



WATER REFILLING SYSTEM



Prepared For

Dr.Kiran Sharma
Ass.Professor
BML Munjal University

By

Godavarthi Sai Nikhil
210214
CSE-1



**BML MUNJAL
UNIVERSITY™**

Description:

The water refilling system is to provide our customer needs through an efficient delivery system and a quick response on their orders and queries. And also enable the user to record things that are being purchased by the customers.

- ❖ The water purifying system at a water refilling station produces portable drinking water. So, I designed a system in order to be on top and align with the fast-growing business which is in demand in rural areas.
- ❖ The purpose of the water refilling management system is to overcome difficulties in manual operation in the refilling stations. The difficulty in the manual system is one of the reasons that the customer is not satisfying and keeping of records is often misplaced and not secure.
- ❖ This system manages to display the data to be filled by the user according to the information of the customer in an organized manner, such that their personal details, the services they want to avail as well as the payment on the transaction they purchased. The system keeps the information of the customer and the details of what they purchased.
- ❖ It consists all the records for the location of the customer, date of transaction, schedule of delivery, contact number of people assign to deliver, and the payment of the customer to the quantity of product that about to deliver.
- ❖ The system also views the information about the availability of the products as well as the containers. The system views the available containers to provide stocks again.
- ❖ It stores the information in an organized so that it is easy to the owner to access the detail of his/her employee. Upon having this system, it will provide the capacity for the owner and customer to transact without spending time and effort.

The water refilling system is represented using **ER-diagram** and the structure is described as follows:

- Entity is represented by the rectangle shape. The entity will be our database table of Water Refilling System ER Diagram later on.
- Attribute is represented by the oval shape. This will be the columns or fields of each table in the Water Refilling System ER Diagram.
- Relationship is represented by diamond shape. This will determine the relationships among entities. This is usually in a form of primary key to foreign key connection.

The design of Water Refilling System ER Diagram consists of 7 entities; the specified entities will be the database tables in the design.

The details of customer (name, address, phone no) is stored in customer entity and details of employee (name, email, age, address, phone no) is stored in employee entity.

Relationships among Entities:

- Customer requests to purchase the products which leads to **one-to-many** relationship.
- Each Customer makes many payments which leads to **one-to-many** relationship between them.
- The customer makes payment on their delivery request of water supply which leads to **one-to-many** relationship between them.
- Different Products has different types of containers which lead to **many-to-many** relationship between them.
- Many Products has many payments which lead to **many-to-many** relationship between them.
- Different Employee manages the different transactions which leads to **many-to-many** relationship between them.
- Employees tracks the payments done by customers which leads to **many-to-many** relationship between them.
- Each payment has its own unique transaction so, it has **one-to-one** relationship between them.
- The customer makes transaction on their delivery request which leads to **one-to-one** relationship.

Water Refilling System entities and their attributes:

- **Customer Entity:**
 - **Attributes** are C_Id, C_Name, C_Address and C_Contact_Number.
 - C_Id is the **primary key**.
 - C_Contact_Number is the **Multivalued key**.
- **Product Entity:**
 - **Attributes** are P_Id, P_Name, Quantity, P_Refill_Price, P_Discount, P_Category and P_Price.
 - P_Id is the **primary key**.
 - P_Name is the **Candidate Key**.
- **Types_of_Container Entity:**
 - **Attributes** are ToC_Id, Container_Name, Quantity and ToC_Unit_Price.
 - ToC_Id is the **primary key**.
- **Employee Entity:**
 - **Attributes** are E_Id, E_Name, E_Address, E_Age, E_Email and E_Contact_No.
 - E_Id is the **primary key**.
 - E_Email is the **Candidate Key**.
 - E_Age is a **Derived Key**.
 - E_Contact_No is the **Multivalued key**.

- **Payment Entity:**
 - **Attributes** are **Pay_Id, C_Id, P_Id, Quantity, P_Price** and **Pay_Date_Order**.
 - **Pay_Id** is the **primary key**.
 - **C_Id, P_Id** are **Foreign Keys**.
- **Transaction Entity:**
 - **Attributes** are **T_Id, Pay_Id, C_Id, P_Price** and **T_Date**.
 - **T_Id** is the **primary key**.
 - **C_Id, Pay_Id** are **Foreign Keys**.
- **Delivery Entity:**
 - **Attributes** are **D_Id, T_Id, C_Id, Delivery_Man, P_Name** and **Date_of_Delivery**.
 - **D_Id** is the **primary key**.
 - **C_Id, T_Id,** are **Foreign Keys**.

