



WINTER SEMESTER 2024

CSE250 - DATABASE MANAGEMENT SYSTEM

PROJECT : WALMART - INDIA (E-commerce website)

FACULTY : PROF. SHEFALI NAIK

SECTION : 1

GROUP MEMBERS :

Tanay Shah	AU2240007
Sarav Shah	AU2240015
Harsh Panchal	AU2240160

Project Description

In today's digital era, online shopping has become a cornerstone of convenience, which offers a wide range of products on our fingertips. There are countless platforms that are comparable with this one and are offering many functionalities to interact with users. Database is the root of any E-commerce websites. And therefore, our online shopping platform "Walmart India" aims to develop a comprehensive database system for managing the operations of a Walmart store located in different states of India. This system covers everything from users and sellers to all products we sell. It is important because it helps us to keep everything well organized and working smoothly. Along with this we tried to make it easy to use and reliable by using the different concepts of database management systems like functions, triggers, and PLSQL so that our website/platform runs without any problems. This report will provide an overview of database schema, its functionality and the basic explanation behind its design.

Tech Used in Making E-Commerce Website

Front-End:

- Android Studio
- Java

Back-End:

- PHP

Database:

- MySQL

Applications

- Customers can shop anytime, anywhere with our website and we tried our best to provide the best shopping experience with us. Also, our responsive design makes it easy to browse and purchase products whenever you want and from any devices.
- To promote products and services to a wider audience, we use email marketing and social media advertising. This ensures that more people know about our products, and therefore, increasing our chances of attracting potential customers.
- Our system provides real - time inventory to track the status of purchased products and also automatically replenishes stock when it runs low, so that products are always available.
- Our platform offers secure payment processing through multiple gateways like UPI, Debit/Credit Card, Cash on delivery, etc. with SSL encryption to maintain data privacy of customers during transactions.
- Customers can create accounts to personalize their experience and can track their order history and manage their preferences after logging in.
- Our website has detailed descriptions, images and prices of all products. Which makes it easy to find what customers are looking for with our user - friendly interface.
- Customers can track their ordered products status like purchase date and time, dispatched date, expected delivery date and time.
- Once customers get their products they can give the review on that product on our websites with text or rating or both. So that other customers can make proper decisions if they want to buy the same product/s.

Functionalities provided in our website

Supplier Management:

- Add: Ability to add new suppliers to the database with their contact information.
- Update: Capability to update supplier details such as address, contact name, email, and phone number.
- Delete: Option to remove suppliers from the database, cascading to related product information if necessary.
- View: Ability to view a list of all suppliers along with their details.

Category Management:

- Add: Capability to add new product categories to classify the products effectively.
- Update: Ability to modify category names or attributes.
- Delete: Option to delete categories, with cascading effects on related product information.
- View: Ability to view all available product categories.

Store Management:

- Add: Ability to add new stores with their respective names and locations.
- Update: Capability to update store details such as name and location.
- Delete: Option to delete stores from the database, cascading to related transaction and sales information if necessary.
- View: Ability to view a list of all stores along with their details.

Customer Management:

- Add: Capability to add new customers to the database with their personal details.
- Update: Ability to update customer information including name, gender, email, phone number, and address.
- Delete: Option to delete customers from the database, cascading to related transaction and order information if necessary.
- View: Ability to view a list of all customers along with their details.

Product Management:

- Add: Capability to add new products with details such as name, description, quantity, price, supplier, and category.
- Update: Ability to update product information including quantity, price, supplier, and category.
- Delete: Option to delete products from the database, cascading to related transaction, sales, and order information if necessary.
- View: Ability to view a list of all products along with their details.

Transaction Management:

- Record: Capability to record transactions made by customers including payment type, unit price, quantity, total amount, and associated customer and store.
- View: Ability to view transaction history including details of each transaction such as date, payment type, total amount, and associated customer and store.

Sales Management:

- Record: Ability to record sales transactions including date, unit price, quantity, total amount, and associated product, store, and transaction.
- View: Capability to view sales history including details of each sale such as date, product, quantity, total amount, and associated store and transaction.

Order Management:

- Place: Capability to place customer orders including product, quantity, unit price, total amount, order date, expected delivery date, and customer details.
- Update: Ability to update order status (e.g., pending, shipped, delivered).
- Cancel: Option to cancel pending orders.
- View: Ability to view order history including details of each order such as order date, expected delivery date, status, product, quantity, and total amount.

