CE3109/IT3210/ITX3006 Database Management System

By

Mr. Sattavath Phitsavath	6228009
Mr. Rajay	6018127
Ms. Ashika Yeasmin Angkur	6228006

DBMS PROJECT REPORT ON
“Glass House Application”

Instructor:

Ms. Rachsuda Setthawong

Assumption University
Faculty of Engineering

Thailand
Semester 1/2021

Summary

Our store is to sustain agriculture under the rallying twice as much as food and plant using half as many resources. The ambitious goal has involved considerable innovation by the farmers and gardeners. Glasshouse production has evolved and become highly efficient through innovations in Mechanization, climate control, Greenhouse Design, Energy use, lighting Optimization, pest control, carbon dioxide enrichment, fertilization, and water use.

Therefore, to achieve this as close to real world application, In this project, we used web languages which are HTML, CSS, PHP, SQL, JavaScript, AJAX and Bootstrap to finish the project. Moreover, we used XAMPP for database connection to local and we store data and update data from local database. We include some function like button and some action in our project using PHP and JavaScript language. Our project is based on Website, and it is also good for the environment and human being.

Introduction

Nowadays, around the world there is increasing pollution and facing a lot of environmental problems. So due to that problem, we humans are facing food problems and not getting enough fresh air. We consider that kind of problem, so we choose to do a glasshouse store for our term project. Glasshouse has unique properties of high light transmission and attracts solar energy, trapping warmth re-emitted from the soil. First, we did research about glasshouse stores, and we discussed.

There are three stakeholders who are admin, customer, and retailer. The gardeners analyze and control the glasshouse key factors which are fertilization, water use, climate control, and temperature and report to the admin. Admin can view and edit the gardeners list and customer list and details.

Scenario

- **Employees** are identified by their ***employee_id*** *valu*es. The store's admin stores the employee's name, role, birthdate, and salary of each employee. The store also has employees' joining year, thus, the length of employment years. Employees can view the customer/retailer transaction order.

- **The glasshouse records** are identified by their glasshouse_id. Each glasshouse may have one or more types of plants and each plant can be in zero or more glasshouses. The employees keep track of the temperature 3 times a day then the admin stores the average temperature, average humidity level, fertilizer status and pH of soil.

- **Stocks** are identified by their product_id . It stores the product's name, product category, price per unit and the quantity of the availability of flowers, fruits, and gardening tools, which can be further viewed by the customers and retailer from the website, furthermore their transaction in the stock directly reaches the transaction entity.

- **Customers** are identified by their customer_id values. The store stores each customer's name, nationality, occupation, birthdate, phone number, email address, and the address of customers which include state, city, and zip code. A customer may view their transaction record, and order the plants, or gardening tool (The stock).

- **Retailers** are identified by their retailer_id. The admin stores the store's name, the name of the owner, phone number, email address, and the store's location. Retailers can view the product through our website, also their transaction record, and check information about exclusive discounts that are for retailers.

- **Transactions** are identified by its transaction_id and product_id values. The website stores each customer transaction record which are the quantity that they brought, the discount, transaction_date, and the total price. Transaction can be viewed by all stakeholders which are employees, customers and retailers.

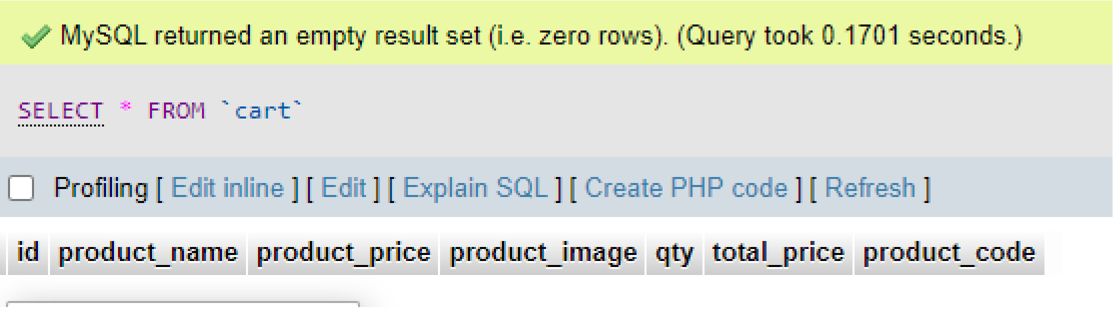
The sql Codes for retrieving the data from the database (in phpMyAdmin)

1) Table 'cart'
The commands used to create the table:

```
CREATE TABLE `cart` (
  `id` int(11) NOT NULL,
  `product_name` varchar(100) NOT NULL,
  `product_price` varchar(50) NOT NULL,
  `product_image` varchar(255) NOT NULL,
  `qty` int(10) NOT NULL,
  `total_price` varchar(100) NOT NULL,
  `product_code` varchar(10) NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
```

