

# **DBMS - MINI PROJECT**

**TITLE: WATER SUPPLY MANAGEMENT SYSTEM**

Submitted By:

Name : JASAL SINGH SADERA

SRN : PES1UG20CS175

V Semester Section \_ : 5C

# **ABSTRACT**

## **DESCRIPTION OF PROJECT:**

The water supply management system is a simple database management project with a normal frontend which will let the water supply management companies to handle and manage the supply of water in containers throughout their areas efficiently and in a systematic manner. For this project I used MySQL for database operations which was connected to a basic frontend using PHP, HTML, CSS and JAVASCRIPT. The user of this project will be able to execute the basic CRUD operations of data entries into the MYSQL database and also will be able to execute certain aggregate and functional operations.

## **SCOPE OF THE PROJECT:**

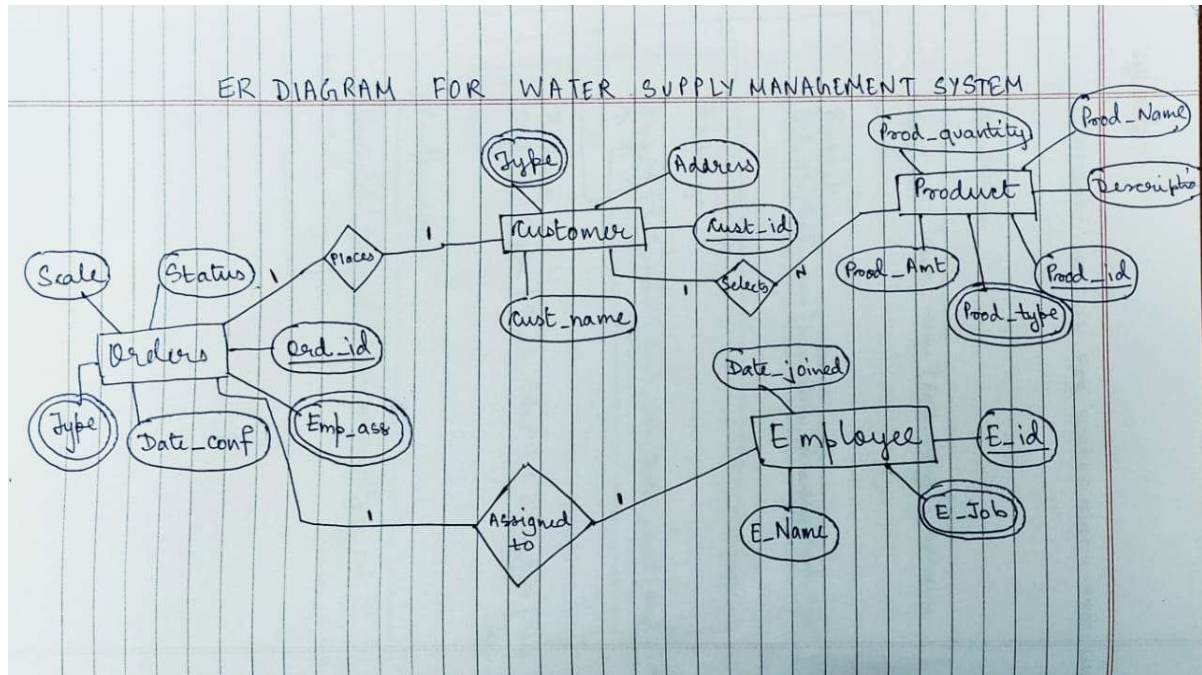
As we are all aware of the fact that in today's world, the scarcity and shortage of water supply is increasing day by day where people are finding it hard to get the normal required supply of water for their basic needs. Thus in order to help them we can use this efficient web-based application which will help them to order optimum amount of water as required by them for their needs. Also since many of the areas which live in poorly maintained municipal areas will be able to get a clean and faster supply of water resource just by placing their order through the project.

