Problem Description:

An LPG company wants to maintain Customer's details such as Customer Id, Name, Gender, Address, Phone No, Connection Type which is product Type (14.2 Kg or 19.0 Kg or 5Kg cylinders), Order Id, Order Date, Quantity (No of cylinders ordered), Payment Type, Ordered Status (Ordered or Cancelled), Order cancelled date, Reason for Order Cancellation, Invoice Id, Date of Invoice, Delivery Status (Delivered or Undelivered), If Undelivered, Date of bill cancelled, and Reason for Undelivery, Price of Product in every Month and Year.

Having all these details in place we need to create a Database called LPG and various tables in it. The tables needed and attributes which need to be in every table are given by the Organization. All you have to do is create tables with data in it and some queries so that the Organization can retrieve required information.

Look into the below points and do the needful

1. Write a program to create below tables.

Table:cust_details
Columns:

Id	int AI PK
Name	varchar(50)
Gender	varchar(1)
Address	varchar(100)
Phone_NO	bigint
Connection_Type	decimal(3,1)
No_Of_Cylinders	int

Table:orders
Columns:

Id	int AI PK
Date	date

Cust_Id	int FK
Quantity	int
Payment_type	varchar(30)
Status	varchar(30)

Table:cancelled_orders
Columns:

Order_Id	int FK
Date	date
Reason	varchar(50)

Table:billing_details
Columns:

Inv_Id	int AI PK
Date	date
Order_Id	int FK
Delivery_Status	varchar(30)

Table:cancelled_bills
Columns:

Inv_Id	int FK
Date	date
Reason	varchar(50)

Table:pricing

Columns: Insert Pricing Details every month of all products (14.2, 19.0, 5.0 Kg cylinders)

Type	decimal(3,1)
Month	varchar(10)
Year	int
Price	int

2. Insert data to tables. Below is the sample data for your information.
   Note: *You can change auto Increment value in attribute to any number. Use the below query to set the value. Start every Id with 1.*
   Query: *ALTER TABLE TableName AUTO_INCREMENT = 1;*

cust_details

(Let learners do the Address normalization)

Id	Name	Gender	Address	Phone_No	Connection_Type	No_of_Cylinders
1	Harish	M	1-2, bglr	1987654322	14.2	1
2	Amisha	F	32-12, bglr	1614322387	14.2	1
3	Ujjawal	M	19-0, gurgaon	1871614322	14.2	1
4	Anu	F	2-10, hyd	1000614322	19.0	5
5	Rakshitha	F	3-1-3, chennai	1614322551	19.0	10
6	Varuni	F	10-4, gurgaon	1432245789	14.2	1
7	Vamshi	M	31-14, hyd	1443324578	19.0	6

Orders

Id	Date	Cust_Id	Quantity	Payment_Type	Status
1	2021-10-01	6	1	online	cancelled
2	2021-10-01	3	1	POD	Ordered
3	2021-10-02	5	4	POD	Cancelled

4	2021-10-03	6	1	POD	Ordered
5	2021-10-04	3	1	Online	Ordered
6	2021-11-05	6	1	Online	Ordered
7	2021-11-06	4	4	Online	Ordered
8	2021-11-07	5	9	POD	Ordered
9	2021-11-09	7	5	Online	Ordered

cancelled_orders

Order_Id	**Date**	**Reason**
1	2021-10-02	Out of Station
3	2021-10-03	Mistakenly Ordered

billing_details

Inv_Id	**Date**	**Order_Id**	**Delivery_Status**
1	2021-10-03	2	Undelivered
2	2021-10-04	4	Delivered
3	2021-10-06	5	Delivered
4	2021-11-06	6	Delivered
5	2021-11-06	7	Delivered
6	2021-11-08	8	Delivered