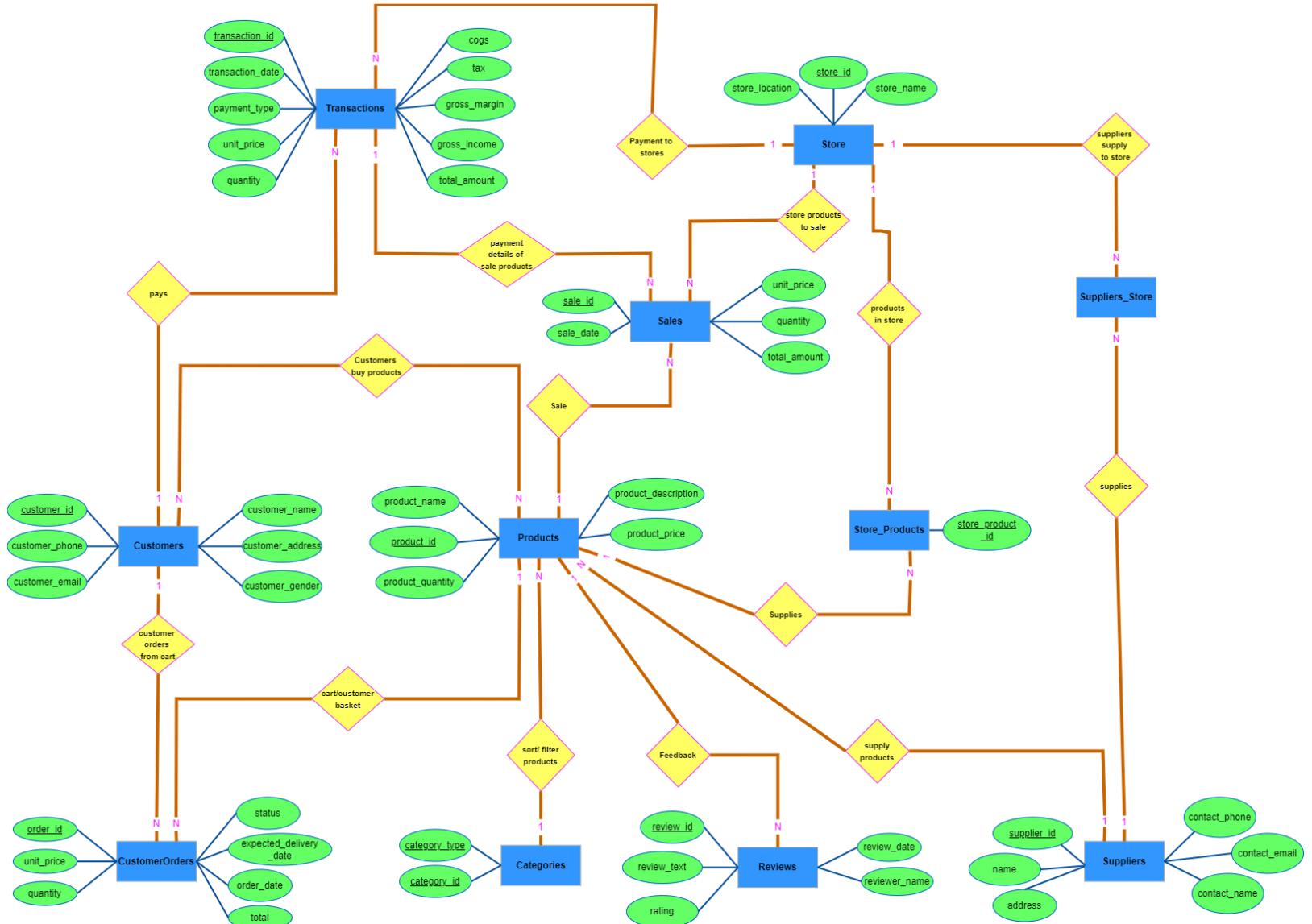
Review Management:

- Add: Capability to add product reviews including review text, rating, reviewer name, and review date.
- View: Ability to view product reviews along with their details.
-

Relationship Management:

- Suppliers to Stores: Capability to establish relationships between suppliers and stores, allowing stores to procure products from specific suppliers.
- Stores to Products: Ability to establish relationships between stores and products they carry, facilitating inventory management and product availability.