Normalization:

1NF Form:

Relations in 1NF:

- Product
- Containers
- Payment
- Transaction
- Delivery

Relations are not in 1NF:

- Customer
- Employee

1. Customer table is not in 1NF because there are multivalued in Customer Contact number.

C_Id	C_Name	C_Address	C_Contact_Number
101	chris grey	aguisan	9461243372
102	mark john	talaban	9368787217
103	kim paterno	carabalan	9356484858, 12345675433
104	rex notar	aguisan	8486486945

So, the normalization for the above table is

C_Id	C_Name	C_Address	C_Contact_Number
101	chris grey	aguisan	9461243372
102	mark john	talaban	9368787217
103	kim paterno	carabalan	9356484858
103	kim paterno	carabalan	12345675433
104	rex notar	aguisan	8486486945

2. Employee table is not in 1NF because there are multivalued in Employee Contact number.

E_Id	E_Name	E_Age	E_Email	E_Address	E_Contact_Number
50	aurelio laja	21	aurelio32@gmail.com	binalbagan	9126534451
51	noel dalagan	30	noeld@gmail.com	enclaro	9214567999, 4567848767
52	mario tanio	30	mario34@gmail.com	binalbagan	9461263436

So, the normalization for the above table is

E_Id	E_Name	E_Age	E_Email	E_Address	E_Contact_Number
50	aurelio laja	21	aurelio32@gmail.com	binalbagan	9126534451
51	noel dalagan	30	noeld@gmail.com	enclaro	9214567999
51	noel dalagan	30	noeld@gmail.com	enclaro	4567848767
52	mario tanio	30	mario34@gmail.com	binalbagan	9461263436

2NF Form:

Relations in 2NF:

- Customer
- Product
- Containers
- Employee
- Payment
- Transaction

Relations are not in 2NF:

- Delivery

1. Delivery table is not in the 2NF because P_Name is not depended on D_Id.

D_Id	C_Id	T_Id	P_Name	Date_of_Delivery	Delivery_Man
200	101	10101	round gallon	04-02-2021	mario tanio
201	102	10102	gallon with faucet	05-02-2021	bruno mars
202	103	10103	round gallon	04-03-2021	mario tanio

So, the normalization for the above table is

D_Id	C_Id	T_Id	Date_of_Delivery	Delivery_Man
200	101	10101	04-02-2021	mario tanio
201	102	10102	05-02-2021	bruno mars
202	103	10103	04-03-2021	mario tanio

D_Id	P_Name
200	round gallon
201	gallon with faucet
202	round gallon

3NF Form:

Relations in 3NF:

- Customer
- Product
- Employee
- Delivery

Relations are not in 3NF:

- Containers
- Payment
- Transaction

1. Container table is not in 3NF Because Quantity and unit price depends on container name not on ToC_Id.

ToC_Id	Containers_Name	Quantity	ToC_Unit_Price
1	round gallon	10	125.5
2	gallon with faucet	5	180.65
3	bottle	20	60.75
4	bottle container	13	175.52

So, we divide the table into 2 tables.

ToC_Id	Containers_Name
1	round gallon
2	gallon with faucet
3	bottle
4	bottle container

Containers_Name	Quantity	ToC_Unit_Price
round gallon	10	125.5
gallon with faucet	5	180.65
bottle	20	60.75
bottle container	13	175.52

2. Payment table is not in 3NF Because Quantity and price depends on product id not on payment id.

Pay_Id	P_Id	C_Id	Quantity	P_Price	Pay_Date_Order
1001	1015	101	2	299	01-02-2021
1002	1016	101	3	711	01-02-2021
1003	1016	102	2	474	02-02-2021
1004	1019	102	1	245.52	02-02-2021
1005	1018	103	5	417.225	03-03-2021
1006	1017	103	1	145.5	04-03-2021

So, we divide the table into 2 tables.

Pay_Id	P_Id	C_Id	Pay_Date_Order
1001	1015	101	01-02-2021
1002	1016	101	01-02-2021
1003	1016	102	02-02-2021
1004	1019	102	02-02-2021
1005	1018	103	03-03-2021
1006	1017	103	04-03-2021

P_Id	Quantity	P_Price
1015	2	299
1016	3	711
1016	2	474
1019	1	245.52
1018	5	417.225
1017	1	145.5

3. transaction table is not in 3NF Because price depends on payment id not on Transaction id.

