**Problem Description:**

You are a developer in the IT division of the popular LPG company, as part of the Automation spree, the LPG company wants to maintain Customer’s details such as

**1.Customer Id, Name, Gender, Address, Phone No, Connection Type** which is

**2.product Type (14.2 Kg or 19.0 Kg or 5Kg cylinders),**

**3.Order Id, Order Date, Quantity (No of cylinders ordered), Payment Type, Ordered Status (Ordered or Cancelled),**

**4.Order cancelled date, Reason for Order Cancellation**,

**5.Invoice Id, Date of Invoice, Delivery Status (Delivered or Undelivered),** If Undelivered, Date of

**6.bill cancelled, and Reason for Non Delivery**,

**Price of Product in every Month and Year**.

You are required to create an SQL based application(including Queries) following the below mentioned requirements.

1. Having all the above mentioned details in place a Database needs to be created , called LPG and various tables are to be created under it. The tables needed and attributes which are to be there in every table are given by the Organization and are as given below.
2. You are asked to create tables with the sample data in it and queries fulfilling the below mentioned requirements, so that the Organization can retrieve required information in an easier manner..

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 |

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 a maximum number

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 for question 5 taking the threshhold as parameter, customers having total qty orders above the threshold are 'GOOD CUSTOMER' else 'BAD CUSTOMER'.

9. Display Customer Name and count of orders cancelled by that customer.

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:**

**Answer 1:**

CREATE TABLE cust\_details (

id int(11) NOT NULL AUTO\_INCREMENT,

name varchar(10) ,

gender varchar(1) ,

address varchar(10) ,

phone int(10) ,

Connection\_Type decimal(3,1) ,

No\_Of\_Cylinders int(11) ,

PRIMARY KEY ( id )

);

CREATE TABLE orders (

id INT( 11 ) NOT NULL AUTO\_INCREMENT ,

odate DATE ,

qty INT( 11 ) ,

payment\_type VARCHAR( 6 ) ,

status VARCHAR( 1 ) ,

cust\_id INT( 11 ) ,

PRIMARY KEY ( id ) ,

FOREIGN KEY ( Cust\_Id ) REFERENCES Cust\_details( Id )

);

CREATE TABLE IF NOT EXISTS cancelled\_orders (

order\_id INT( 11 ) ,

cdate DATE ,

reason VARCHAR( 30 ) ,

FOREIGN KEY ( Order\_Id ) REFERENCES Orders( Id )

);

CREATE TABLE Billing\_Details (

bill\_no INT( 11 ) NOT NULL AUTO\_INCREMENT ,

bdate DATE ,

delivery\_status VARCHAR( 1 ) ,

order\_id INT( 11 ) ,

PRIMARY KEY ( bill\_no ) ,

FOREIGN KEY ( order\_Id ) REFERENCES Orders( Id )

);

CREATE TABLE IF NOT EXISTS cancelled\_bills (

bill\_no INT( 11 ) ,

cdate DATE ,

reason VARCHAR( 30 ) ,

FOREIGN KEY ( bill\_no ) REFERENCES billing\_details( bill\_no )

);

CREATE TABLE pricing (

connection\_type DECIMAL( 3, 1 ) ,

year INT( 11 ) ,

month VARCHAR( 10 ) ,

price INT( 11 )

);

**Answer 2)**

INSERT INTO cust\_details (`id`, `name`, `gender`, `address`, `phone`, `Connection\_Type`, `No\_Of\_Cylinders`) VALUES

(1, 'Harish', 'M', '1-2, bglr', 1987654322, '14.2', 1),

(2, 'Amisha', 'F', '32-12, bgl', 1614322387, '14.2', 1),

(3, 'Ujjawal', 'M', '19-0, gurg', 1871614322, '14.2', 1),

(4, 'Anu', 'F', '2-10, hyd', 1000614322, '19.0', 5),

(5, 'Rakshitha', 'F', '3-1-3, che', 1614322551, '19.0', 10),

(6, 'Varuni', 'F', '10-4, gurg', 1432245789, '14.2', 1),

(7, 'Vamshi', 'M', '31-14, hyd', 14433245789, '19.0', 6);