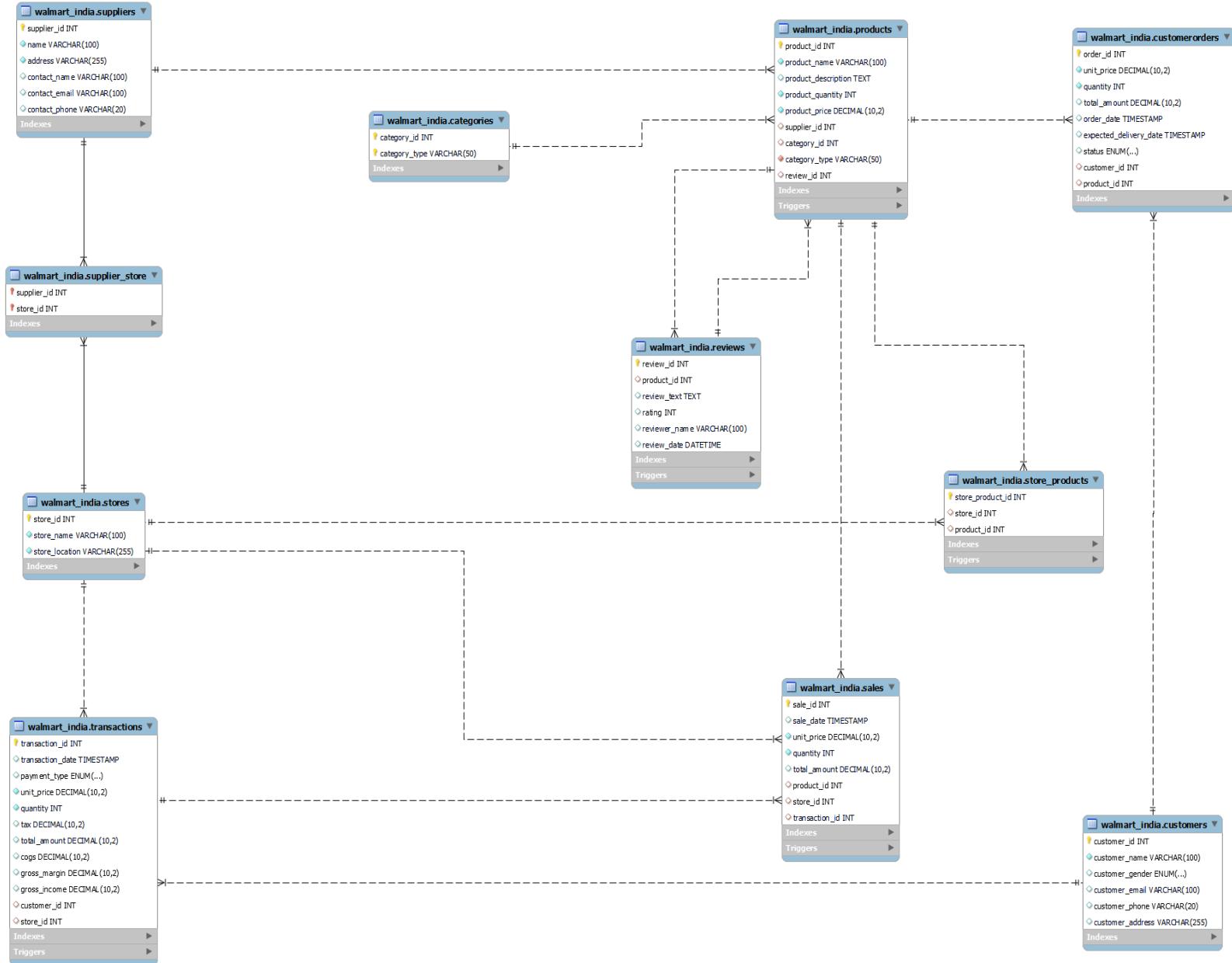
ER Diagram



Link for ER diagram :

https://drive.google.com/file/d/1DLJybzbSzcZ1AKliZ3nu8queluLXrvt/view?usp=drive_link