cancelled_bills

Inv_Id	**Date**	**Reason**
1	2021-10-04	Insufficient Amount

Pricing

Type	**Month**	**Year**	**Price**

14.2	January	2021	925
19.0	January	2021	1223
5.0	January	2021	352
14.2	February	2021	931
19.0	February	2021	1025
5.0	February	2021	361
14.2	March	2021	910
19.0	March	2021	1225
5.0	March	2021	365
14.2	April	2021	942
19.0	April	2021	1300
5.0	April	2021	330
14.2	May	2021	942
19.0	May	2021	1280
5.0	May	2021	333
14.2	June	2021	958
19.0	June	2021	1283
5.0	June	2021	320
14.2	July	2021	950
19.0	July	2021	1295
5.0	July	2021	330
14.2	August	2021	947
19.0	August	2021	1298
5.0	August	2021	337
14.2	September	2021	963

19.0	September	2021	1306
5.0	September	2021	340
14.2	October	2021	960
19.0	October	2021	1310
5.0	October	2021	347
14.2	November	2021	970
19.0	November	2021	1313
5.0	November	2021	350
14.2	December	2021	974
19.0	December	2021	1320
5.0	December	2021	362
14.2	January	2022	999
19.0	January	2022	1309
5.0	January	2022	359

3. Write a query to display a table with customer Id, Name, Connection_Type and No_Of Cylinders ordered from orders table.
4. Display one customer from each product category who purchased maximum no of cylinders with Connection_Type, Cust_Id, Name and Quantity purchased.
5. Display Customer Id, Successfully_Delivered and value of customer based on purchase of cylinders using SQL Case Statement.
   when Successfully_Delivered >= 8 then 'Highly Valued'
   when Successfully_Delivered between 5 and 7 then 'Moderately Valued'
   Else 'Low Valued'
6. Display Customer Id, Name, Order_Id, Inv_Id, Delivery Date of all deliveries received by customer for all customers
7. Find the amount paid by the customer for every delivery taken for all customers with following details Customer_Id, Name, Order_Id, Order_Date, Inv_Id, Delivery_Date, Connection_Type and Price.
8. Create an SQL Stored Procedure “**PriceOfCurrentMonth**” to Identify the Price of all Products in the Current Month with Product_Type, Month, Year and Price in table.
9. Find Last Delivery Date from billing_details table of every customer and display customer Id and Name, Last_Delivery_Date and Quantity using Joins.
   (Note that the date in billing_details will act as last delivery date)

10. Display customer Id, Name, undelivered date and reason for undelivery using joins.
11. Display customer Id, Name, Date and reason for Cancelled Orders of all cancellations made by all customers.

Solution:

```
//create a database before proceeding. Mysql version 8 and above is recommended.
Create database LPG
```

1.
   ```
   create table Cust_Details (Id int primary key auto_increment, Name varchar(50), Gender varchar(1), Address varchar(100), Phone_NO bigint, Connection_Type decimal(3,1), No_Of_Cylinders int);

   create table Orders (Id int primary key auto_increment, Date date, Cust_Id int, Quantity int, Payment_type varchar(30), Status varchar(30), foreign key(Cust_Id) references Cust_details(Id));

   create table Cancelled_Orders (Order_Id int, Date date, Reason varchar(50), foreign key(Order_Id) references Orders(Id));

   create table Billing_Details(Inv_Id int Primary key auto_increment, Date date, Order_Id int, Delivery_Status varchar(30), foreign key(Order_Id) references Orders(Id));

   create table Cancelled_Bills (Inv_Id int, Date date, Reason varchar(50), foreign key(Inv_Id) references billing_details(Inv_Id));

   create table Pricing (Type decimal(3,1), Month varchar(10), Year int, Price int);
   ```

