```sql
-- Creating Database aircargo and querying to start using it
create database if not exists aircargo;
use aircargo;

-- Creating 4 tables for database (customer, pof, routes, ticket_details)
and inserting data using import wizard.
drop table if exists customer;
CREATE TABLE if not exists customer (
  customer_id int,
  first_name varchar(100) NOT NULL,
  last_name varchar(100) DEFAULT NULL,
  date_of_birth date NOT NULL,
  gender varchar(1) NOT NULL,
  PRIMARY KEY (customer_id),
  CONSTRAINT Gender_check CHECK ((gender in ('M','F','O')))
);

describe customer;

CREATE TABLE routes (
  route_id int NOT NULL,
  flight_num int NOT NULL,
  origin_airport varchar(3) NOT NULL,
  destination_airport varchar(100) NOT NULL,
  aircraft_id varchar(100) NOT NULL,
  distance_miles int NOT NULL,
  PRIMARY KEY (route_id),
  CONSTRAINT Flight_number_check CHECK ((substr(flight_num,1,2) = 11)),
  CONSTRAINT routes_chk_1 CHECK ((distance_miles > 0))
);
CREATE TABLE ticket_details (
  p_date date NOT NULL,
  customer_id int NOT NULL,
  aircraft_id varchar(100) NOT NULL,
  class_id varchar(100) DEFAULT NULL,
  no_of_tickets int DEFAULT NULL,
  a_code varchar(3) DEFAULT NULL,
  Price_per_ticket int DEFAULT NULL,
  brand varchar(100) DEFAULT NULL,
  KEY customer_id (customer_id),
  CONSTRAINT ticket_details_ibfk_1 FOREIGN KEY (customer_id) REFERENCES
customer (customer_id)
);

-- Finding 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 Name from customer
order by name;
-- Querying data of customers who have booked at least a ticket and total
tickets booked by them.
select c.customer_id , concat(c.first_name, ' ' , c.last_name) as Name,
count(t.no_of_tickets) as Total_Tickets_booked
from customer c
join ticket_details t using (customer_id)
```

```sql
group by c.customer_id, Name
order by Total_tickets_booked desc;

drop  procedure if exists revenue;
delimiter //
create procedure revenue ( in target int, out Revenue varchar(100))
begin
declare y int;
select sum(no_of_tickets*price_per_ticket) into y from ticket_details;
if y > target
then set Revenue = concat('Revenue Crossed ', ' ' , target);
else set Revenue = concat('Revenue less than', ' ' , target);
end if;
end //
delimiter ;
call revenue(15000, @Rev);
select @Rev as Revenue_Status;
use aircargo;


-- Fetching max ticket price for each class
with cte as (
select class_id, max(price_per_ticket) as Maximum_price,
dense_rank () over (partition by class_id) as dense
from ticket_details
group by class_id)
select class_id, Maximum_price from cte where dense = 1;

select brand, class_id, price_per_ticket, max(price_per_ticket)
over(partition by class_id ) from ticket_details
order by 2;

-- Fetching total revenue generated for each aircraft
select if (grouping (aircraft_id), 'Total',aircraft_id) as Aircraft,
sum(no_of_tickets) as Total_tickets,
sum(no_of_tickets*price_per_ticket) as Total_Revenue
from ticket_details
group by aircraft_id with rollup
order by 3;

-- Create view for business class customers with brand of airline
drop view if exists business_class;
create view  business_class as select c.first_name, c.last_name, t.brand
from customer c
join ticket_details t using (customer_id)
where class_id in ('Business');

select brand from business_class order by 1;

-- Fetching customer details using procedure where distance travelled is
more than 2000

drop procedure if exists distance;
delimiter //
```

```
create procedure distance( in miles int)
begin
select * from routes
where distance_miles >miles
order by distance_miles;
end//
delimiter ;

call distance(2000);

-- Creating a procedure to analyze if the distance travelled by a aircraft
on particular route is Short distance, intermediate distance
-- or long distance travel

show procedure status where db = 'aircargo';
drop procedure if exists Distance_info;

delimiter //
create procedure distance_info(in route int, out info varchar(200))
begin
declare x int;
select distance_miles into x from routes where route_id = route;
if x < 2000 then set info = 'Short Distance Travel';
elseif x between 2000 and 6500 then set info = 'Intermediate Distance
travel';
elseif x >=6500 then set info = 'Long Distance Travel';
end if;
end//
delimiter ;

use aircargo;
```