Table Design



TABLES FOR DATABASE

```
CREATE DATABASE IF NOT EXISTS Walmart_India;  
USE Walmart_India;
```

```
CREATE TABLE Suppliers (  
    supplier_id INT PRIMARY KEY AUTO_INCREMENT,  
    name VARCHAR(100) NOT NULL,  
    address VARCHAR(255) NOT NULL,  
    contact_name VARCHAR(100),  
    contact_email VARCHAR(100),  
    contact_phone VARCHAR(20)  
);  
ALTER TABLE Suppliers AUTO_INCREMENT = 1;
```

```
CREATE TABLE Categories (  
    category_id INT AUTO_INCREMENT,  
    category_type VARCHAR(50) NOT NULL,  
    PRIMARY KEY (category_id, category_type)  
);  
ALTER TABLE Categories AUTO_INCREMENT = 1;
```

```
CREATE TABLE Stores (  
    store_id INT PRIMARY KEY AUTO_INCREMENT,  
    store_name VARCHAR(100) NOT NULL,  
    store_location VARCHAR(255) not null  
);  
ALTER TABLE Stores AUTO_INCREMENT = 1;
```

```
CREATE TABLE Customers (  
    customer_id INT PRIMARY KEY AUTO_INCREMENT,  
    customer_name VARCHAR(100) NOT NULL,  
    customer_gender ENUM('male', 'female', 'other'),
```

```
        customer_email VARCHAR(100),
        customer_phone VARCHAR(20),
        customer_address VARCHAR(255)
);
ALTER TABLE Customers AUTO_INCREMENT = 1;
```

```
CREATE TABLE Products (
    product_id INT PRIMARY KEY AUTO_INCREMENT,
    product_name VARCHAR(100) NOT NULL,
    product_description TEXT,
    product_quantity INT NOT NULL,
    product_price DECIMAL(10, 2) NOT NULL,
    supplier_id INT,
    category_id INT,
    category_type VARCHAR(50) NOT NULL,
    FOREIGN KEY (supplier_id) REFERENCES Suppliers(supplier_id) ON
    DELETE CASCADE ON UPDATE CASCADE,
    FOREIGN KEY (category_id, category_type) REFERENCES
    Categories(category_id, category_type) ON DELETE CASCADE ON UPDATE
    CASCADE
);
ALTER TABLE Products AUTO_INCREMENT = 1;
```

```
CREATE TABLE Transactions (
    transaction_id INT PRIMARY KEY AUTO_INCREMENT,
    transaction_date TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
    payment_type ENUM('cash', 'credit_card', 'debit_card', 'online_payment',
    'cheque', 'other') DEFAULT 'cash',
    unit_price DECIMAL(10, 2) NOT NULL,
    quantity INT NOT NULL,
    tax DECIMAL(10, 2) DEFAULT (total_amount * 0.18),
    total_amount DECIMAL(10, 2),
    cogs DECIMAL(10, 2) DEFAULT 0.00,
    gross_margin DECIMAL(10, 2) DEFAULT 0.00,
    gross_income DECIMAL(10, 2) DEFAULT 0.00,
```

```
customer_id INT,  
store_id INT,  
FOREIGN KEY (customer_id) REFERENCES Customers(customer_id) ON  
DELETE CASCADE ON UPDATE CASCADE,  
FOREIGN KEY (store_id) REFERENCES Stores(store_id) ON DELETE  
CASCADE ON UPDATE CASCADE  
);  
ALTER TABLE Transactions AUTO_INCREMENT = 1;
```

```
CREATE TABLE Sales (  
sale_id INT PRIMARY KEY AUTO_INCREMENT,  
sale_date TIMESTAMP DEFAULT CURRENT_TIMESTAMP,  
unit_price DECIMAL(10, 2) NOT NULL,  
quantity INT NOT NULL,  
total_amount DECIMAL(10, 2) AS (unit_price * quantity) STORED,  
product_id INT,  
store_id INT,  
transaction_id INT,  
FOREIGN KEY (product_id) REFERENCES Products(product_id) ON  
DELETE CASCADE ON UPDATE CASCADE,  
FOREIGN KEY (store_id) REFERENCES Stores(store_id) ON DELETE  
CASCADE ON UPDATE CASCADE,  
FOREIGN KEY (transaction_id) REFERENCES Transactions(transaction_id)  
ON DELETE CASCADE ON UPDATE CASCADE  
);  
ALTER TABLE Sales AUTO_INCREMENT = 1;
```

```
CREATE TABLE CustomerOrders (  
order_id INT PRIMARY KEY AUTO_INCREMENT,  
unit_price DECIMAL(10, 2) NOT NULL,  
quantity INT NOT NULL,  
total_amount DECIMAL(10, 2) AS (unit_price * quantity) STORED,  
order_date TIMESTAMP DEFAULT CURRENT_TIMESTAMP,  
expected_delivery_date TIMESTAMP DEFAULT (CURRENT_TIMESTAMP +  
INTERVAL 7 DAY),
```

```
status ENUM('pending', 'shipped', 'delivered') DEFAULT 'pending',
customer_id INT,
product_id INT,
FOREIGN KEY (customer_id) REFERENCES Customers(customer_id) ON
DELETE CASCADE ON UPDATE CASCADE,
FOREIGN KEY (product_id) REFERENCES Products(product_id) ON
DELETE CASCADE ON UPDATE CASCADE
);
ALTER TABLE CustomerOrders AUTO_INCREMENT = 1;
```

```
CREATE TABLE Reviews (
review_id INT PRIMARY KEY AUTO_INCREMENT,
product_id INT,
review_text TEXT,
rating INT,
reviewer_name VARCHAR(100),
review_date DATETIME DEFAULT CURRENT_TIMESTAMP,
FOREIGN KEY (product_id) REFERENCES Products(product_id) ON
DELETE CASCADE ON UPDATE CASCADE
);
ALTER TABLE Reviews AUTO_INCREMENT = 1;
```

```
ALTER TABLE Products
ADD COLUMN review_id INT,
ADD FOREIGN KEY (review_id) REFERENCES Reviews(review_id) ON
DELETE CASCADE ON UPDATE CASCADE;
```

```
CREATE TABLE Supplier_Store (
supplier_id INT,
store_id INT,
PRIMARY KEY (supplier_id, store_id),
FOREIGN KEY (supplier_id) REFERENCES Suppliers(supplier_id) ON
DELETE CASCADE ON UPDATE CASCADE,
FOREIGN KEY (store_id) REFERENCES Stores(store_id) ON DELETE
CASCADE ON UPDATE CASCADE
```

);