The sql commands (in bits)for storing the cart data from the app to the DB:

```
    $stmt = $conn->prepare('SELECT product_code FROM cart WHERE product_code=?');
    $stmt->bind_param('s',$pcode);
    $stmt->execute();
    $res = $stmt->get_result();
    $r = $res->fetch_assoc();
    $code = $r['product_code'] ?? '';

    if (!$code) {
      $query = $conn->prepare('INSERT INTO cart
(product_name,product_price,product_image,qty,total_price,product_code) VALUES
(?,?,?,?,?,?)');


    }
  }

  // Get no.of items available in the cart table
  {
    $stmt = $conn->prepare('SELECT * FROM cart');


  // Remove single items from cart
  '];

    $stmt = $conn->prepare('DELETE FROM cart WHERE id=?');
              hea     }

  // Remove all items at once from cart
  if (isset($_GET['clear'])) {
    $stmt = $conn->prepare('DELETE FROM cart');
    $stmt->execute();
    $_SESSION['showAlert'] = 'block';
    $_SESSION['message'] = 'All Item removed from the cart!';
    header('location:cart.php');
  }

  // Set total price of the product in the cart table

    $stmt = $conn->prepare('UPDATE cart SET qty=?, total_price=? WHERE id=?');
           }
```



✓ MySQL returned an empty result set (i.e. zero rows). (Query took 0.1701 seconds.)

```
SELECT * FROM `cart`
```

□ Profiling [Edit inline] [Edit] [Explain SQL] [Create PHP code] [Refresh]

id	product_name	product_price	product_image	qty	total_price	product_code

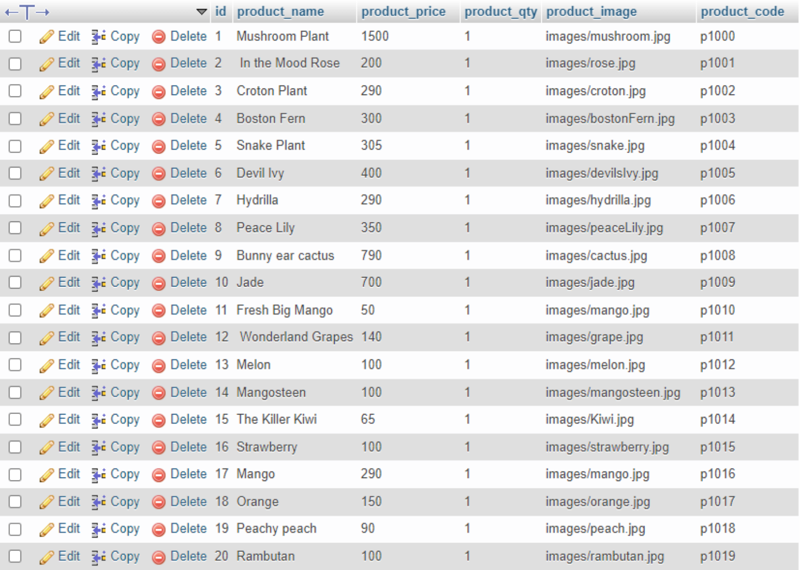
2)Table ‘Product”

The commands used to create the table:

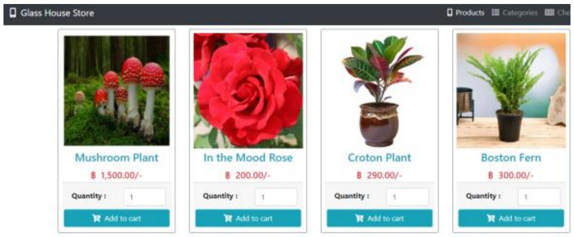
```
CREATE TABLE `product` (
  `id` int(11) NOT NULL,
  `product_name` varchar(255) NOT NULL,
  `product_price` varchar(100) NOT NULL,
  `product_qty` int(11) NOT NULL DEFAULT 1,
  `product_image` varchar(255) NOT NULL,
  `product_code` varchar(50) NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
```