INSERT INTO orders ( id , odate , qty , payment\_type , status , cust\_id ) VALUES

(2, '2021-10-01', 1, 'Online', 'C', 6),

(3, '2021-10-01', 1, 'POD', 'O', 3),

(4, '2021-10-02', 4, 'POD', 'C', 5),

(5, '2021-10-03', 1, 'POD', 'O', 6),

(6, '2021-10-04', 1, 'Online', 'O', 3),

(7, '2021-11-05', 1, 'Online', 'O', 6),

(8, '2021-11-06', 4, 'Online', 'O', 4),

(9, '2021-11-07', 9, 'POD', 'O', 5),

(10, '2021-11-09', 5, 'Online', 'O', 7);

INSERT INTO cancelled\_orders ( order\_id , cdate , reason ) VALUES

(1, '2021-10-02', 'Out of Station'),

(3, '2021-10-03', 'Mistakenly Ordered');

INSERT INTO billing\_details ( bill\_no , bdate , delivery\_status , order\_id ) VALUES

(1, '2021-10-03', 'D', 1),

(2, '2021-10-03', 'U', 2),

(3, '2021-10-04', 'D', 4),

(4, '2021-10-06', 'D', 5),

(5, '2021-11-06', 'D', 6),

(6, '2021-11-06', 'D', 7),

(7, '2021-11-08', 'D', 8);

INSERT INTO cancelled\_bills ( bill\_no , cdate , reason ) VALUES

(2, '2021-10-04', 'Insufficient Amount');

INSERT INTO pricing ( connection\_type , year , month , price ) VALUES

('5.0', 2023, 'Jan', 500),

('14.2', 2023, 'Jan', 1500),

('19.0', 2023, 'Jan', 1000),

('5.0', 2023, 'Feb', 500),

('14.2', 2023, 'Feb', 1500),

('19.0', 2023, 'Feb', 1000);

**Answer 3)**

SELECT Cust\_details.id, name, connection\_type, SUM( qty )

FROM Cust\_details, orders

WHERE Cust\_details.id = orders.cust\_id

AND

STATUS = 'O'

GROUP BY name;

**Answer 4)**

select \* 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(Qty) as no\_of\_cylinders from orders where status = 'O' 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(Qty) as no\_of\_cylinders from orders where status = 'O'

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;

**Answer 5:**

SELECT name, SUM( qty ),

CASE

when SUM( qty )>=8 then 'Platinum'

when SUM( qty ) between 5 and 7 then 'Gold'

else 'Silver'

END as 'Customer Value'

FROM Cust\_details

JOIN orders ON Cust\_details.id = orders.cust\_id

WHERE

STATUS = 'O'

GROUP BY name

**Answer 6:**

select \*

from Cust\_details

join

orders

on Cust\_details.id=orders.cust\_id

join

billing\_details

on orders.id =billing\_details.order\_id

where delivery\_status='D';

**Answer 7 :**

SELECT name, price \* qty

FROM pricing

JOIN cust\_details

USING ( connection\_type )

JOIN orders ON cust\_details.id = orders.cust\_id

JOIN billing\_details ON orders.id = billing\_details.order\_id

WHERE delivery\_status IN ('D')

GROUP BY name

**Answer 8:**

DELIMITER &&

CREATE PROCEDURE customerValueNew(in threshold int)

BEGIN

SELECT name, SUM( qty ),

CASE

WHEN SUM( qty ) > threshold THEN 'Good'

ELSE 'Not Good'

END AS 'Customer Value'

FROM Cust\_details

JOIN orders ON Cust\_details.id = orders.cust\_id

WHERE

STATUS = 'O'