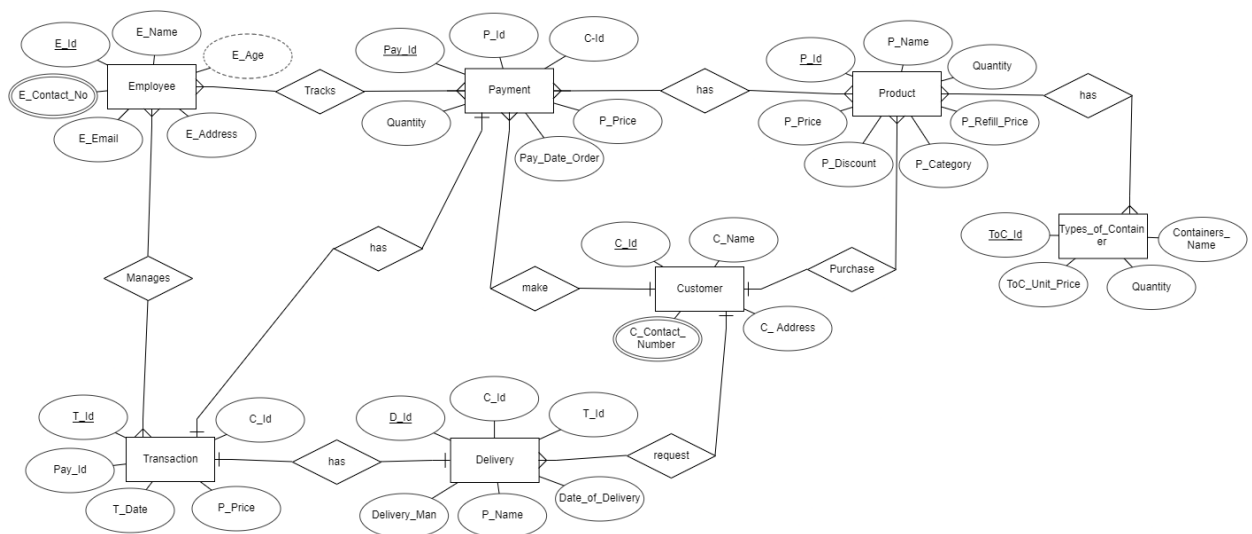
T_Id	C_Id	Pay_Id	P_Price	T_Date
10101	101	1001	299	01-02-2021
10102	102	1003	474	02-02-2021
10103	103	1006	145.5	04-03-2021

So, we divide the table into 2 tables i.e.,

T_Id	C_Id	Pay_Id	T_Date
10101	101	1001	01-02-2021
10102	102	1003	02-02-2021
10103	103	1006	04-03-2021

Pay_Id	P_Price
1001	299
1003	474
1006	145.5

ER Model:



Queries

DDL Statements:

[1] Table for Customer Entity:

```
CREATE TABLE `dbms_project`.`tblcustomer` (  
  `C_Id` INT NOT NULL AUTO_INCREMENT, `C_Name` VARCHAR(45) NOT NULL, `C_Address`  
  VARCHAR(45) NOT NULL, `C_Contact_Number` VARCHAR(30) NOT NULL,  
  PRIMARY KEY (`C_Id`))  
ENGINE = InnoDB;
```

[2] Table for Product Entity:

```
CREATE TABLE `dbms_project`.`tblproduct` (  
  `P_Id` INT NOT NULL AUTO_INCREMENT, `P_Name` VARCHAR(45) NOT NULL, `P_Category`  
  VARCHAR(45) NOT NULL, `P_Refill_Price` FLOAT NOT NULL, `Quantity` INT(11) NOT NULL,  
  `P_Discount` FLOAT NOT NULL, `P_Price` FLOAT NOT NULL,  
  PRIMARY KEY (`P_Id`));
```

[3] Table for Container Entity:

```
CREATE TABLE `dbms_project`.`tblcontainer` (  
  `Toc_Id` INT(11) NOT NULL AUTO_INCREMENT, `Containers_ Name` VARCHAR(100) NOT NULL,  
  `Quantity` INT(11) NOT NULL, `ToC_Unit_Price` FLOAT NOT NULL,  
  PRIMARY KEY (`tblcontainer`));
```