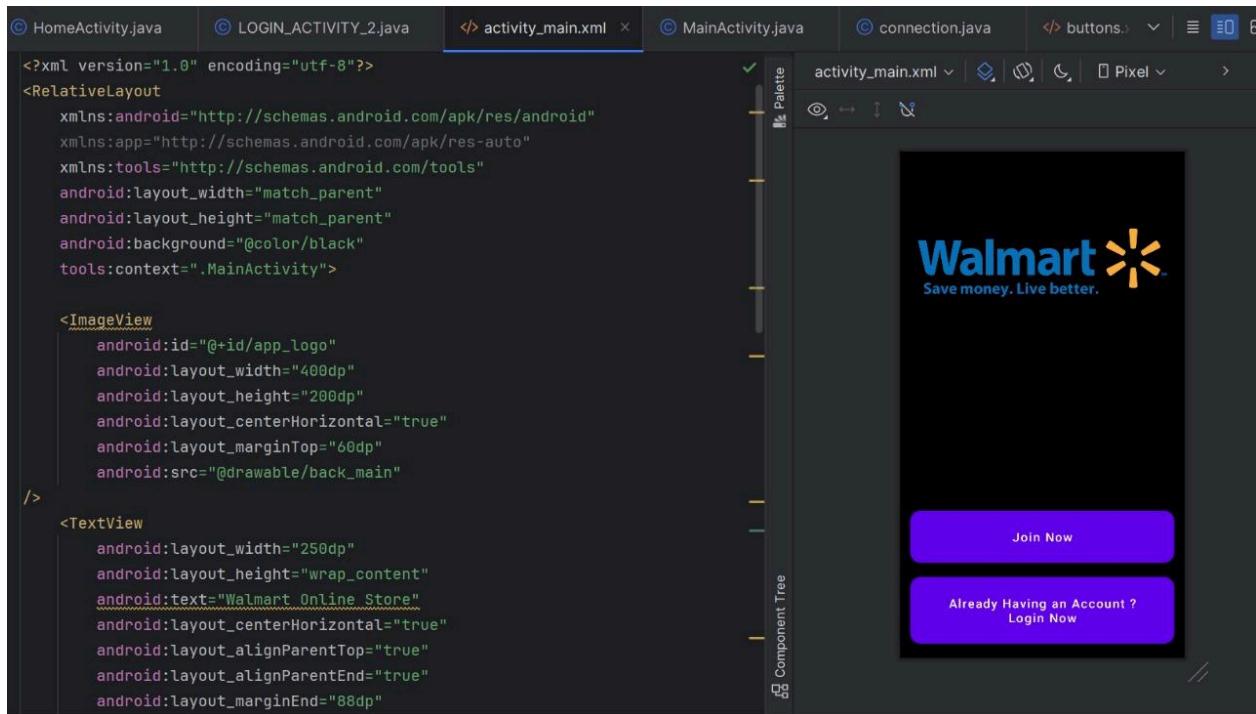
```
CREATE TABLE Store_Products (
    store_product_id INT PRIMARY KEY AUTO_INCREMENT,
    store_id INT,
    product_id INT,
    FOREIGN KEY (store_id) REFERENCES Stores(store_id) ON DELETE
    CASCADE ON UPDATE CASCADE,
    FOREIGN KEY (product_id) REFERENCES Products(product_id) ON
    DELETE CASCADE ON UPDATE CASCADE
);
ALTER TABLE Store_Products AUTO_INCREMENT = 1;
```