The commands used to insert values the table:

```
INSERT INTO `product` (`id`, `product_name`, `product_price`, `product_qty`, `product_image`, `product_code`) VALUES
(1, 'Mushroom Plant', '1500', 1, 'images/mushroom.jpg', 'p1000'),
(2, ' "In the Mood" Rose', '200', 1, 'images/rose.jpg', 'p1001'),
(3, 'Croton Plant', '290', 1, 'images/croton.jpg', 'p1002'),
(4, 'Boston Fern', '300', 1, 'images/bostonFern.jpg', 'p1003'),
(5, 'Snake Plant', '305', 1, 'images/snake.jpg', 'p1004'),
(6, 'Devil ivy', '400', 1, 'images/devilsIvy.jpg', 'p1005'),
(7, 'Hydrilla', '290', 1, 'images/hydrilla.jpg', 'p1006'),
(8, 'Peace Lily', '350', 1, 'images/peaceLily.jpg', 'p1007'),
(9, 'Bunny ear cactus', '790', 1, 'images/cactus.jpg', 'p1008'),
(10, 'Jade', '700', 1, 'images/jade.jpg', 'p1009'),

(11, 'Fresh Big Mango', '50', 1, 'images/mango.jpg', 'p1010'),
(12, ' Wonderland Grapes', '140', 1, 'images/grape.jpg', 'p1011'),
(13, 'Melon', '100', 1, 'images/melon.jpg', 'p1012'),
(14, 'Mangosteen', '100', 1, 'images/mangosteen.jpg', 'p1013'),
(15, 'The Killer Kiwi', '65', 1, 'images/Kiwi.jpg', 'p1014'),
(16, 'Strawberry', '100', 1, 'images/strawberry.jpg', 'p1015'),
(17, 'Mango', '290', 1, 'images/mango.jpg', 'p1016'),
(18, 'Orange', '150', 1, 'images/orange.jpg', 'p1017'),
(19, 'Peachy peach', '90', 1, 'images/peach.jpg', 'p1018'),
(20, 'Rambutan', '100', 1, 'images/rambutan.jpg', 'p1019'),


(21, 'The super Axe', '350', 1, 'images/axe.jpg', 'p1020'),
(22, ' Hedge shears', '240', 1, 'images/shears.jpg', 'p1021'),
(23, 'Gardening Fork', '100', 1, 'images/gardeningFork.jpg', 'p1022'),
(24, 'Sickle', '90', 1, 'images/sickle.jpg', 'p1023'),
(25, 'Pruning Saw', '85', 1, 'images/saw.jpg', 'p1024'),
(26, 'The Shovel', '130', 1, 'images/shovel.jpg', 'p1025'),
(27, 'Fertilizer', '190', 1, 'images/fertilizer.jpg', 'p1026');
```



←T→ ▼	id	product_name	product_price	product_qty	product_image	product_code
□ Edit Copy Delete	1	Mushroom Plant	1500	1	images/mushroom.jpg	p1000
□ Edit Copy Delete	2	In the Mood Rose	200	1	images/rose.jpg	p1001
□ Edit Copy Delete	3	Croton Plant	290	1	images/croton.jpg	p1002
□ Edit Copy Delete	4	Boston Fern	300	1	images/bostonFern.jpg	p1003
□ Edit Copy Delete	5	Snake Plant	305	1	images/snake.jpg	p1004
□ Edit Copy Delete	6	Devil Ivy	400	1	images/devilsIvy.jpg	p1005
□ Edit Copy Delete	7	Hydrilla	290	1	images/hydrilla.jpg	p1006
□ Edit Copy Delete	8	Peace Lily	350	1	images/peaceLily.jpg	p1007
□ Edit Copy Delete	9	Bunny ear cactus	790	1	images/cactus.jpg	p1008
□ Edit Copy Delete	10	Jade	700	1	images/jade.jpg	p1009
□ Edit Copy Delete	11	Fresh Big Mango	50	1	images/mango.jpg	p1010
□ Edit Copy Delete	12	Wonderland Grapes	140	1	images/grape.jpg	p1011
□ Edit Copy Delete	13	Melon	100	1	images/melon.jpg	p1012
□ Edit Copy Delete	14	Mangosteen	100	1	images/mangosteen.jpg	p1013
□ Edit Copy Delete	15	The Killer Kiwi	65	1	images/Kiwi.jpg	p1014
□ Edit Copy Delete	16	Strawberry	100	1	images/strawberry.jpg	p1015
□ Edit Copy Delete	17	Mango	290	1	images/mango.jpg	p1016
□ Edit Copy Delete	18	Orange	150	1	images/orange.jpg	p1017
□ Edit Copy Delete	19	Peachy peach	90	1	images/peach.jpg	p1018
□ Edit Copy Delete	20	Rambutan	100	1	images/rambutan.jpg	p1019



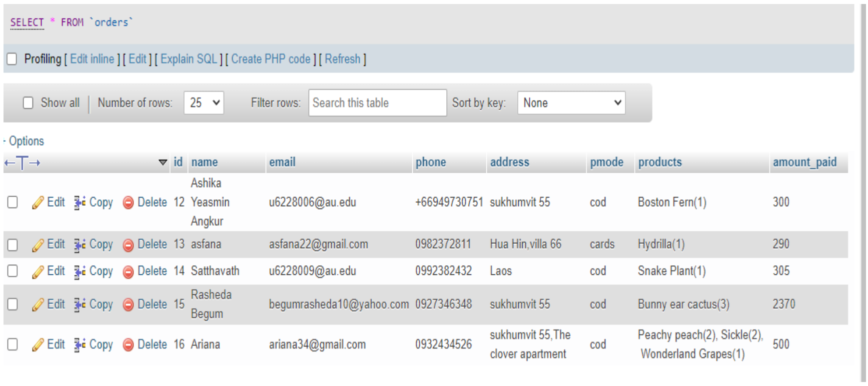
The sql commands (in bits)for storing the cart data from the DB to the App:

```
 <?php
        require 'config.php';
        $stmt = $conn->prepare('SELECT * FROM cart');
        $stmt->execute();
        $result = $stmt->get_result();
        $grand_total = 0;
        while ($row = $result->fetch_assoc()):
       ?>
```

3)Table ‘ orders’