## **Tools Used:**

**FRONTEND TOOLS:PHP, JAVASCRIPT(Jquery and Ajax),CSS,HTML**

**BACKEND TOOLS USED:MYSQL(For database)**

## ER Diagram of the project



## Relational Schema of the project

### RELATIONAL SCHEMA FOR WATER SUPPLY MANAGEMENT

Customer

<u>Cust_id</u>	Cust_Name	Cust_addrtype	Address
----------------	-----------	---------------	---------

Product

<u>Prod_id</u>	Prod_name	Description	Quantity	Prod-type
----------------	-----------	-------------	----------	-----------

Employee

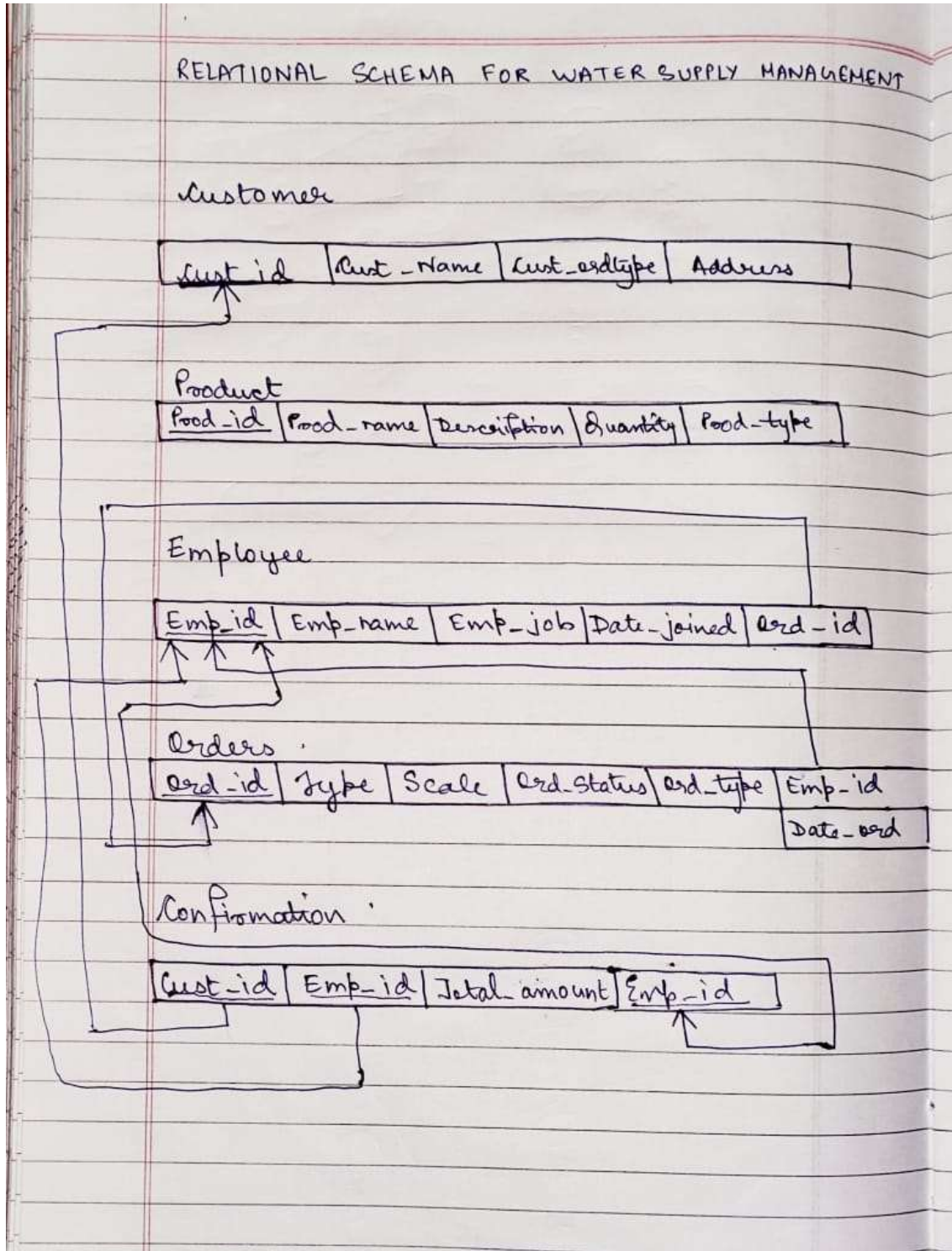
<u>Emp_id</u>	Emp_name	Emp_job	Date_joined	<u>Ord_id</u>
---------------	----------	---------	-------------	---------------