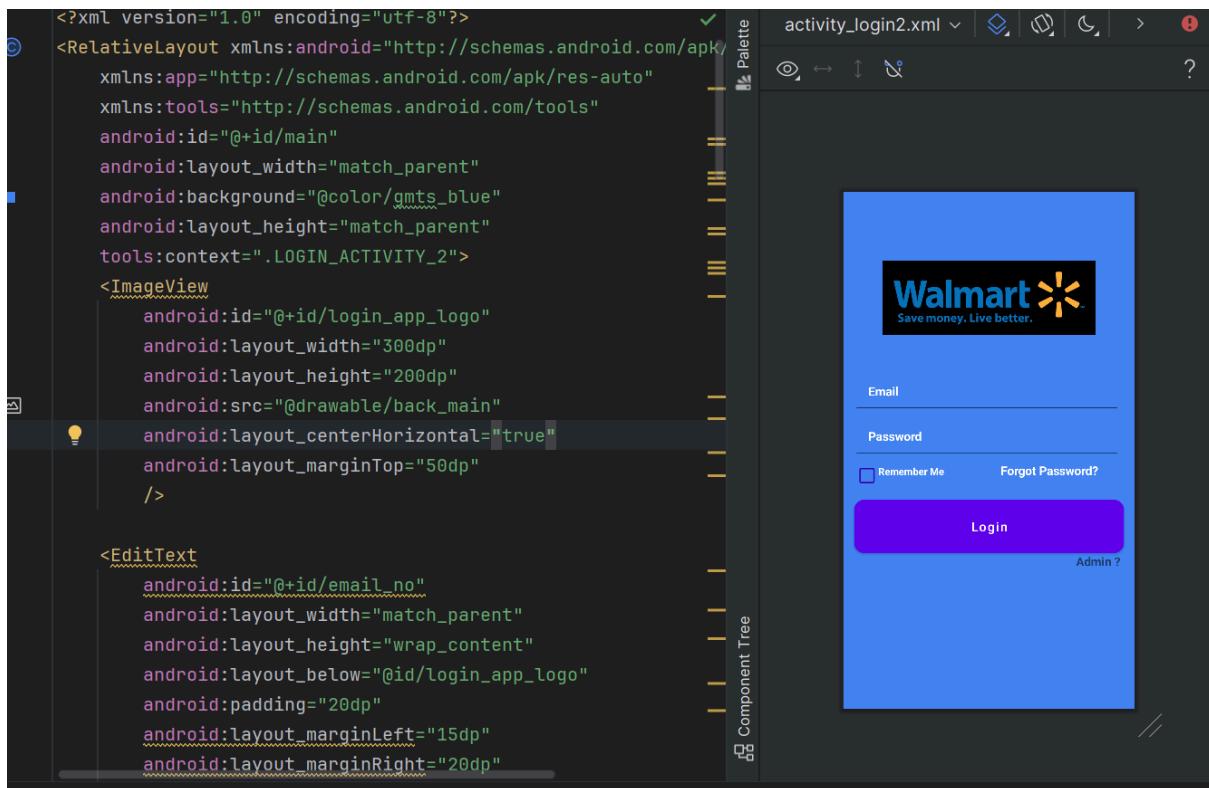
Code to Connect database with Front-End

```
PHP Constants.php    PHP DbConnect.php X    PHP DbOperations.php    PHP registeruser.php
C: > xampp > htdocs > android > includes > PHP DbConnect.php
1  <?php
2  class Dbconnect {
3      private $con;
4
5      function __construct() {
6          []
7
8          function connect() {
9              include_once dirname(__FILE__) . '/Constants.php';
10             $this->con = new mysqli($servername, $username, $password, $dbname);
11             if (mysqli_connect_errno()) {
12                 echo "Failed to connect with database: " . mysqli_connect_error();
13                 return false;
14             }
15             return $this->con;
16         }
17     }
18
19 }
```

```
C: > xampp > htdocs > android > includes > PHP DbOperations.php
1  <?php
2  class DbOperations{
3      private $con;
4      function __construct(){
5          require_once dirname(__FILE__).'/DbConnect.php';
6
7          $db = new DbConnect();
8          $this->con = $db->connect();
9
10     }
11     //CRUD
13     function createuser($uname, $pass, $email) {
14         $password = password_hash($pass, PASSWORD_DEFAULT);
15         $stmt = $this->con->prepare("INSERT INTO `users` (`id`, `uname`, `pass`, `email`) VALUES (
16             $stmt->bind_param("sss", $uname, $password, $email);
17             if ($stmt->execute()) {
18                 return true;
19             } else {
20                 return false;
21             }
22         }
23
24     }
25 }
```

```
Constants.php DbConnect.php DbOperations.php registeruser.php ×  
C: > xampp > htdocs > android > v1 > registeruser.php  
1  <?php  
2  
3  require_once '../includes/DbOperations.php';  
4  $response = array();  
5  
6  if($_SERVER['REQUEST_METHOD']=='POST'){  
7      if(isset($_POST['username']) and isset($_POST['email']) and isset($_POST['password'])){  
8          $db = new DbOperations();  
9          if($db->create_user(  
10              $_POST['username'], $_POST['password'],  
11              $_POST['email'])  
12          ){  
13              $response['error'] = false;  
14              $response['message'] = "User Registered Successfully";  
15          }else{  
16              $response['error'] = true;  
17              $response['message'] = "Some error has occurred!  
Please Try Again After Sometime";  
18          }  
19      }  
20  }  
21  
22 else{  
23     $response['error'] = true;  
24     $response['message'] = "Required Fields are missing";  
25 }else{  
26     $response['error'] = true;  
27     $response['message'] = "Invalid Request";  
28 }  
29 echo json_encode($response);
```