[4] Table for Employee Entity:

```
CREATE TABLE `dbms_project`.`tblemployee` (  
  `E_Id` INT NOT NULL AUTO_INCREMENT, `E_Name` VARCHAR(45) NOT NULL,  
  `E_Age` INT NOT NULL, `E_Email` VARCHAR(45) NOT NULL,  
  `E_Address` VARCHAR(45) NOT NULL, `E_Contact_No` VARCHAR(45) NOT NULL,  
  PRIMARY KEY (`E_Id`));
```

[5] Table for Payment Entity:

```
CREATE TABLE `dbms_project`.`tblpayment` (  
  `Pay_Id` INT NOT NULL AUTO_INCREMENT, `P_Id` INT NOT NULL, `C_Id` INT NOT NULL,  
  `Quantity` INT(11) NOT NULL, `P_Price` FLOAT NOT NULL, `Pay_Date_Order` DATE NOT NULL,  
  PRIMARY KEY (`Pay_Id`));
```

[6] Table for Transaction Entity:

```
CREATE TABLE `dbms_project`.`tbltransaction` (  
  `T_Id` INT NOT NULL AUTO_INCREMENT, `C_Id` INT NOT NULL, `Pay_Id` INT NOT NULL, `P_Price`  
  FLOAT NOT NULL, `T_Date` DATE NOT NULL,  
  PRIMARY KEY (`T_Id`));
```


[7] Table for Delivery Entity:

```
CREATE TABLE `dbms_project`.`tbldelivery` (  
  `D_Id` INT NOT NULL AUTO_INCREMENT, `C_Id` INT NOT NULL, `T_Id` INT NOT NULL, `P_Name`  
  VARCHAR(45) NOT NULL, `Date_of_Delivery` DATE NOT NULL,  
  `Delivery_Man` VARCHAR(45) NOT NULL,  
  PRIMARY KEY (`D_Id`));
```

Values:

[1] Values in Customer Entity:

```
INSERT INTO `dbms_project`.`tblcustomer` (`C_Id`, `C_Name`, `C_Address`, `C_Contact_Number`)  
VALUES ('101', 'chris grey', 'aguisan', '9461243372');  
INSERT INTO `dbms_project`.`tblcustomer` (`C_Id`, `C_Name`, `C_Address`, `C_Contact_Number`)  
VALUES ('102', 'mark john', 'talaban', '9368787217');  
INSERT INTO `dbms_project`.`tblcustomer` (`C_Id`, `C_Name`, `C_Address`, `C_Contact_Number`)  
VALUES ('103', 'kim paterno', 'carabalan', '9356484858,1234567872');  
INSERT INTO `dbms_project`.`tblcustomer` (`C_Id`, `C_Name`, `C_Address`, `C_Contact_Number`)  
VALUES ('104', 'rex notar', 'aguisan', '8486486945');
```

```
SELECT * FROM tblcustomer;
```

```
mysql> select * from tblcustomer;  
+-----+-----+-----+-----+  
| C_Id | C_Name      | C_Address | C_Contact_Number |  
+-----+-----+-----+-----+  
| 101  | chris grey  | aguisan  | 9461243372       |  
| 102  | mark john   | talaban  | 9368787217       |  
| 103  | kim paterno | carabalan | 9356484858,1234567872 |  
| 104  | rex notar   | aguisan  | 8486486945       |  
+-----+-----+-----+-----+  
4 rows in set (0.00 sec)
```