Orders

<u>Ord_id</u>	Type	Scale	Ord-status	Ord-type	Emp_id	Date-ord
---------------	------	-------	------------	----------	--------	----------

Confirmation

<u>Cust-id</u>	<u>Emp-id</u>	Total amount	<u>Emp-id</u>
----------------	---------------	--------------	---------------



# DDL statements - Building the database

## 1. Creating the table 'sales':

```
CREATE TABLE `sales` (  
  `id` int(30) NOT NULL,  
  `customer_name` text NOT NULL,  
  `type` tinyint(4) NOT NULL DEFAULT 1 COMMENT '1 = walk-in, 2 = for  
delivery',  
  `delivery_address` text NOT NULL,  
  `amount` float NOT NULL,  
  `status` tinyint(1) NOT NULL DEFAULT 0 COMMENT '0= Unpaid, 1=Paid',  
  `date_created` datetime NOT NULL DEFAULT current_timestamp(),  
  `date_updated` datetime DEFAULT NULL ON UPDATE current_timestamp()  
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;
```

## 2. Creating the table 'sales\_items':

```
CREATE TABLE `sales_items` (  
  `id` int(30) NOT NULL,  
  `sales_id` int(30) NOT NULL,  
  `jar_type_id` int(30) NOT NULL,  
  `quantity` float NOT NULL,  
  `price` float NOT NULL,  
  `total_amount` float NOT NULL  
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;
```

## 3. Creating the table 'system\_info':

```
CREATE TABLE `system_info` (  
  `id` int(30) NOT NULL,  
  `meta_field` text NOT NULL,  
  `meta_value` text NOT NULL  
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;
```

## 4.Creating the table “jar\_types”:

```
CREATE TABLE `jar_types` (  
  `id` int(30) NOT NULL,  
  `name` text NOT NULL,  
  `description` text NOT NULL,  
  `pricing` float NOT NULL,  
  `date_created` datetime NOT NULL DEFAULT current_timestamp(),  
  `date_updated` datetime DEFAULT NULL ON UPDATE current_timestamp()  
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;
```

## 5.Creating the table ‘emp’:

```
CREATE TABLE `emp` (  
  `id` int(30) NOT NULL,  
  `emp_name` text NOT NULL,  
  `type` tinyint(4) NOT NULL DEFAULT 1 COMMENT '1 = Managing goods, 2 =  
Delivery_staff',  
  `delivery_address` text NOT NULL,  
  `date_created` datetime NOT NULL DEFAULT current_timestamp(),  
  `date_updated` datetime DEFAULT NULL ON UPDATE current_timestamp()  
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;
```

## 6.Making jar\_id as the primary key of the jar\_Types tables:

```
ALTER TABLE `jar_types`  
  ADD PRIMARY KEY (`jar_id`);
```

## 7.Making sales\_id as the primary key of the ‘sales’ tables:

```
ALTER TABLE `sales`  
  ADD PRIMARY KEY (`sales_id`);
```

## 8. Adding auto increment for the following tables – sales, jar\_types, sales\_items

```
ALTER TABLE `jar_types`  
  MODIFY `id` int(30) NOT NULL AUTO_INCREMENT, AUTO_INCREMENT=5;  
  
--  
-- AUTO_INCREMENT for table `sales`  
--  
ALTER TABLE `sales`  
  MODIFY `id` int(30) NOT NULL AUTO_INCREMENT, AUTO_INCREMENT=4;  
  
--  
-- AUTO_INCREMENT for table `sales_items`  
--  
ALTER TABLE `sales_items`  
  MODIFY `id` int(30) NOT NULL AUTO_INCREMENT, AUTO_INCREMENT=9;
```

## 9. Adding Foreign key 'sales\_id' to sales\_items which is primary key of the 'sales' table

```
ALTER TABLE `sales_items`  
  ADD CONSTRAINT `sales_items_ibfk_1` FOREIGN KEY (`sales_id`) REFERENCES  
  `sales` (`id`) ON DELETE CASCADE;  
COMMIT;
```