The commands used to create the table:

```
CREATE TABLE `orders` (
  `id` int(11) NOT NULL,
  `name` varchar(100) NOT NULL,
  `email` varchar(100) NOT NULL,
  `phone` varchar(20) NOT NULL,
  `address` varchar(255) NOT NULL,
  `pmode` varchar(50) NOT NULL,
  `products` varchar(255) NOT NULL,
  `amount_paid` varchar(100) NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
```



```
SELECT * FROM `orders`
```

□ Profiling [Edit inline] [Edit] [Explain SQL] [Create PHP code] [Refresh]
□ Show all
Number of rows: 25
Filter rows: Search this table
Sort by key: None
- Options

			id	name	email	phone	address	pmode	products	amount_paid
□	Edit	Copy	Delete	12	Ashika Yeasmin Angkur	u6228006@au.edu	+66949730751	sukhumvit 55	cod	Boston Fern(1)	300
□	Edit	Copy	Delete	13	asfana	asfana22@gmail.com	0982372811	Hua Hin,villa 66	cards	Hydrilla(1)	290
□	Edit	Copy	Delete	14	Satthavath	u6228009@au.edu	0992382432	Laos	cod	Snake Plant(1)	305
□	Edit	Copy	Delete	15	Rasheda Begum	begumrasheda10@yahoo.com	0927346348	sukhumvit 55	cod	Bunny ear cactus(3)	2370
□	Edit	Copy	Delete	16	Ariana	ariana34@gmail.com	0932434526	sukhumvit 55,The clover apartment	cod	Peachy peach(2), Sickle(2), Wonderland Grapes(1)	500

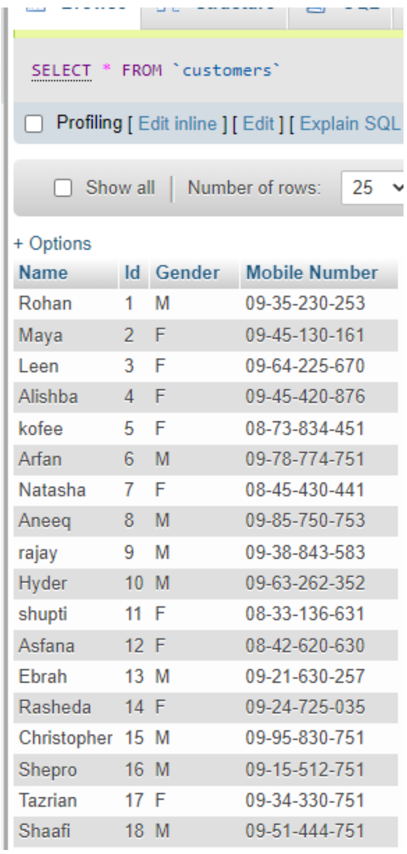
The sql commands (in bits)for storing the cart data from the App to the DB :

```
<?php
$conn = mysqli_connect("localhost", "root", "", "shopping_cart");
// Check connection
if ($conn->connect_error) {
die("Connection failed: " . $conn->connect_error);
}
$sql = "SELECT * FROM orders
ORDER by Id  ASC;";
```

4)Table ‘customers’

The commands used to insert values the table:

```
INSERT INTO customers (`Name`, `Id`,  `Gender`, `Mobile Number`)
VALUES ('Rohan', '1', 'M','09-35-230-253'),('Maya', '2', 'F','09-45-130-161'),
('Leen', '3', 'F','09-64-225-670'),('Alishba', '4', 'F','09-45-420-876'),
 ('kofee', '5', 'F','08-73-834-451'),('Arfan', '6', 'M','09-78-774-751'),
 ('Natasha', '7', 'F','08-45-430-441'),('Aneeq', '8', 'M','09-85-750-753'),

 ('rajay', '9', 'M','09-38-843-583'),('Hyder', '10', 'M','09-63-262-352'),
('shupti', '11', 'F','08-33-136-631'),('Asfana', '12', 'F','08-42-620-630'),
 ('Ebrah', '13', 'M','09-21-630-257'),('Rasheda', '14', 'F','09-24-725-035'),


 ('Christopher', '15', 'M','09-95-830-751'),('Shepro', '16', 'M','09-15-512-751'),
 ('Tazrian', '17', 'F','09-34-330-751'),('Shaafi', '18', 'M','09-51-444-751'),
 ('Tasfia', '19', 'F','08-11-132-751'),('Neha', '20', 'F','09-44-034-751');
```



```
SELECT * FROM `customers`
```

□ Profiling [Edit inline] [Edit] [Explain SQL
□ Show all | Number of rows: 25
+ Options

