```
show databases;

create database air cargo;

show databases


create table if not exists customer(

          customer_id int not null auto_increment primary key,

   first_name varchar(20) not null,

   last_name varchar(20) not null,

   date_of_birth date not null,

   gender char(1) not null

   );


describe customer;


select * from customer;


create table if not exists routes(

                    route_id int not null unique primary key,

      flight_num int constraint chk_1 check (flight_num is not null),

      origin_airport char(3) not null,

      destination_airport char(3) not null,

      aircraft_id varchar(10) not null,

      distance_miles int not null constraint check_2 check (distance_miles > 0) );


      select * from routes;

      create  table if not exists pof(

   pof_id int auto_increment primary key,

   customer_id int not null,

   aircraft_id varchar(10) not null,

   route_id int not null,

   depart char(3) not null,
   arrival char(3) not null,
   seat_num char(4) not null,
   class_id varchar(15) not null,
   travel_date date not null,
   flight_num int not null,
   constraint fk_pof foreign key (customer_id) references customer(customer_id)
   );
   select * from POF;

   create table if not exists ticket_details(
           tkt_id int auto_increment primary key,
   p_date date not null,
   customer_id int not null,
   aircraft_id varchar(10) not null,
   class_id varchar(15) not null,
   no_of_tkts int not null,
   a_code char(3) not null,
   price_per_tkt decimal(5,2) not null,
   brand varchar(30) not null,
   constraint fk_tkt_dts foreign key (customer_id) references customer(customer_id)
   );

   describe ticket_details;
select * from ticket_details;

select * from customer
   where customer_id in (select distinct customer_id from pof where route_id between 1 and 25)
   order by customer_id;

   select * from customer
   where customer_id in (select distinct customer_id from pof where route_id between 1 and 25)
   order by customer_id;
```

```
SELECT CONCAT(first_name," ",last_name ) as full_name from customer;
```

```
SELECT CONCAT(first_name," ",last_name ) as full_name from customer;
```

```
select first_name, Last_name  from customer
where customer_id in (select distinct b.customer_id from customer a, ticket_details b);
```

```
select first_name, Last_name  from customer
where customer_id in (select distinct b.customer_id from customer a, ticket_details b);
```

```
select first_name, Last_name  from customer
where customer_id in (select distinct b.customer_id from customer a, ticket_details b);
```

```
select * from customer a
 inner join (select distinct customer_id from pof where class_id = 'Economy Plus') b
 on a.customer_id = b.customer_id;
```

```
  select if ((select sum(no_of_tkts * price_per_tkt) as total_revenue from ticket_details)> 10000,
'Crossed 10K', 'Not Crossed 10K') as revenue_check;
```

```
  create user if not exists 'pavan'@'127.0.0.1' identified by 'Password123';
 grant all privileges on aircargo to pavan@127.0.0.1 ;
```

```
  select class_id, max(price_per_tkt)
 from ticket_details
 group by class_id
```

```
  select class_id, max(price_per_tkt)
 from ticket_details
 group by class_id
```

```
 create index idx_rid on pof (route_id);
 explain select * from pof where route_id = 4;
```

```
  explain select * from pof where route_id = 4;
```

```
SELECT customer_id, aircraft_id,
    SUM(price_per_tkt * no_of_tkts) AS total_price
FROM ticket_details
GROUP BY customer_id , aircraft_id
ORDER BY customer_id , aircraft_id;
```

```
SELECT customer_id, aircraft_id,
    SUM(price_per_tkt * no_of_tkts) AS total_price
FROM ticket_details
GROUP BY customer_id , aircraft_id
WITH ROLLUP ORDER BY customer_id , aircraft_id;
```

```
 create view buss_class_customers as
 select a.*, b.brand from customer a
 inner join (select distinct customer_id, brand from ticket_details where class_id = 'Bussiness' order by customer_id) b
 on a.customer_id = b.customer_id;


 create view buss_class_customers as

 select a.*, b.brand from customer a

 inner join (select distinct customer_id, brand from ticket_details where class_id = 'Bussiness' order by customer_id) b

 on a.customer_id = b.customer_id;


 select * from buss_class_customers;


  select * from customer where customer_id in (select distinct customer_id from pof where route_id in (1,5));


 delimiter //

 create procedure check_route (in rid varchar(255))

 begin

            declare TableNotFound condition for 1146;

   declare exit handler for TableNotFound

                         select 'Please check if table customer/route id are created - onr/both are missing' Message;

       set @query = concat('select * from customer where customer_id in ( select distinct customer_id from pof where route_id in (',rid,'));');

       prepare sql_query from@query;

       execute sql_query;

       end//

       delimiter ;

       call check_route("1,5");


       delimiter //

create procedure check_dist()

begin

            select * from routes where distance_miles > 2000;
   end //
   delimiter ;

   call check_dist;



select flight_num, distance_miles, case
                        when distance_miles between 0 and 2000 then "SDT"
      when distance_miles between 2001 and 6500 then "IDT"
      else "LDT"
      end distance_category from routes;

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 //

create procedure group_dist_proc()
begin
```

```
            select flight_num, distance_miles, group_dist(distance_miles) as distance_category from
routes;
   end //
   delimiter ;

call group_dist_proc();

select p_date, customer_id, class_id, case
                                                                                          when class_id in
('Bussiness','Economy Plus') then "Yes"
                                                                                          else "No"
                                                                                          end as
complimentary_service from ticket_details;

       DELIMITER //
CREATE FUNCTION check_comp_serv(cls VARCHAR(15))
RETURNS CHAR(3)
DETERMINISTIC
BEGIN
   DECLARE comp_ser CHAR(3);
   IF cls IN ('Business', 'Economy Plus') THEN
      SET comp_ser = 'Yes';
   ELSE
      SET comp_ser = 'No';
   END IF;
   RETURN comp_ser;
END //

CREATE PROCEDURE check_comp_serv_proc()
BEGIN
   SELECT p_date, customer_id, class_id, check_comp_serv(class_id) AS complimentary_service
   FROM ticket_details;
```

```
END;
CALL check_comp_serv_proc();



select * from customer where last_name = 'Scott' limit 1;


delimiter //
create procedure cust_lname_Scott()
begin
        declare c_id int;
   declare f_name varchar (20);
   declare l_name varchar(20);
   declare dob date;
   declare gen char(1);
   declare cust_rec cursor
   for
   select * from customer where last_name = "Scott";

   create table if not exists cursor_table(
                c_id int,
      f_name varchar(20),
      l_name varchar(20),
      dob date,
      gen char (1) );

   open cust_rec;
   fetch cust_rec into c_id, f_name, l_name, dob, gen;
   insert into cursor_table(c_id, f_name, l_name, dob, gen) values (c_id, f_name, l_name, dob, gen);
   close cust_rec;

   select * from cursor_table;
```

```
end //

delimiter ;

call cust_lname_scott();
```