create database if not exists `travelonthego`;

use `travelonthego`;

create table if not exists `passenger`(

Passenger\_name varchar(50),

Category varchar(20),

Gender varchar(5),

Boarding\_City varchar(50),

Destination\_City varchar(50),

Distance int,

Bus\_Type varchar(50)

);

create table if not exists `price`(

Bus\_Type varchar(50),

Distance int,

Price int

);

insert into `passenger` values ("Sejal", "AC", "F", "Bengaluru", "Chennai", 350, "Sleeper");

insert into `passenger` values ("Anmol", "Non-AC", "M", "Mumbai", "Hyderabad", 700, "Sitting");

insert into `passenger` values ("Pallavi", "AC", "F", "Panaji", "Bengaluru", 600, "Sleeper");

insert into `passenger` values ("Khusboo", "AC", "F", "Chennai", "Mumbai", 1500, "Sleeper");

insert into `passenger` values ("Udit", "Non-AC", "M", "Trivandrum", "panaji", 1000, "Sleeper");

insert into `passenger` values ("Ankur", "AC", "M", "Nagpur", "Hyderabad", 500, "Sitting");

insert into `passenger` values ("Hemant", "Non-AC", "M", "panaji", "Mumbai", 700, "Sleeper");

insert into `passenger` values ("Manish", "Non-AC", "M", "Hyderabad", "Bengaluru", 500, "Sitting");

insert into `passenger` values ("Piyush", "AC", "M", "Pune", "Nagpur", 700, "Sitting");

insert into `price` values ("Sleeper", 350, 770);

insert into `price` values ("Sleeper", 500, 1100);

insert into `price` values ("Sleeper", 600, 1320);

insert into `price` values ("Sleeper", 700, 1540);

insert into `price` values ("Sleeper", 1000, 2200);

insert into `price` values ("Sleeper", 1200, 2640);

insert into `price` values ("Sleeper", 350, 434);

insert into `price` values ("Sitting", 500, 620);

insert into `price` values ("Sitting", 500, 620);

insert into `price` values ("Sitting", 600, 744);

insert into `price` values ("Sitting", 700, 868);

insert into `price` values ("Sitting", 1000, 1240);

insert into `price` values ("Sitting", 1200, 1488);

insert into `price` values ("Sitting", 1500, 1860);

-- How many females and how many male passengers travelled for a minimum distance of 600 KM s? --

select passenger.gender, count(passenger.gender) as count from passenger where passenger.distance >= 600 group by passenger.gender;

-- Find the minimum ticket price for Sleeper Bus. --

select price.price from price where price.bus\_type = "sleeper" having min(price.price);

-- Select passenger names whose names start with character 'S' --

select passenger.passenger\_name from passenger where passenger.passenger\_name like "S%";

-- Calculate price charged for each passenger displaying Passenger name, Boarding City, Destination City, Bus\_Type, Price in the output --

select passenger.\*, price.price from passenger, price where passenger.distance = price.distance

and passenger.bus\_type = price.bus\_type group by passenger.passenger\_name;

-- What is the passenger name and his/her ticket price who travelled in Sitting bus for a distance of 1000 KMs --

select passenger.passenger\_name, price.price from passenger, price where passenger.distance = 1000

and passenger.bus\_type = "sleeper" group by passenger.passenger\_name;

-- What will be the Sitting and Sleeper bus charge for Pallavi to travel from Bangalore to Panaji?--

select price.bus\_type, price.price from price where price.distance = 600 group by price.price;

-- List the distances from the "Passenger" table which are unique (non-repeated distances) in descending order. --

select distinct(passenger.distance) from passenger order by passenger.distance desc;

/\* Display the passenger name and percentage of distance travelled by that passenger

from the total distance travelled by all passengers without using user variables \*/

select passenger.passenger\_name, ((passenger.distance / (select sum(passenger.distance) from passenger) \* 100))

as percentage\_of\_distance\_travelled from passenger group by passenger.passenger\_name;

/\* Display the distance, price in three categories in table Price

a) Expensive if the cost is more than 1000

b) Average Cost if the cost is less than 1000 and greater than 500

c) Cheap otherwise

\*/

select price.distance, price.price,

case

when price.price > 1000 then "expensive"

when price.price < 1000 and price.price > 500 then "Average cost"

else "cheap"

end as verdict from price;