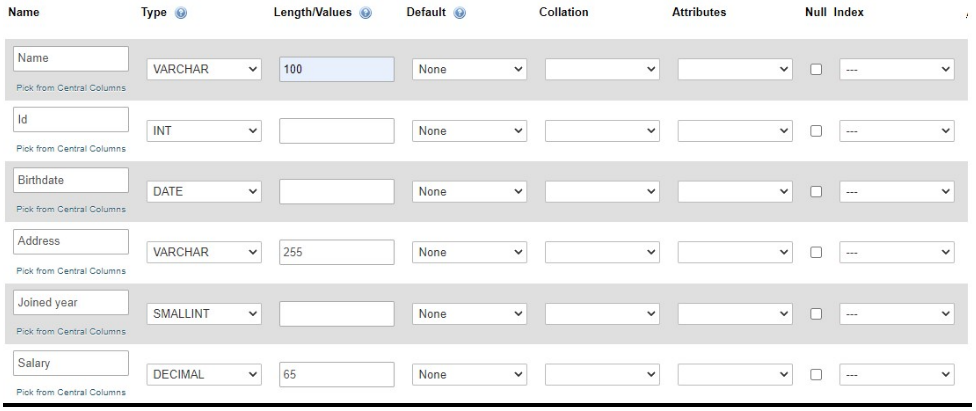
Name	Id	Gender	Mobile Number
Rohan	1	M	09-35-230-253
Maya	2	F	09-45-130-161
Leen	3	F	09-64-225-670
Alishba	4	F	09-45-420-876
kofee	5	F	08-73-834-451
Arfan	6	M	09-78-774-751
Natasha	7	F	08-45-430-441
Aneeq	8	M	09-85-750-753
rajay	9	M	09-38-843-583
Hyder	10	M	09-63-262-352
shupti	11	F	08-33-136-631
Asfana	12	F	08-42-620-630
Ebrah	13	M	09-21-630-257
Rasheda	14	F	09-24-725-035
Christopher	15	M	09-95-830-751
Shepro	16	M	09-15-512-751
Tazrian	17	F	09-34-330-751
Shaafi	18	M	09-51-444-751

The sql commands (in bits)for storing the data from the DB to the App:

```
<?php
$conn = mysqli_connect("localhost", "root", "", "list");
// Check connection
if ($conn->connect_error) {
die("Connection failed: " . $conn->connect_error);
}
$sql = "SELECT * FROM customers
ORDER by Id  ASC;";
```

5)Table ‘gardeners’

The specifications used to create the table in phpMyAdmin:



Name	Type	Length/Values	Default	Collation	Attributes	Null	Index
Name Pick from Central Columns	VARCHAR	100	None			□	---
Id Pick from Central Columns	INT		None			□	---
Birthdate Pick from Central Columns	DATE		None			□	---
Address Pick from Central Columns	VARCHAR	255	None			□	---
Joined year Pick from Central Columns	SMALLINT		None			□	---
Salary Pick from Central Columns	DECIMAL	65	None			□	---

The commands used to insert values the table:

```
INSERT INTO `gardeners`
(`Name`, `Id`, `Birthdate`, `Address`, `Joined year`, `Salary`)
VALUES ('Anas', '1001', '1999-03-29', 'Grilled Town', '2020', '24000'),
('Asfana', '1002', '2005-07-04', 'Sukhumvit 55,thonglor soi 18,
Tower B', '2021', '19000')
,('Alicia', '1003', '1987-03-29', 'Sukhumvit soi 71,Nich apartment', '2019', '27000'),
('Amna', '1004', '1999-07-04', 'pattaya city,gracious apartment', '20120', '28000'),
('Anika', '1005', '1987-03-29', 'Ramkamhaeng,huamak', '2018', '27000'),
('Pranto', '1006', '1999-08-09', 'Onnut Road,streetsvalley', '2020', '14000'),
('Deepti', '1007', '2000-03-29', 'Sukhumvit 55,clover apartment', '2019', '27000'),
('Preeyarat', '1008', '20002-08-20', 'Taipeng tower', '2019', '8000'),
('Mashfy', '1009', '1999-08-21', 'Victory monument road,janicia hub', '2018', '29000'),
('Sadad', '1010', 'Luxary condo ,near bts nana', '2017', '2019', '8000'),
('Ayaaz', '1011', '1997-08-21', 'Phuket city,nanaria apartment', '2019', '9000'),
('Amna', '1012', '1999-07-04', 'pattaya city,gracious apartment', '20120', '8000');
```

The sql commands (in bits)for storing the cart data from the DB to the App:

```
<?php
$conn = mysqli_connect("localhost", "root", "", "list");
if ($conn->connect_error) {
die("Connection failed: " . $conn->connect_error);}
$sql = "SELECT * FROM gardeners

ORDER by Id  ASC;";
```

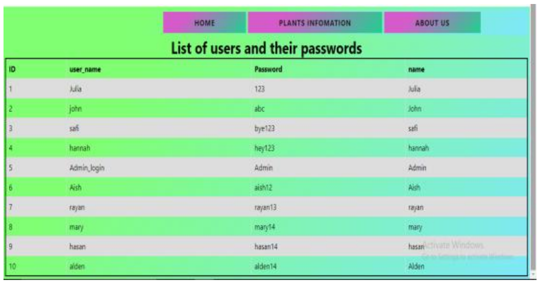
5)Table ‘users’

The table is created by:

```
CREATE TABLE `users` (
  `id` int(11) NOT NULL,
  `user_name` varchar(255) NOT NULL,
  `password` varchar(255) NOT NULL,
  `name` varchar(255) NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
```