[2] Values in Product Entity:

```
INSERT INTO `dbms_project`.`tblproduct` (`P_Id`, `P_Name`, `P_Category`, `P_Refill_Price`,  
`Quantity`, `P_Discount`, `P_Price`) VALUES ('1015', 'round gallon', 'mineral', '60', '10', '60', '149.5');  
INSERT INTO `dbms_project`.`tblproduct` (`P_Id`, `P_Name`, `P_Category`, `P_Refill_Price`,  
`Quantity`, `P_Discount`, `P_Price`) VALUES ('1016', 'gallon with faucet', 'mineral', '80.5', '20', '30',  
'237');  
INSERT INTO `dbms_project`.`tblproduct` (`P_Id`, `P_Name`, `P_Category`, `P_Refill_Price`,  
`Quantity`, `P_Discount`, `P_Price`) VALUES ('1017', 'round gallon', 'alkaline', '50', '5', '40', '145.5');  
INSERT INTO `dbms_project`.`tblproduct` (`P_Id`, `P_Name`, `P_Category`, `P_Refill_Price`,  
`Quantity`, `P_Discount`, `P_Price`) VALUES ('1018', 'bottle', 'alkaline', '75.65', '18', '30', '83.445');  
INSERT INTO `dbms_project`.`tblproduct` (`P_Id`, `P_Name`, `P_Category`, `P_Refill_Price`,  
`Quantity`, `P_Discount`, `P_Price`) VALUES ('1019', 'bottle container', 'ALKALINE', '100', '8', '70',  
'245.52');
```

```
SELECT * FROM tblproduct;
```

```
mysql> SELECT * FROM tblproduct;
```

P_Id	P_Name	P_Category	P_Refill_Price	Quantity	P_Discount	P_Price
1015	round gallon	mineral	60	10	60	149.5
1016	gallon with faucet	mineral	80.5	20	30	237
1017	round gallon	alkaline	50	5	40	145.5
1018	bottle	alkaline	75.65	18	30	83.445
1019	bottle container	ALKALINE	100	8	70	245.52

```
5 rows in set (0.00 sec)
```

[3] Values in Container Entity:

```
INSERT INTO `dbms_project`.`tblcontainer` (`ToC_Id`, `Containers_Name`, `Quantity`,
`ToC_Unit_Price`) VALUES ('1', 'round gallon', '10', '125.5');
INSERT INTO `dbms_project`.`tblcontainer` (`ToC_Id`, `Containers_Name`, `Quantity`,
`ToC_Unit_Price`) VALUES ('2', 'gallon with faucet', '5', '180.65');
INSERT INTO `dbms_project`.`tblcontainer` (`ToC_Id`, `Containers_Name`, `Quantity`,
`ToC_Unit_Price`) VALUES ('3', 'bottle', '20', '60.75');
INSERT INTO `dbms_project`.`tblcontainer` (`ToC_Id`, `Containers_Name`, `Quantity`,
`ToC_Unit_Price`) VALUES ('4', 'bottle container', '13', '175.52');
```

```
SELECT * FROM tblcontainer;
```

```
mysql> SELECT * FROM tblcontainer;
```

ToC_Id	Containers_Name	Quantity	ToC_Unit_Price
1	round gallon	10	125.5
2	gallon with faucet	5	180.65
3	bottle	20	60.75
4	bottle container	13	175.52

```
4 rows in set (0.00 sec)
```

[4] Values in Employee Entity:

```
INSERT INTO `dbms_project`.`tblemployee` (`E_Id`, `E_Name`, `E_Age`, `E_Email`, `E_Address`,
`E_Contact_No`) VALUES ('50', 'aurelio laja', '21', 'aurelio32@gmail.com', 'binalbagan',
'9126534451');
INSERT INTO `dbms_project`.`tblemployee` (`E_Id`, `E_Name`, `E_Age`, `E_Email`, `E_Address`,
`E_Contact_No`) VALUES ('51', 'noel dalagan', '30', 'noeld@gmail.com', 'enclaro', '9214567999,
4567848767');
INSERT INTO `dbms_project`.`tblemployee` (`E_Id`, `E_Name`, `E_Age`, `E_Email`, `E_Address`,
`E_Contact_No`) VALUES ('52', 'mario tanio', '30', 'mario34@gmail.com', 'binalbagan', '9461263436');
```

```
SELECT * FROM tblemployee;
```

```
mysql> select * from tblemployee;
```

E_Id	E_Name	E_Age	E_Email	E_Address	E_Contact_No
50	aurelio laja	21	aurelio32@gmail.com	binalbagan	9126534451
51	noel dalagan	30	noeld@gmail.com	enclaro	9214567999,4567848767
52	mario tanio	30	mario34@gmail.com	binalbagan	9461263436

```
3 rows in set (0.00 sec)
```