GROUP BY name;

END &&

DELIMITER ;

**Answer 9 :**

SELECT name,count(\*)

FROM cust\_details

JOIN orders ON cust\_details.id = orders.cust\_id

JOIN cancelled\_orders ON orders.id = cancelled\_orders.order\_id

group by name;

**Answer 10:**

select cust\_details.Id, Name, cdate,reason

from cust\_details

join

orders

on cust\_details.id=orders.cust\_id

join billing\_details

on orders.id=billing\_details.order\_id

join

cancelled\_bills

using(bill\_no);

**Answer 11:**

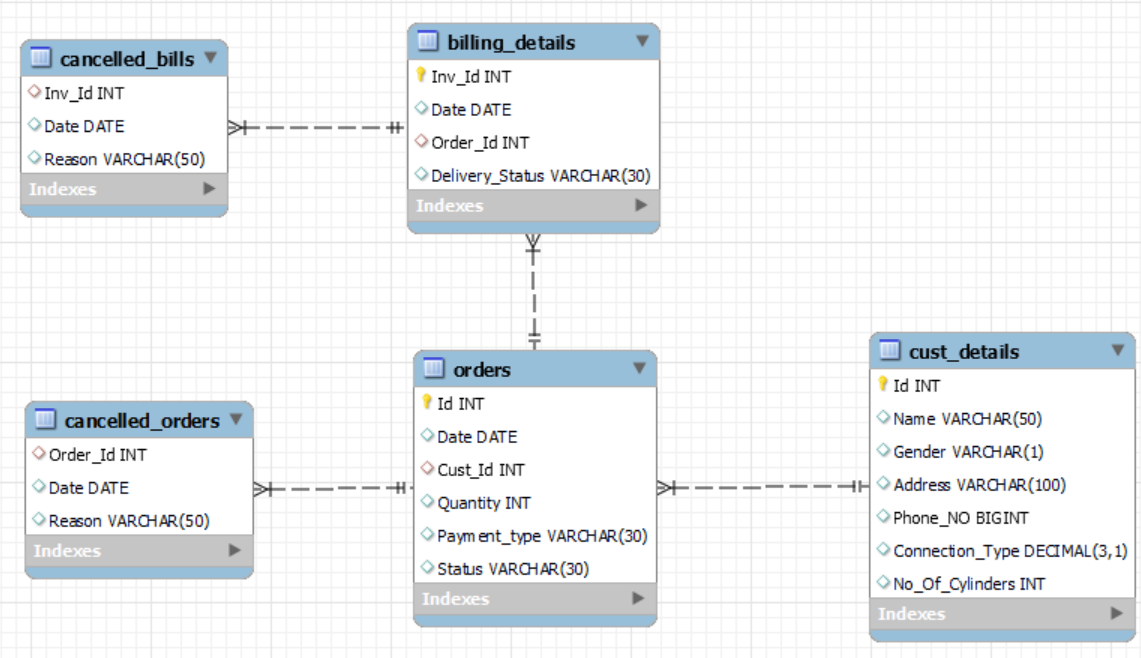
select cust\_details.Id, Name, cdate,reason

FROM cust\_details

JOIN orders ON cust\_details.id = orders.cust\_id

JOIN cancelled\_orders ON orders.id = cancelled\_orders.order\_id;

**ER-Diagram:**



**Learning Objectives:**

The Objective of this Mentoring Session based Project is to enable the learners to gain the knowledge on the RDBMS/SQL related concepts on the following heads and also post this Project, learners should be able to comfortably create Databases, Tables & Manage & manipulate data in them.

1. Insert Data into the tables.
2. Query Database using Joins
3. Query Database using Sub Queries
4. Query Database using Joins in conjunction with Sub Queries
5. Create Procedures and invoke them
6. Sorting records using OrderBY clause
7. Query Database using Group By clause
8. Use Case construct in Queries.