The value is inserted by:

```
INSERT INTO `users` (`id`, `user_name`, `password`, `name`) VALUES
(1, 'Julia', '123', 'Julia'),
(2, 'john', 'abc', 'john'),
(3, 'safi', 'bye123', 'safi'),
(4, 'hannah', 'hey123', 'hannah'),
(5, 'Admin_login', 'Admin', 'Admin_login');
(6, 'Aish', 'aish12', 'Aish'),
(7, 'rayan', 'rayan13', 'rayan'),
(8, 'mary', 'mary14', 'mary'),
(9, 'hasan', 'hasan14', 'hasan'),
(10, 'alden', 'alden14', 'Alden');
```



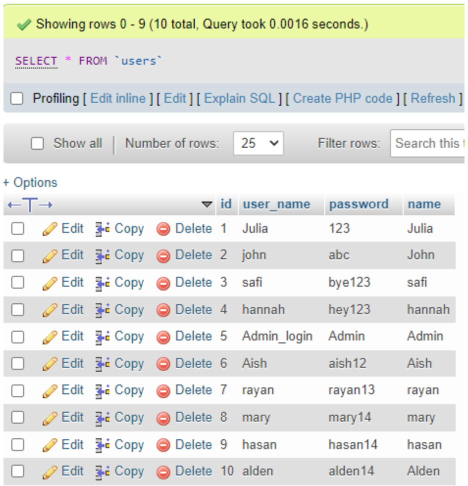
List of users and their passwords

ID	user_name	Password	name
1	Julia	123	Julia
2	john	abc	John
3	safi	bye123	safi
4	hannah	hey123	hannah
5	Admin_login	Admin	Admin
6	Aish	aish12	Aish
7	rayan	rayan13	rayan
8	mary	mary14	mary
9	hasan	hasan14	hasan
10	alden	alden14	Alden

The sql commands (in bits)for storing the user data from the DB to the App:

```
<?php
$conn = mysqli_connect("localhost", "root", "", "test_db");
if ($conn->connect_error) {
```

```
die("Connection failed: " . $conn->connect_error);
}
$sql = "SELECT * FROM users
ORDER by id  ASC;"
```


Profiling [Edit inline] [Edit] [Explain SQL] [Create PHP code] [Refresh]
Show all | Number of rows: 25
Filter rows: Search this
+ Options
←T→

✓ Showing rows 0 - 9 (10 total, Query took 0.0016 seconds.)

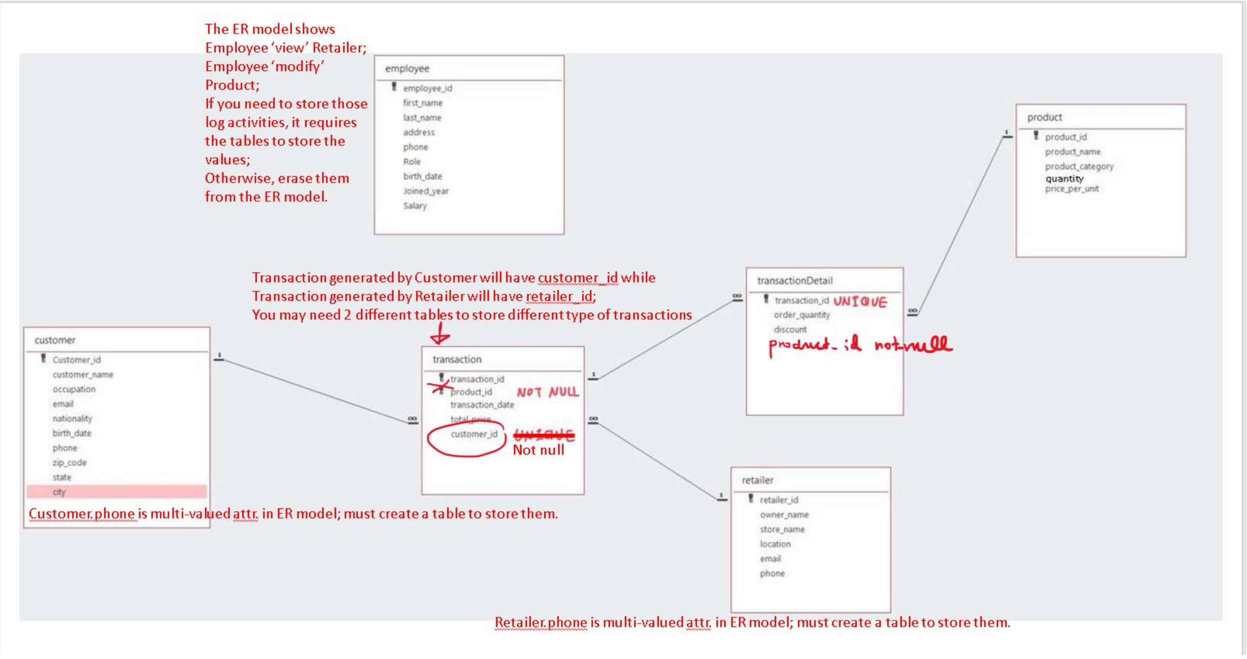
```
SELECT * FROM `users`
```

	id	user_name	password	name
□ Edit Copy Delete	1	Julia	123	Julia
□ Edit Copy Delete	2	john	abc	John
□ Edit Copy Delete	3	safi	bye123	safi
□ Edit Copy Delete	4	hannah	hey123	hannah
□ Edit Copy Delete	5	Admin_login	Admin	Admin
□ Edit Copy Delete	6	Aish	aish12	Aish
□ Edit Copy Delete	7	rayan	rayan13	rayan
□ Edit Copy Delete	8	mary	mary14	mary
□ Edit Copy Delete	9	hasan	hasan14	hasan
□ Edit Copy Delete	10	alden	alden14	Alden