## 10. Adding Foreign key 'sales\_items\_id' to emp table which is primary key of the 'sales\_items' table

```
ALTER TABLE `emp`  
  ADD CONSTRAINT `emp_ibfk_1` FOREIGN KEY (`sales_items_id`) REFERENCES  
  `sales` (`id`) ON DELETE CASCADE;  
COMMIT;
```

# POPULATING THE DATABASE

## (All insertion commands)

### 1.Inserting the default values of the jar\_types table

```
INSERT INTO `jar_types` (`id`, `name`, `description`, `pricing`,  
`date_created`, `date_updated`) VALUES  
(1, 'Slim Container with cap and faucet', '<span style=\"color: rgb(0, 0, 0);  
font-family: \"Open Sans\", Arial, sans-serif; font-size: 14px; text-align:  
justify;\">Integer a risus enim. Mauris justo erat, tempus eu mauris sed,  
scelerisque tincidunt diam.</span>', 30, '2021-08-14 14:29:40', '2021-08-14  
14:32:00'),  
(2, 'Round Container with Cap', '<p><span style=\"color: rgb(0, 0, 0); font-  
family: &quot;Open Sans&quot;; Arial, sans-serif; font-size: 14px; text-align:  
justify;\">Nunc a massa id ligula varius convallis in non augue. Sed feugiat  
blandit mattis.</span><br></p>', 30, '2021-08-14 14:32:18', NULL);
```

### 2.Inserting the default values into the ‘sales’ table:

```
INSERT INTO `sales` (`id`, `customer_name`, `type`, `delivery_address`,  
`amount`, `status`, `date_created`, `date_updated`) VALUES  
(1, 'John Smith', 1, '', 360, 1, '2021-08-14 15:41:36', '2021-08-14  
15:50:29'),  
(2, 'Claire Blake', 2, 'Sample Address', 150, 1, '2021-08-14 15:51:44', '2021-  
08-14 15:55:17');
```

### 3.Inserting the default values into the ‘sales\_item’ table:

```
INSERT INTO `sales_items` (`id`, `sales_id`, `jar_type_id`, `quantity`,  
`price`, `total_amount`) VALUES  
(3, 1, 1, 10, 30, 300),  
(4, 1, 2, 2, 30, 60),  
(7, 2, 2, 5, 30, 150);
```



#### 4.Inserting the default values into the ‘system\_info’ table.

```
INSERT INTO `system_info` (`id`, `meta_field`, `meta_value`) VALUES
(1, 'name', 'Simple Water Refilling Management System'),
(6, 'short_name', 'Water Refilling System - PHP'),
(11, 'logo', 'uploads/1628916900_water_refilling.png'),
(13, 'user_avatar', 'uploads/user_avatar.jpg'),
(14, 'cover', 'uploads/1626249540_dark-bg.jpg');
```

#### 5.Inserting the extacted ‘\$data’ values into the table ‘sales\_item’

```
$sql2 = $this->conn->query("INSERT INTO `sales_items`
(`sales_id`,`jar_type_id`,`quantity`,`price`,`total_amount`) VALUES {$data}
");
```

#### 6. Inserting the extacted ‘\$data’ values into the table ‘emp’.

```
emp`(`emp_id`,`emp_name`,`type`,`delivery_address`,`date_updated`) VALUES
{$data} " );
```

#### 7. Inserting the a new entry of container type into the ‘jar\_types’ table.

```
$sql = "INSERT INTO `jar_types` set {$data}";
```

# QUERIES OF THE PROJECT

## JOIN QUERIES:

1.The “inner join” command is used in order to give a preview of the quantity, jar type, price and total amount of the customer order. The jar\_type attribute is taken from the jar\_types table and rest from sales\_items.

```
Command: $qry2 = $conn->query("SELECT i.*,j.name FROM `sales_items` i inner join `jar_types` j on j.id = i.jar_type_id where i.sales_id = '{$id}' order by id asc ");
```

## SCREENSHOT:

The screenshot shows the 'Simple Water Refilling Management System - Admin' interface. On the left is a sidebar with navigation links: Dashboard, Customer Sales, Employee portal, Product details, Orders, SQL QUERIES, Settings, and System Info. The main content area displays a form for a delivery address. At the top, there's a text input for the name (filled with 'Henry') and a dropdown for 'For Delivery'. Below this is a 'Delivery Address' section with a text input (filled with 'PES UNIVERSITY'). Further down, there's a 'Jar Type' dropdown and a 'Quantity' input field, followed by a '+ Add' button. At the bottom, a table displays the current order items.