2. **Cust_Details**

   ```
   insert into cust_details (Name, Gender, Address, Phone_No, Connection_Type, No_Of_Cylinders) values ('Harish', 'M', '1-2, bglr', 1987654322, 14.2, 1);
   insert into cust_details (Name, Gender, Address, Phone_No, Connection_Type, No_Of_Cylinders) values ('Amisha', 'F', '32-12, bglr', 1614322387, 14.2, 1);
   insert into cust_details (Name, Gender, Address, Phone_No, Connection_Type, No_Of_Cylinders) values ('Ujjawal', 'M', '19-0, gurgaon', 1871614322, 14.2, 1);
   insert into cust_details (Name, Gender, Address, Phone_No, Connection_Type, No_Of_Cylinders) values ('Anu', 'F', '2-10, hyd', 1000614322, 19.0, 5);
   insert into cust_details (Name, Gender, Address, Phone_No, Connection_Type, No_Of_Cylinders) values ('Rakshitha', 'F', '3-1-3, chennai', 1614322551, 19.0, 10);
   insert into cust_details (Name, Gender, Address, Phone_No, Connection_Type, No_Of_Cylinders) values ('Varuni', 'F', '10-4, gurgaon', 1432245789, 14.2, 1);
   insert into cust_details (Name, Gender, Address, Phone_No, Connection_Type, No_Of_Cylinders) values ('Vamshi', 'M', '31-14, hyd', 14433245789, 19.0, 6);
   ```

   Orders

```
insert into orders (Date, Cust_Id, Quantity, Payment_type, Status) values('2021-10-01', 6, 1, 'online', 'cancelled');
insert into orders (Date, Cust_Id, Quantity, Payment_type, Status) values('2021-10-01', 3, 1, 'POD', 'Ordered');
insert into orders (Date, Cust_Id, Quantity, Payment_type, Status) values('2021-10-02', 5, 4, 'POD', 'Cancelled');
insert into orders (Date, Cust_Id, Quantity, Payment_type, Status) values('2021-10-03', 6, 1, 'POD',      'Ordered');
insert into orders (Date, Cust_Id, Quantity, Payment_type, Status) values('2021-10-04', 3, 1, 'Online', 'Ordered');
insert into orders (Date, Cust_Id, Quantity, Payment_type, Status) values('2021-11-05', 6, 1, 'Online', 'Ordered');
insert into orders (Date, Cust_Id, Quantity, Payment_type, Status) values('2021-11-06', 4, 4, 'Online', 'Ordered');
insert into orders (Date, Cust_Id, Quantity, Payment_type, Status) values('2021-11-07', 5, 9, 'POD', 'Ordered');
insert into orders (Date, Cust_Id, Quantity, Payment_type, Status) values('2021-11-09', 7, 5, 'Online', 'Ordered');
```

Cancelled Orders

```
insert into cancelled_orders values(1, '2021-10-02', 'Out of Station');
insert into cancelled_orders values(3, '2021-10-03', 'Mistakenly Ordered');
```

Billing_details

```
insert into billing_details(Date, Order_Id, Delivery_Status) values ('2021-10-03', 1, 'Delivered');
insert into billing_details(Date, Order_Id, Delivery_Status) values ('2021-10-03', 2, 'Undelivered');
insert into billing_details(Date, Order_Id, Delivery_Status) values ('2021-10-04', 4, 'Delivered');
insert into billing_details(Date, Order_Id, Delivery_Status) values ('2021-10-06', 5, 'Delivered');
insert into billing_details(Date, Order_Id, Delivery_Status) values ('2021-11-06', 6, 'Delivered');
insert into billing_details(Date, Order_Id, Delivery_Status) values ('2021-11-06', 7, 'Delivered');
insert into billing_details(Date, Order_Id, Delivery_Status) values ('2021-11-08', 8, 'Delivered');
```

Cancelled_Bills

```
insert into cancelled_bills values(2, '2021-10-04', 'Insufficient Amount');
```

