**Week 10 Day 2 Lab Coding Challenges**

**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 |

1. 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 |

1. Write a query to display a table with customer Id, Name, Connection\_Type and No\_Of Cylinders ordered from orders table.
2. Display one customer from each product category who purchased maximum no of cylinders with Connection\_Type, Cust\_Id, Name and Quantity purchased.
3. 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'

1. Display Customer Id, Name, Order\_Id, Inv\_Id, Delivery Date of all deliveries received by customer for all customers
2. 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.
3. 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.
4. Get the customer details whose id's are between 3(excluded) and 7(excluded)
5. 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)

1. Display customer Id, Name, undelivered date and reason for undelivery using joins.
2. 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);

1. **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);

1. 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;

1. 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;

1. 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;

1. 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;

1. 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

1. 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;

1. select \* from cust\_details limit 3 offset 3;
2. 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;

1. 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;

1. 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:**