Normalization & Functional Dependencies

We have fixed the relational model to be more effective, we think that having customer and retailer table might cause data redundancy, so we combine it in the customer and add store_name which can be NULL if the customer does not have a store (mean that they are not a retailer if they have store_name it implies that they are a retailer). We have also added another table called customerPhone because one customer can have more than 1 phone number.

The old relational model



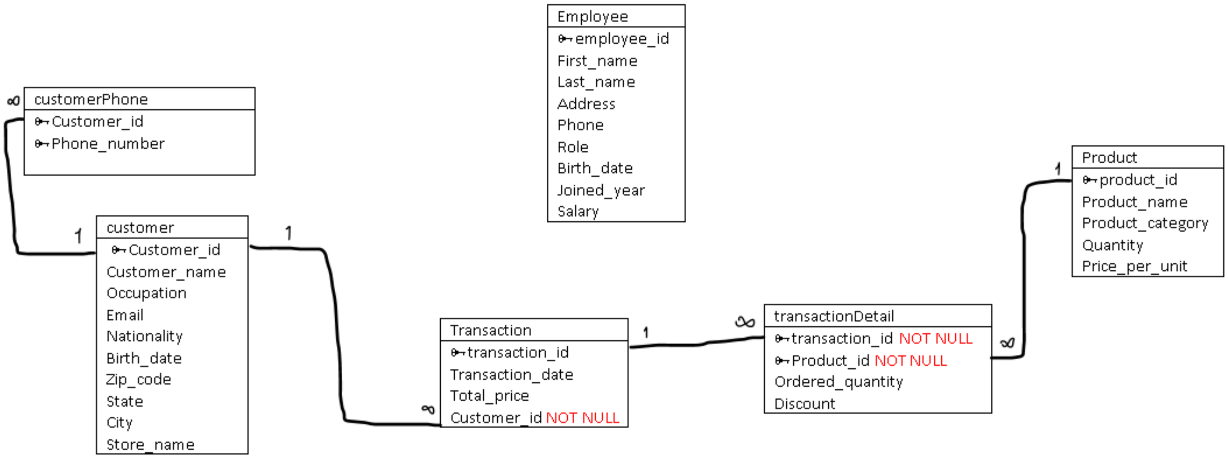
The ER model shows
Employee 'view' Retailer;
Employee 'modify'
Product;
If you need to store those
log activities, it requires
the tables to store the
values;
Otherwise, erase them
from the ER model.

Transaction generated by Customer will have customer_id while
Transaction generated by Retailer will have retailer_id;
You may need 2 different tables to store different type of transactions

Customer.phone is multi-valued attr. in ER model; must create a table to store them.

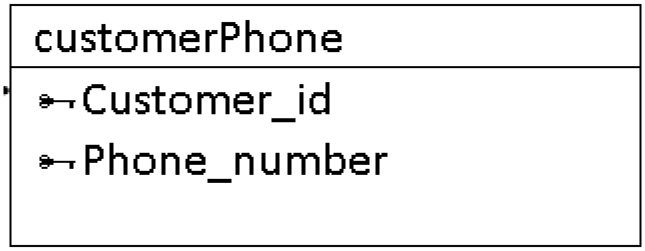
Retailer.phone is multi-valued attr. in ER model; must create a table to store them.

The new relational model



- **Functional Dependencies of each table in the relational model.**

  1.

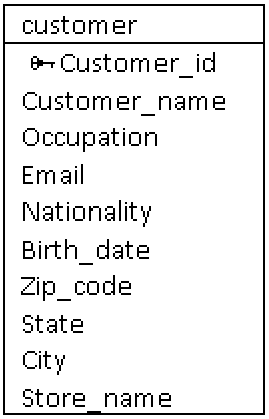


customerPhone

Customer_id
Phone_number

Customer_id → phone_number

2.



customer

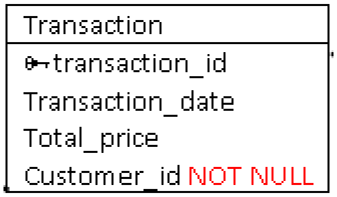
🔑 Customer_id
Customer_name
Occupation
Email
Nationality
Birth_date
Zip_code
State
City
Store_name

customer_id→ customer_name, occupation, email, nationality, birth_date, store_name, state, city, zip_code

Zip_code → state, city

City → zip_code, state

3.