	QTY	Jar Type	Price	Total Amount
	20	Slim Container with cap and faucet	30	600
Total				600

2.I have used the “natural join” to execute and display the table mentioned below to get the customer details and the order details the customer has bought. The tables I have joined are ‘sales’ and ‘jar\_types’.

## Command:

```
$sqry = $conn->query("SELECT i.*,j.name FROM `sales` i natural join jar_types j on j.id = i.jar_type_id where i.sales_id = '{$row['id']}' ");
```

## SCREENSHOT :

The screenshot shows the 'Simple Water Refilling Management System - Admin' interface. The left sidebar contains navigation links: Dashboard, Customer Sales, Employee portal, Product details, Orders (highlighted), SQL QUERIES, Settings, and System Info. The main content area displays a 'Sales Report' for the period from 21-11-2022 to 26-11-2022. The report includes a table with columns: #, Date, Customer, Type, Details, and Total Amount. The data shows three orders: Order 1 (Delivered, 12 Round Containers, 360 total), Order 2 (Waik-In, 10 Slim Containers and 2 Round Containers, 360 total), and Order 3 (Delivered, 5 Round Containers, 150 total). A total of 870 is shown at the bottom.

#	Date	Customer	Type	Details	Total Amount
1	2022-11-24	Mark	Delivered	12 Round Container with Cap	360
2	2022-11-22	John Smith	Waik-In	10 Slim Container with cap and faucet 2 Round Container with Cap	360
3	2022-11-22	Claire Blake	Delivered	5 Round Container with Cap	150
Total					870

## AGGREGATE QUERIES:

1. This command will be used to calculate the total amount of the order of customer before inserting the customer entry into the database.

### Command:

```
$item_count = $conn->query("SELECT sum(quantity) as total FROM sales_items  
where sales_id = '{$row['id']}'")->fetch_array()['total'];
```

## Screenshot:

The screenshot shows the 'Customer Sales' section of the 'Simple Water Refilling Management System - Admin' interface. The left sidebar contains navigation links: Customer Sales (highlighted), Employee portal, Product details, Orders, SQL QUERIES, Settings, and System Info. The main content area displays a form to add a new order. The form includes a text input for 'Customer Name' (containing 'pes university'), a dropdown for 'Jar Type', and a text input for 'Quantity'. A '+ Add' button is present. Below the form is a table showing the current order items and their total amount.

	QTY	Jar Type	Price	Total Amount
<input type="checkbox"/>	12	Round Container with Cap	30	360
<input type="checkbox"/>	5	Slim Container with cap and faucet	30	150
Total				510

2.This function will be used to show the total number of entries in the “jar\_types” table .

Command:

```
$sql= select count(id) as number_of_jars from jar_types;
```

Screenshot:

```
MariaDB [water_refilling_db]> select count(id) as number_of_jars from jar_types;
+-----+
| number_of_jars |
+-----+
|                2 |
+-----+
1 row in set (0.001 sec)
```

List of Jar Types & Pricing [+ Create New](#)

Show  entries

Search:

#	Date Created	Name	Description	Price	Action
1	2021-08-14 14:32:18	Round Container with Cap	hello	25	Action ▼
2	2021-08-14 14:29:40	Slim Container with cap and faucet	Integer a risus enim. Mauris justo erat, tempus eu mauris sed, scelerisque tincidunt diam. Na...	30	Action ▼

Showing 1 to 2 of 2 entries

Previous **1** Next

## SET QUERIES:

1.The set function “UNION” will be used to get the result of the customer names and the jar types they have selected.