The screenshot shows the Java code editor with the file `MainActivity.java` open. The code defines a `MainActivity` class that extends `AppCompatActivity`. It contains fields for two buttons: `join_now` and `login_now`. The `onCreate` method sets the content view to `R.layout.activity_main` and initializes both buttons. The `login_now` button's `onClick` listener starts an activity named `LOGIN_ACTIVITY_2`. The `join_now` button's `onClick` listener starts an activity named `Register_Activity`.

```
1 package com.example.walmart_sales;
2
3 > import ...
4
5
6 D </> public class MainActivity extends AppCompatActivity {
7     2 usages
8     private Button join_now;
9     2 usages
10    private Button login_now;
11    @Override
12    protected void onCreate(Bundle savedInstanceState) {
13        super.onCreate(savedInstanceState);
14        setContentView(R.layout.activity_main);
15
16        join_now = (Button) findViewById(R.id.join_now);
17        login_now = (Button) findViewById(R.id.login_now);
18
19        login_now.setOnClickListener(new View.OnClickListener() {
20            @Override
21            public void onClick(View view) {
22                Intent intent = new Intent(packageContext: MainActivity.this, LOGIN_ACTIVITY_2.class);
23                startActivity(intent);
24            }
25        });
26        join_now.setOnClickListener(new View.OnClickListener() {
27            @Override
28            public void onClick(View v) {
29                Intent intent = new Intent(packageContext: MainActivity.this, Register_Activity.class);
30                startActivity(intent);
31            }
32        });
33    }
34}
```

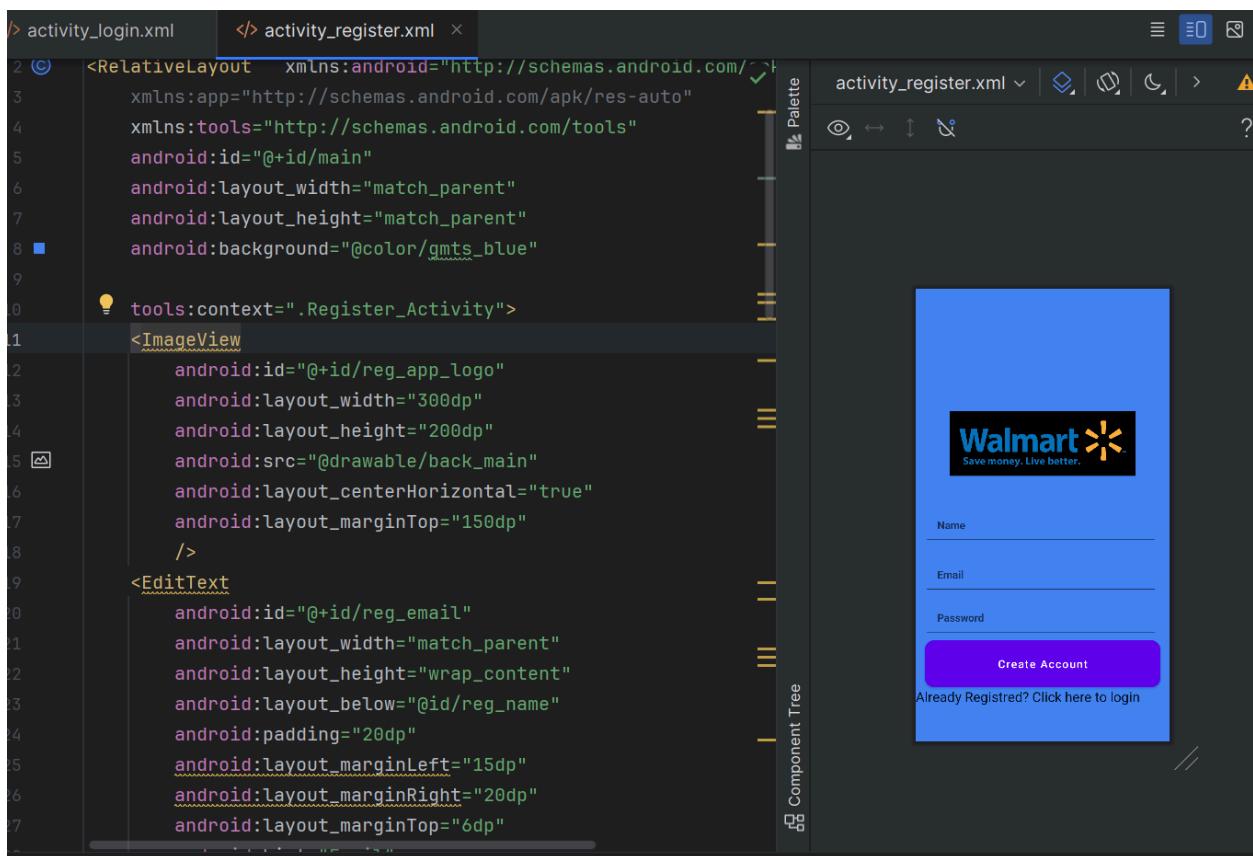
</> activity_login2.xml © MainActivity.java </> activity_categories.xml © connection.java × </> buttons.xml

```
package com.example.walmart_sales;

import ...

1 usage
public class connection {
    //public static String url_register;
    3 usages
    Connection con;
    2 usages
    String uname, pass, ip, port, database;
    1 usage
    final String root_url = "http://127.0.0.1/android/v1/registeruser.php";
    1 usage
    final String url_register = root_url + "registeruser.php";
    no usages
    public Connection connectionClass() {
        ip = "127.0.0.1";
        database = "Walmart_India";
        uname = "root";
        pass = "Harshe@149";
        port = "3306";
        StrictMode.ThreadPolicy policy = new StrictMode.ThreadPolicy.Builder().permitAll().build();
        StrictMode.setThreadPolicy(policy);

        con = null;
        String conurl = null;
        try {
            Class.forName( className: "net.sourceforge.jtds.jdbc.Driver");
            conurl = "jdbc:jtds:sqllserver://" + ip + ":" + port + ";" + "databasename" + database + ";user" + uname + ";password" + pass + ";";
            con = DriverManager.getConnection(conurl);
        } catch (Exception ex) {
            Log.e( tag: "Error", ex.getMessage());
        }
    }
    return con;
}}
```



```
package com.example.walmart_sales;

> import ...

public class LOGIN_ACTIVITY_2 extends AppCompatActivity {
    3 usages
    private EditText email, passwordEditText;
    2 usages
    private Button loginButton;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        EdgeToEdge.enable( $this$enableEdgeToEdge: this);
        setContentView(R.layout.activity_login2);
        email = (EditText) findViewById(R.id.email_no);
        passwordEditText = (EditText) findViewById(R.id.password);
        loginButton = (Button) findViewById(R.id.login_btn);

        loginButton.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                String emailn = email.getText().toString().trim();
                String password = passwordEditText.getText().toString().trim();

                if (emailn.equals(email) && password.equals(passwordEditText)) {
                    Intent intent = new Intent( packageContext: LOGIN_ACTIVITY_2.this, CategoriesActivity.class);
                    startActivity(intent);
                    finish();
                } else {
                    Toast.makeText( context: LOGIN_ACTIVITY_2.this, text: "Invalid username or password", Toast.LENGTH_SHORT).show();
                }
            }
        });
        ViewCompat.setOnApplyWindowInsetsListener(findViewById(R.id.main), (v, insets) -> {
            Insets systemBars = insets.getInsets(WindowInsetsCompat.Type.systemBars());
            v.setPadding(systemBars.left, systemBars.top, systemBars.right, systemBars.bottom);
            return insets;
        });
    }
}
```