Pricing

```
insert into pricing values(14.2, 'January', 2021, 925);
insert into pricing values(19.0, 'January', 2021, 1223);
insert into pricing values(5.0, 'January', 2021, 352);

insert into pricing values(14.2, 'February', 2021, 931);
insert into pricing values(19.0, 'February', 2021, 1025);
insert into pricing values(5.0, 'February', 2021, 361);

insert into pricing values(14.2, 'March', 2021, 910);
insert into pricing values(19.0, 'March', 2021, 1225);
insert into pricing values(5.0, 'March', 2021, 365);

insert into pricing values(14.2, 'April', 2021, 942);
insert into pricing values(19.0, 'April', 2021, 1300);
insert into pricing values(5.0, 'April', 2021, 330);

insert into pricing values(14.2, 'May', 2021, 942);
insert into pricing values(19.0, 'May', 2021, 1280);
insert into pricing values(5.0, 'May', 2021, 333);

insert into pricing values(14.2, 'June', 2021, 958);
insert into pricing values(19.0, 'June', 2021, 1283);
insert into pricing values(5.0, 'June', 2021, 320);

insert into pricing values(14.2, 'July', 2021, 950);
insert into pricing values(19.0, 'July', 2021, 1295);
insert into pricing values(5.0, 'July', 2021, 330);

insert into pricing values(14.2, 'August', 2021, 947);
insert into pricing values(19.0, 'August', 2021, 1298);
insert into pricing values(5.0, 'August', 2021, 337);

insert into pricing values(14.2, 'September', 2021, 963);
insert into pricing values(19.0, 'September', 2021, 1306);
insert into pricing values(5.0, 'September', 2021, 340);

insert into pricing values(14.2, 'October', 2021, 960);
insert into pricing values(19.0, 'October', 2021, 1310);
insert into pricing values(5.0, 'October', 2021, 347);

insert into pricing values(14.2, 'November', 2021, 970);
insert into pricing values(19.0, 'November', 2021, 1313);
insert into pricing values(5.0, 'November', 2021, 350);

insert into pricing values(14.2, 'December', 2021, 974);
insert into pricing values(19.0, 'December', 2021, 1320);
insert into pricing values(5.0, 'December', 2021, 362);

insert into pricing values(14.2,January, 2022, 999);
insert into pricing values(19.0, 'January', 2022, 1309);
insert into pricing values(5.0, 'January', 2022, 359);
```

3.

```
select C.Name, C.Connection_Type, Q.No_of_cylinders from Cust_Details as C inner join
(select Cust_Id, sum(Quantity) as no_of_cylinders from orders where status = 'ordered' group by cust_Id)
as Q on C.Id = Q.cust_id;
```

4. Using Rank

```
---------------------------------
select a.Cust_Id, a.Name, a.no_of_cylinders, a.Connection_Type from
(select C.Id as Cust_Id, C.Name, P.no_of_cylinders, C.Connection_Type,
          rank() over(partition by C.Connection_Type order by P.no_of_cylinders desc) as rnk
   from Cust_Details as C inner join
(select Cust_Id, sum(Quantity) as no_of_cylinders from orders where status = 'ordered' group by cust_Id)
as P on P.Cust_Id = C.Id
) a
where a.rnk=1;
```