**Command:** select customer\_name from sales

-> union

-> select name from jar\_types;

## Screenshot:

```
MariaDB [water_refilling_db]> select customer_name from sales
-> union
-> select name from jar_types;
+-----+
| customer_name |
+-----+
| John Smith    |
| Claire Blake  |
| Slim Container with cap and faucet |
| Round Container with Cap |
+-----+
4 rows in set (0.016 sec)
```

2.This command uses the set operation “EXCEPT” which will display the details which are present in emp table and not in jar\_types table.

**Command:** > select \* from emp

-> except

-> select \* from jar\_types;

## Screenshot:

```
MariaDB [water_refilling_db]> select * from emp
-> except
-> select * from jar_types;
+----+-----+-----+-----+-----+-----+
| id | emp_name | type | delivery_address | date_created | date_updated |
+----+-----+-----+-----+-----+-----+
| 1  | Rohit    | 1    | pes univ         | 2022-11-22 15:41:36 | 2022-11-22 15:50:29 |
+----+-----+-----+-----+-----+-----+
1 row in set (0.024 sec)
```

## VIEW QUERIES:

1.The virtual table “customer\_walk\_in” will display all the customers and their bill amount who has their delivery type as Walk-in:

Command:

```
CREATE VIEW customer_walk_in AS
-> SELECT customer_name,amount
-> FROM sales
-> WHERE type=1;
```

Screenshot:

```
MariaDB [water_refilling_db]> select * from customer_walk_in;
+-----+-----+
| customer_name | amount |
+-----+-----+
| John Smith    | 360    |
+-----+-----+
1 row in set (0.020 sec)

MariaDB [water_refilling_db]> |
```

## TRIGGER QUERIES:

1.The trigger that I have created here will insert the column “scale” of the sales\_items tables to “bulk order” if the bill amount of the customer is greater than the mentioned values in the trigger function:

Command:

```
create trigger new_scale
-> AFTER INSERT ON sales
-> for each row
-> begin
-> if new.amount < 200 then
-> insert into sales_items
values(id,sales_id,jar_type_id,quantity,price,total_amount,'Bulk order');
-> end;
```

## Screenshots:

### Sales\_items

```
MariaDB [water_refilling_db]> select * from sales_items;
```

id	sales_id	jar_type_id	quantity	price	total_amount	scale
3	1	1	10	30	300	Bulk order
4	1	2	2	30	60	Normal order
13	2	2	5	30	150	Normal order

3 rows in set (0.001 sec)

### Sales:

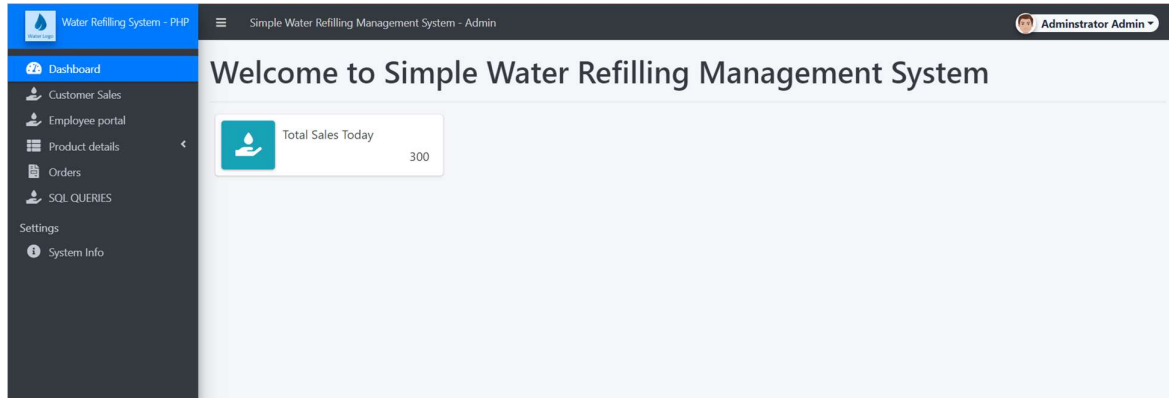
```
MariaDB [water_refilling_db]> select * from sales;
```

id	customer_name	type	delivery_address	amount	status	date_created	date_updated
1	John Smith	1		60	1	2022-11-22 15:41:36	2022-11-26 00:52:57
2	Claire Blake	2	Sample Address	150	0	2022-11-22 11:41:36	2022-11-23 18:31:36
16	Henry	1		300	0	2022-11-26 00:51:47	NULL

