```
create database aircargo_project;
show databases;
show tables;

/*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 */

DROP TABLE route;
CREATE TABLE
route_details(route_id INT(10) UNIQUE ,flight_num INT(10) CHECK(flight_num
>1000),origin_airport VARCHAR(225),destination_airport VARCHAR(225),aircraft_id
VARCHAR(225),distance_miles INT(10) CHECK(distance_miles>0) );

select * from
route_details;

/* 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 FROM
passengers_on_flights P LEFT JOIN customer C ON(C.customer_id=P.customer_id) WHERE `route_id`
BETWEEN 1 AND 25;

/* Query to identity the number of passengers and total revenue in business
class from the ticket_details table*/

SELECT COUNT(customer_id)AS No_of_Customers,
SUM(`Price_per_ticket`)AS Total_Price FROM `ticket_details`
WHERE`class_id`='Bussiness';

/*Query to display the Full name ot the customer by extracting
the first name and last name trom the customer table*/

SELECT CONCAT(first_name,last_name) AS
Full_name FROM customer

/*Query to extract the customers who have registered ana book a
ticket. use data trom the customer and ticket_details tables*/

SELECT DISTINCT(C.customer_id)
FROM ticket_details T LEFT JOIN customer C ON (C.customer_id = T.customer_id) WHERE
T.customer_id IS NOT NULL;

/*Query to identity the customer's first name and last name based
on their customer ID and  brand (Emirates) from the ticket_details table*/

SELECT
CONCAT(C.first_name,C.last_name) AS Full_name FROM customer C LEFT JOIN ticket_details T
ON(C.customer_id = T.customer_id) WHERE T.brand='Emirates' ORDER BY
C.customer_id,T.brand;

/*Query to identity the customers who have travelled by Economy Plus
class using Group By and Having clause on the passenger_on_flight  table*/

SELECT
COUNT(customer_id) AS Total_Customers FROM passengers_on_flights_csv GROUP BY class_id HAVING
class_id="Economy Plus";

/*Query to identity whether the revenue has crossed  10000
using the IF clause on the ticket_details table*/

SELECT
IF(SUM(Price_per_ticket)>10000,"Yes Revenue has Crossed 10000", "no Revenue
has Crossed not 10000") AS Total_Revenue FROM `ticket_details`

/*Query to create and
grant access to a new user to perform operations on a database*/

USE 'aircargo_project'
```

```
GRANT
ALL ON *.* TO 'root'@'localhost';

/*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;

/*Query to extract the
passengers whose route ID is 4 by improving the speed and performance of the
passenger_on_flight table*/

SELECT customer_id FROM `passengers_on_flights` WHERE
route_id=4;

/*For the route ID 4, write a query to view the execution plan of the
passengers_on_flight table*/

SELECT * FROM `passengers_on_flights` WHERE route_id=4;

/*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;

/*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;

/*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 get_total_passengers_()
BEGIN
DECLARE totalpassengers INT DEFAULT 0;
SELECT
COUNT(*)
INTO totalpassengers
FROM passengers_on_flights;
SELECT totalpassengers;
END
&&
DELIMITER ;
SHOW PROCEDURE STATUS;

/*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();

/*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.*/
```

```
delimiter //
create function group_dist(dist int)
returns
varchar(10)
deterministic
begin
  declare dist_cat char(3);
  if dist between 0 and 2000 then

   set dist_cat ='SDT';
  elseif dist between 2001 and 6500 then
    set dist_cat ='IDT';

elseif dist > 6500 then
   set dist_cat ='LDT';
 end if;
 return(dist_cat);
end //

/*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;

/*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';
```