[5] Values in Payment Entity:

```
INSERT INTO `dbms_project`.`tblpayment` (`Pay_Id`, `P_Id`, `C_Id`, `Quantity`, `P_Price`, `Pay_Date_Order`) VALUES ('1001', '1015', '101', '2', '299', '2021-02-01');
INSERT INTO `dbms_project`.`tblpayment` (`Pay_Id`, `P_Id`, `C_Id`, `Quantity`, `P_Price`, `Pay_Date_Order`) VALUES ('1002', '1016', '101', '3', '711', '2021-02-01');
INSERT INTO `dbms_project`.`tblpayment` (`Pay_Id`, `P_Id`, `C_Id`, `Quantity`, `P_Price`, `Pay_Date_Order`) VALUES ('1003', '1016', '102', '2', '474', '2021-02-02');
INSERT INTO `dbms_project`.`tblpayment` (`Pay_Id`, `P_Id`, `C_Id`, `Quantity`, `P_Price`, `Pay_Date_Order`) VALUES ('1004', '1019', '102', '1', '245.52', '2021-02-02');
INSERT INTO `dbms_project`.`tblpayment` (`Pay_Id`, `P_Id`, `C_Id`, `Quantity`, `P_Price`, `Pay_Date_Order`) VALUES ('1005', '1018', '103', '5', '417.225', '2021-03-03');
INSERT INTO `dbms_project`.`tblpayment` (`Pay_Id`, `P_Id`, `C_Id`, `Quantity`, `P_Price`, `Pay_Date_Order`) VALUES ('1006', '1017', '103', '1', '145.5', '2021-03-04');
```

```
SELECT * FROM tblpayment;
```

```
mysql> SELECT * FROM tblpayment;
```

Pay_Id	P_Id	C_Id	Quantity	P_Price	Pay_Date_Order
1001	1015	101	2	299	2021-02-01
1002	1016	101	3	711	2021-02-01
1003	1016	102	2	474	2021-02-02
1004	1019	102	1	245.52	2021-02-02
1005	1018	103	5	417.225	2021-03-03
1006	1017	103	1	145.5	2021-03-04

```
6 rows in set (0.00 sec)
```

[6] Values in Transaction Entity:

```
INSERT INTO `dbms_project`.`tbltransaction` (`T_Id`, `C_Id`, `Pay_Id`, `P_Price`, `T_Date`) VALUES ('10101', '101', '1001', '299', '2021-02-01');
INSERT INTO `dbms_project`.`tbltransaction` (`T_Id`, `C_Id`, `Pay_Id`, `P_Price`, `T_Date`) VALUES ('10102', '102', '1003', '474', '2021-02-02');
INSERT INTO `dbms_project`.`tbltransaction` (`T_Id`, `C_Id`, `Pay_Id`, `P_Price`, `T_Date`) VALUES ('10103', '103', '1006', '145.5', '2021-03-04');
```

```
SELECT * FROM tbltransaction;
```

```
mysql> SELECT * FROM tbltransaction;
+-----+-----+-----+-----+-----+
| T_Id  | C_Id | Pay_Id | P_Price | T_Date  |
+-----+-----+-----+-----+-----+
| 10101 | 101  | 1001   | 299     | 2021-02-01 |
| 10102 | 102  | 1003   | 474     | 2021-02-02 |
| 10103 | 103  | 1006   | 145.5   | 2021-03-04 |
+-----+-----+-----+-----+-----+
3 rows in set (0.00 sec)
```

[7] Values in Delivery Entity:

```
INSERT INTO `dbms_project`.`tbldelivery` (`D_Id`, `C_Id`, `T_Id`, `P_Name`, `Date_of_Delivery`,
`Delivery_Man`) VALUES ('200', '101', '10101', 'round gallon', '2021-02-04', 'mario tanio');
INSERT INTO `dbms_project`.`tbldelivery` (`D_Id`, `C_Id`, `T_Id`, `P_Name`, `Date_of_Delivery`,
`Delivery_Man`) VALUES ('201', '102', '10102', 'gallon with faucet', '2021-02-05', 'bruno mars');
INSERT INTO `dbms_project`.`tbldelivery` (`D_Id`, `C_Id`, `T_Id`, `P_Name`, `Date_of_Delivery`,
`Delivery_Man`) VALUES ('202', '103', '10103', 'round gallon', '2021-03-04', 'mario tanio');
```

```
SELECT * FROM tbldelivery;
```

```
mysql> SELECT * FROM tbldelivery;
+-----+-----+-----+-----+-----+-----+
| D_Id | C_Id | T_Id | P_Name           | Date_of_Delivery | Delivery_Man |
+-----+-----+-----+-----+-----+-----+
| 200  | 101  | 10101 | round gallon     | 2021-02-04      | mario tanio |
| 201  | 102  | 10102 | gallon with faucet | 2021-02-05      | bruno mars  |
| 202  | 103  | 10103 | round gallon     | 2021-03-04      | mario tanio |
+-----+-----+-----+-----+-----+-----+
3 rows in set (0.00 sec)
```

1) SELECT STATEMENT USING DIFFERENT FUNCTIONS

Q 1.1 Write a query that display the list of customer payment details that have balance more than 400.