3 rows in set (0.000 sec)

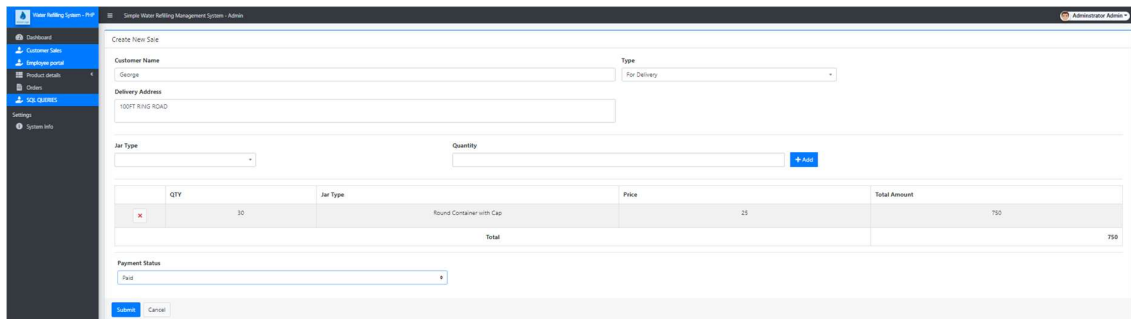
# FRONTEND OF THE PROJECT:

## Front page of the project:

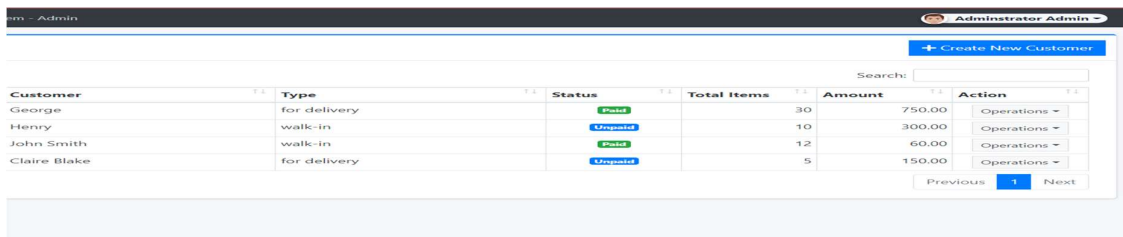


1. Addition, Modification and Deletion of records from any chosen table

1>CREATE FUNCTION:



2>READ FUNCTION:





### 3>UPDATE FUNCTION:

System - Admin Administrator Admin

+ Create New Customer

Search:

Customer	Type	Status	Total Items	Amount	Action
George	for delivery	Unpaid	30	750.00	Operations
Henry	walk-in	Unpaid	10	300.00	Update
John Smith	walk-in	Paid	12	60.00	Delete
Claire Blake	for delivery	Unpaid	5	150.00	Operations

Previous 1 Next

### 4>DELETE FUNCTION:

System - Admin Administrator Admin

+ Create New Customer

Search:

Customer	Type	Status	Total Items	Amount	Action
Henry	walk-in	Unpaid	10	300.00	Operations
John Smith	walk-in	Paid	12	60.00	Operations
Claire Blake	for delivery	Unpaid	5	150.00	Operations

Previous 1 Next

2. There should be a window to accept and run any SQL statement and display the result

### SCREENSHOT:

Water Refilling System - PHP Single Water Refilling Management System - Admin Administrator Admin

Enter the sql query here

SELECT \* FROM jar\_types

#	Date Created	Name	Description	Price
1	2021-08-14 14:32:18	Round Container with Cap	hello	25
2	2021-08-14 14:29:40	Slim Container with cap and faucet	Integer a risus enim. Mauris justo erat, tempus eu mauris sed, scelerisque tincidunt diam. Nam eget augue aliquam, commodo ligula consequat, maximus tellus. Suspendisse elit eros, pellentesque nec enim non, tincidunt pharetra...	30