Using joins

```
--------------------------------
select S.* from (
select Connection_Type, max(no_of_cylinders) as cyl from (
select C.Id as Cust_Id, C.Name, P.no_of_cylinders, C.Connection_Type from Cust_Details as C inner join
(select Cust_Id, sum(Quantity) as no_of_cylinders from orders where status = 'ordered' group by cust_Id)
as P on P.Cust_Id = C.Id) as Q group by connection_Type)
as R inner join (
select C.Id as Cust_Id, C.Name, P.no_of_cylinders, C.Connection_Type from Cust_Details as C inner join
(select Cust_Id, sum(Quantity) as no_of_cylinders from orders where status = 'ordered' group by cust_Id)
as P on P.Cust_Id = C.Id) as S where R.cyl = S.no_of_cylinders and R.connection_type = S.connection_type;
```

```
5.  select Cust_Id, Successfully_Delivered,
    Case
            when Successfully_Delivered >= 8 then 'Highly Valued'
      when Successfully_Delivered between 5 and 7 then 'Moderately Valued'
      Else 'Low Valued'
    End as Value from

    (select O.Cust_Id, sum(O.Quantity) as Successfully_Delivered from Orders as O inner
    join
    (select Order_Id from billing_details where delivery_status = 'Delivered')
    as P on P.Order_Id = O.Id group by Cust_Id)
    as Q;
```

```
6.  select C.Id as Cust_Id, C.Name, Delivery_Date from cust_details as C inner join (
    select cust_id, Delivery_Date from
    (select O.id, O.cust_id, D.Inv_Id, D.Delivery_Date from orders as O inner join
    (select Inv_Id, Order_Id, date as Delivery_Date from billing_details where
    Delivery_Status = 'Delivered')
    as D on O.id = D.Order_id)
    as P)
    as Q on Q.cust_id = C.Id order by Cust_Id;
```

```
7.  select Q.Customer_Id, Q.Name, Q.Order_Id, Q.Order_Date, Q.Inv_Id, Q.Delivery_Date,
    Q.Connection_Type, Pricing.Price from Pricing inner join
    (select C.Id as Customer_Id, C.Name, P.Order_Id, P.Order_Date, P.Inv_Id,
    P.Delivery_Date, C.Connection_Type, monthname(Delivery_Date) as month,
    year(Delivery_Date) as Year from cust_details as C inner join
    (select O.Id as Order_Id, O.date as Order_Date, O.Cust_Id, D.Inv_Id, D.Delivery_Date
    from orders as O inner join
    (select Inv_Id, Order_Id, date as Delivery_Date from billing_details where
    Delivery_Status = 'Delivered')
    as D on D.Order_Id = O.Id)
    as P on P.Cust_Id = C.Id)
    as Q on Q.Month = Pricing.Month and Q.Year = Pricing.Year and Q.Connection_Type =
    Pricing.Type order by Customer_Id;
```

```
    // enter the below code in stored procedure and save it
8.  CREATE PROCEDURE PriceOfCurrentMonth()
    Begin
    select * from Pricing where (Month, Year) In (select monthname(curdate()) as Month,
    year(curdate()) as Year);
    End
```

```
// in the editor, execute the below line
call PriceOfCurrentMonth;
```

9.

```
select C.Id as Cust_Id, C.Name, Q.Last_Delivery_Date, Q.Quantity from cust_details as C inner join
(select cust_id, max(Delivery_Date) as Last_Delivery_Date, Quantity from
(select O.id, O.cust_id, O.Quantity,D.Delivery_Date from orders as O inner join
(select Order_Id, date as Delivery_Date from billing_details where Delivery_Status = 'Delivered')
as D on O.id = D.Order_id)
as P group by (cust_id))
as Q on Q.cust_id = C.Id
order by cust_id;
```