Solution:

```
select Pay_Id,P_Id, C_Id,Quantity,P_Price from dbms_project.tblpayment where P_Price>400;
```

```
mysql> select Pay_Id,P_Id, C_Id,Quantity,P_Price from dbms_project.tblpayment where P_Price>400;
+-----+-----+-----+-----+-----+
| Pay_Id | P_Id | C_Id | Quantity | P_Price |
+-----+-----+-----+-----+-----+
| 1002   | 1016 | 101  | 3        | 711     |
| 1003   | 1016 | 102  | 2        | 474     |
| 1005   | 1018 | 103  | 5        | 417.225 |
+-----+-----+-----+-----+-----+
3 rows in set (0.00 sec)
```

Q 1.2 write a query that display the maximum amount received.

Solution:

select Pay_Id,P_Id,C_Id,max(P_Price) from tblpayment;

```
mysql> select Pay_Id,P_Id,C_Id,max(P_Price) from tblpayment;
+-----+-----+-----+-----+
| Pay_Id | P_Id | C_Id | max(P_Price) |
+-----+-----+-----+-----+
| 1001   | 1015 | 101  | 711          |
+-----+-----+-----+-----+
1 row in set (0.00 sec)
```

2) SELECT STATEMENT USING Sub Query

Q 2.1 Write a query that display the customer names who lived in aguisan

Solution:

select C_Name,C_Contact_Number from tblcustomer where C_Address in(select C_Address from tblcustomer where C_Address = 'aguisan');

```
mysql> select C_Name,C_Contact_Number from tblcustomer where C_Address in(select C_Address from tblcustomer where C_Address = 'aguisan');
+-----+-----+
| C_Name | C_Contact_Number |
+-----+-----+
| chris grey | 9461243372 |
| rex notar | 8486486945 |
+-----+-----+
2 rows in set (0.00 sec)
```

3) SELECT STATEMENT USING Count & Group Functions

Q 3.1 Display the name and the total number of Containers in tblproduct.

Solution:

select P_Name, count(P_Name) "Total Number of Containers" from tblproduct group by P_Name;

```
mysql> select P_Name, count(P_Name) "Total Number of Containers" from tblproduct group by P_Name;
+-----+-----+
| P_Name | Total Number of Containers |
+-----+-----+
| round gallon | 2 |
| gallon with faucet | 1 |
| bottle | 1 |
| bottle container | 1 |
+-----+-----+
4 rows in set (0.00 sec)
```

Q 3.2 Create a query that display the number of customer that transact on the address in the tblcustomer.

Solution:

select C_Address, count(C_Name)"Total number of customer" from tblcustomer group by C_Address;

```
mysql> select C_Address, count(C_Name)"Total number of customer" from tblcustomer group by C_Address;
+-----+-----+
| C_Address | Total number of customer |
+-----+-----+
| aguisan   | 2 |
| talaban   | 1 |
| carabalan | 1 |
+-----+-----+
3 rows in set (0.00 sec)
```

Q 3.3 Display the details of each total price in tblpayment in the increasing order of price.

Solution:

select * from tblpayment order by P_Price;

```
mysql> select * from tblpayment order by P_Price;
+-----+-----+-----+-----+-----+-----+
| Pay_Id | P_Id | C_Id | Quantity | P_Price | Pay_Date_Order |
+-----+-----+-----+-----+-----+-----+
| 1006 | 1017 | 103 | 1 | 145.5 | 2021-03-04 |
| 1004 | 1019 | 102 | 1 | 245.52 | 2021-02-02 |
| 1001 | 1015 | 101 | 2 | 299 | 2021-02-01 |
| 1005 | 1018 | 103 | 5 | 417.225 | 2021-03-03 |
| 1003 | 1016 | 102 | 2 | 474 | 2021-02-02 |
| 1002 | 1016 | 101 | 3 | 711 | 2021-02-01 |
+-----+-----+-----+-----+-----+-----+
6 rows in set (0.00 sec)
```