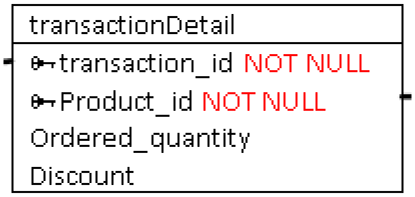


Transaction

🔑 transaction_id
Transaction_date
Total_price
Customer_id NOT NULL

Transaction_id → transaction_date, total_price, customer_id

4.

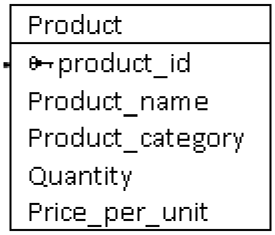


transactionDetail

🔑 transaction_id NOT NULL
🔑 Product_id NOT NULL
Ordered_quantity
Discount

Transaction_id, Product_id → ordered_quantity, discount

5.



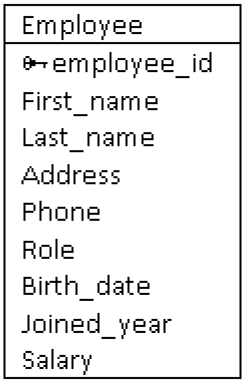
Product

🔑 product_id
Product_name
Product_category
Quantity
Price_per_unit

Product_id → product_name, product_category, quantity, price_per_unit

product_name → product_id, product_category, quantity, price_per_unit

6.



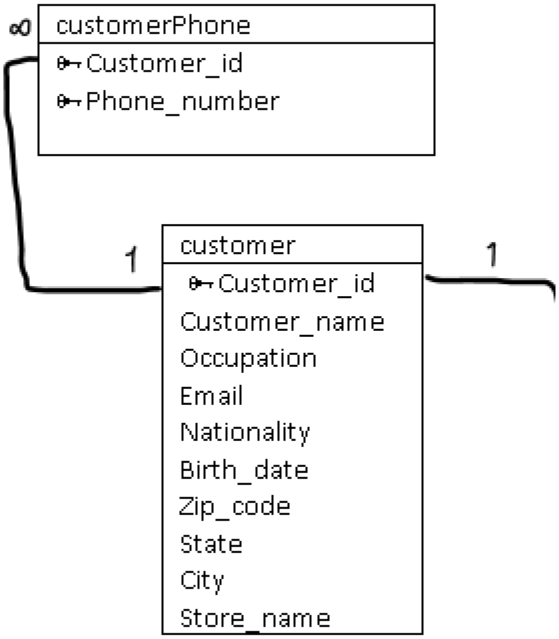
Employee
employee_id
First_name
Last_name
Address
Phone
Role
Birth_date
Joined_year
Salary

Employee_id → first_name, last_name, address, phone, role, birth_date, joined_year, salary

joined_year → salary, role

- **Normalize each table to conform to BCNF**

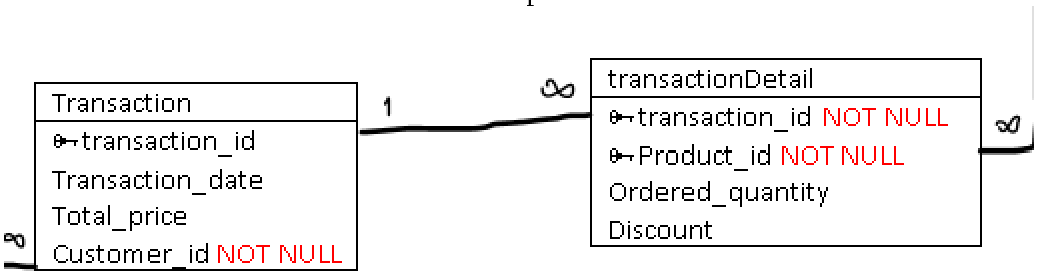
The table customerPhone is the result of our normalization of table customer. Since customers can have more than 1 phone number, we do not put all customer's phone numbers in one cell, but instead, we created another table to store each phone number of a customer which followed the 1NF and 2NF rules. And since we do not have transitive dependencym hench tables is in 3NF.



customerPhone
Customer_id
Phone_number

customer
Customer_id
Customer_name
Occupation
Email
Nationality
Birth_date
Zip_code
State
City
Store_name

Last but not least, transactionDetail is the result of our normalization of table transaction. Because a customer can order more than 1 item in 1 order, therefore we create another table to support that. Instead of putting transaction_id and product_id in the same table which follows the 1NF and 2NF rules. And no transitive dependencies therefore tables are normalized.



CONCLUSION

This project about the glasshouse is an attempt to use a mix of different languages in order to create an app that is close to the real world App. However, there are some defects in this application, such as the issue of convenience and to some extent, the accessibility of different roles of users, However, the team has worked hard to make sure that the database is connected and the SQL codes are correctly implemented, along with the Normalization using the ER model and the relational table. Moreover, throughout this project, our team has explored the various scope of studies related to databases and figured out new tricks for retrieving data and also got to implement knowledge from other courses, such as the HTML, PHP, and CSS and we also attempted to learn and experiment with new languages such as bootstrap and ajax so that we could complete out the cart and checkout process. Overall, This project gave us an overview insight into the functionalities of database access through SQL and other required languages to form a functional application.