distinct(c.customer_id), concat(first_name,' ',last_name) as Full_Name
from customer c
inner join ticket_details td on
c.customer_id=td.customer_id;
```

/*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.*/

```
select c.customer_id, first_name,last_name,concat(first_name,' ',last_name) as Full_Name
from customer c
inner join ticket_details td on
c.customer_id=td.customer_id and brand='Emirates';
```

/*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.*/

```
select count(customer_id) as Total_Customers from passengers_on_flights
group by class_id
having class_id='Economy plus';
```

/*9.Write a query to identify whether the revenue has crossed 10000 using the IF clause on the ticket_details table.*/

```
select sum(price_per_ticket) as Revenue, if(sum(price_per_ticket)>10000,"Yes Revenue has
Crossed 10000",
"no Revenue has Crossed not 10000") as Total_Revenue from ticket_details;
```

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

```
grant all on *.* to'root'@'localhost';
```

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

```
select customer_id,  class_id , max(Price_per_ticket) over(partition by class_id) from ticket_details;
```

#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.

```
select customer_id from passengers_on_flights where route_id=4;
```

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

```
select * from passengers_on_flights where route_id=4;
```

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

```
select customer_id,aircraft_id,SUM(Price_per_ticket) as Total_sales from ticket_details
group by customer_id, aircraft_id with rollup;
```

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

```
create view Bussiness_class as select customer_id,brand from ticket_details where class_id='bussiness';
select*from Bussiness_class;
```

/*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.*/

```
delimiter &&
create procedure passengers_flying()
begin
select * from  routes;
END &&
call passengers_flying() ;
```

#Also, return an error message if the table doesn't exist

```
delimiter &&
create procedure passengers_flying()
begin
select * from  route;
END &&
call passengers_flying() ;
```

/*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.*/

```
delimiter &&
create procedure distance_miles()
begin
select * from  routes where distance_miles > 2000;
END &&
call distance_miles() ;
```

/*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 >=0 AND <= 2000 miles, intermediate distance travel (IDT) for >2000 AND <=6500, and long-distance travel (LDT) for >6500.*/

```
select *,case
when distance_miles >=0 and distance_miles <= 2000 then "short distance travel (SDT)"
when distance_miles >2000 and distance_miles <= 6500 then "intermediate distance travel
(IDT)"
when  distance_miles >6500 then "long-distance travel (LDT)"
end as categories from routes;
```

/*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*/

```
select p_date, customer_id, class_id,
case
        when class_id = 'Bussiness' or class_id = "economy plus" then 'Yes'
   else 'No'
end as Complimentary_service from ticket_details
order by customer_id;
```

#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.

```
select*from Customer where last_name='Scott' limit 1 ;
```