4) SELECT STATEMENT USING Different Joins

Q 4.1 Create a query that display the Customer name and the total cost of purchased product from tbltransaction

Solution:

select C_Name, tbltransaction.P_Price from tblcustomer inner join tbltransaction on tblcustomer.C_Id = tbltransaction.C_Id;

```
mysql> select C_Name, tbltransaction.P_Price from tblcustomer inner join tbltransaction on tblcustomer.C_Id = tbltransaction.C_Id;
+-----+-----+
| C_Name | P_Price |
+-----+-----+
| chris grey | 299 |
| mark john | 474 |
| kim paterno | 145.5 |
+-----+-----+
3 rows in set (0.00 sec)
```

Q 4.2 Write a query that will display the delivery id, customer name in upper case format. Show also the list of delivery man and the date when the product will deliver.

Solution:

Select tbldelivery.D_Id, upper(C_Name), tbldelivery.Delivery_Man, tbldelivery.Date_of_Delivery from tblcustomer right join tbldelivery on tblcustomer.C_Id = tbldelivery.C_Id;

```
mysql> Select tbldelivery.D_Id, upper(C_Name), tbldelivery.Delivery_Man, tbldelivery.Date_of_Delivery from tblcustomer
right join tbldelivery on tblcustomer.C_Id = tbldelivery.C_Id;
+-----+-----+-----+-----+
| D_Id | upper(C_Name) | Delivery_Man | Date_of_Delivery |
+-----+-----+-----+-----+
| 200 | CHRIS GREY    | mario tanio  | 2021-02-04       |
| 201 | MARK JOHN    | bruno mars   | 2021-02-05       |
| 202 | KIM PATERNO  | mario tanio  | 2021-03-04       |
+-----+-----+-----+-----+
3 rows in set (0.00 sec)
```

Q 4.3 Write a query that will display the customer's name, customer number with the help of payment id.

Solution:

select C_Name, C_Contact_Number, tblpayment.Pay_Id from tblcustomer left join tblpayment on tblcustomer.C_Id= tblpayment.C_Id;

```
mysql> select C_Name, C_Contact_Number, tblpayment.Pay_Id from tblcustomer left join tblpayment on tblcustome
r.C_Id= tblpayment.C_Id;
+-----+-----+-----+
| C_Name      | C_Contact_Number | Pay_Id |
+-----+-----+-----+
| chris grey  | 9461243372      | 1002   |
| chris grey  | 9461243372      | 1001   |
| mark john   | 9368787217      | 1004   |
| mark john   | 9368787217      | 1003   |
| kim paterno | 9356484858,1234567872 | 1006   |
| kim paterno | 9356484858,1234567872 | 1005   |
| rex notar   | 8486486945      | NULL   |
+-----+-----+-----+
7 rows in set (0.00 sec)
```

5) Between, IN, Not Between, NOT IN

Q 5.1 Display details of records where date of order in tblpayment is between '2021-02-02' and '2021-03-04'.

Solution:

select * from tblpayment where Pay_Date_Order BETWEEN '2021-02-02' AND '2021-03-04';

```
mysql> select * from tblpayment where Pay_Date_Order BETWEEN '2021-02-02' AND '2021-03-04';
```

Pay_Id	P_Id	C_Id	Quantity	P_Price	Pay_Date_Order
1003	1016	102	2	474	2021-02-02
1004	1019	102	1	245.52	2021-02-02
1005	1018	103	5	417.225	2021-03-03
1006	1017	103	1	145.5	2021-03-04

```
4 rows in set (0.00 sec)
```

Q 5.2 Display details of records where date of delivery in tbldelivery is not between '2021-02-05' and '2021-03-04'.

Solution:

```
select * from tbldelivery where Date_of_Delivery NOT BETWEEN '2021-02-05' AND '2021-03-04';
```

```
mysql> select * from tbldelivery where Date_of_Delivery NOT BETWEEN '2021-02-05' AND '2021-03-04';
```

D_Id	C_Id	T_Id	P_Name	Date_of_Delivery	Delivery_Man
200	101	10101	round gallon	2021-02-04	mario tanio

```
1 row in set (0.00 sec)
```