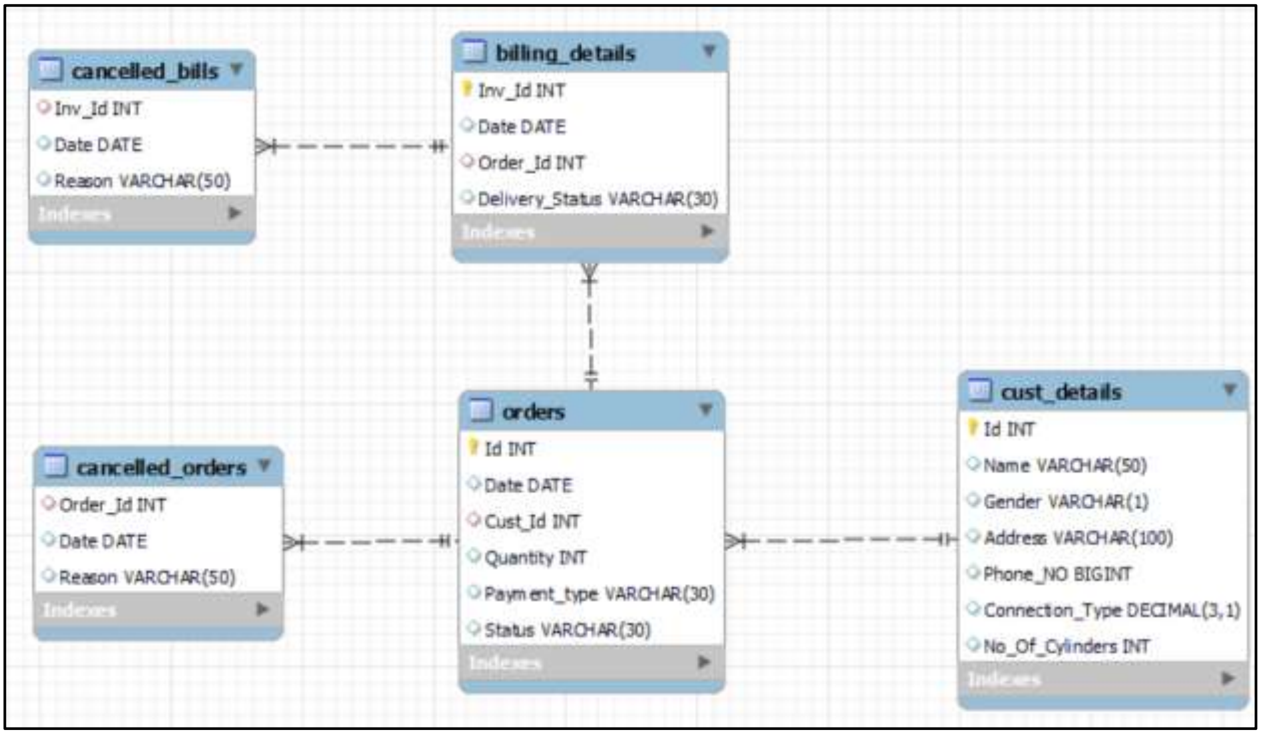
10.

```
select C.Id as Cust_Id, C.Name, R.Cancelled_Bill_Date, R.Reason from cust_details as C inner join
(select cust_id, Cancelled_Bill_Date, Reason from orders as O inner join
(select B.order_id, P.Date as Cancelled_Bill_Date, P.Reason from billing_details as B inner join
(select * from cancelled_bills)
as P on P.Inv_Id = B.Inv_Id)
as Q on Q.order_id = O.Id)
as R on R.cust_id = C.Id;
```

11.

```
select C.Id, C.Name, R.Cancelled_Order_Date, R.Reason from cust_details as C inner join
(select O.cust_id, Q.Cancelled_Order_Date, Q.Reason from orders as O inner join
(select O.Id, P.Date as Cancelled_Order_Date, P.Reason from orders as O inner join
(select * from cancelled_orders)
as P on P.Order_Id = O.Id)
as Q on Q.Id = O.Id)
as R on R.cust_id = C.Id;
```

ER-Diagram:


cancelled_bills
Inv_Id INT
Date DATE
Reason VARCHAR(50)
Indexes
billing_details
Inv_Id INT
Date DATE
Order_Id INT
Delivery_Status VARCHAR(30)
Indexes
cancelled_orders
Order_Id INT
Date DATE
Reason VARCHAR(50)
Indexes
orders
Id INT
Date DATE
Cust_Id INT
Quantity INT
Payment_type VARCHAR(30)
Status VARCHAR(30)
Indexes
cust_details
Id INT
Name VARCHAR(50)
Gender VARCHAR(1)
Address VARCHAR(100)
Phone_NO BIGINT
Connection_Type DECIMAL(3,1)
No_Of_Cylinders INT
Indexes