Font size: 8pt Reset to 13pt

/> activity_login.xml </> activity_register.xml ⚒ Register_Activity.java ×

```
1 package com.example.walmart_sales;
2
3 > import ...
4
5 // Update imports as needed
6
7 >< public class Register_Activity extends AppCompatActivity implements View.OnClickListener {
8
9     3 usages
10    private Button createAccBtn;
11
12    5 usages
13    private ProgressDialog progressDialog;
14
15    2 usages
16    private EditText inputName, inputEmail, inputPassword;
17
18
19    @Override
20    protected void onCreate(Bundle savedInstanceState) {
21        super.onCreate(savedInstanceState);
22        setContentView(R.layout.activity_register);
23
24        createAccBtn = findViewById(R.id.reg_btn);
25        inputName = findViewById(R.id.reg_name);
26        inputEmail = findViewById(R.id.reg_email);
27        inputPassword = findViewById(R.id.reg_password);
28        progressDialog = new ProgressDialog(context: this);
29
30        createAccBtn.setOnClickListener(this);
31    }
32
33
34    1 usage
35    private void createAccount() {
36        String name = inputName.getText().toString().trim();
37        String email = inputEmail.getText().toString().trim();
38        String password = inputPassword.getText().toString().trim();
39
40        if (TextUtils.isEmpty(name) || TextUtils.isEmpty(email) || TextUtils.isEmpty(password)) {
41            Toast.makeText(context: this, text: "Please fill in all fields", Toast.LENGTH_SHORT).show();
42            return;
43        }
44    }
45
```

```
activity_login.xml | activity_register.xml | Register_Activity.java × 20

64     private void createAccount() {
65         String name = inputName.getText().toString().trim();
66         String email = inputEmail.getText().toString().trim();
67         String password = inputPassword.getText().toString().trim();
68
69         if (TextUtils.isEmpty(name) || TextUtils.isEmpty(email) || TextUtils.isEmpty(password)) {
70             Toast.makeText(context, text: "Please fill in all fields", Toast.LENGTH_SHORT).show();
71             return;
72         }
73
74         progressDialog.setMessage("Registering User...");
75         progressDialog.show();
76
77         StringRequest stringRequest = new StringRequest(Request.Method.POST, connection.url_register,
78             new Response.Listener<String>() {
79                 @Override
80                 public void onResponse(String response) {
81                     progressDialog.dismiss();
82                     try {
83                         JSONObject jsonObject = new JSONObject(response);
84                         boolean success = jsonObject.getBoolean( name: "success");
85                         if (success) {
86                             Toast.makeText( context: Register_Activity.this, jsonObject.getString( name: "message"), Toast.LENGTH_LONG).show();
87                             // Handle successful registration, e.g., navigate to another activity
88                         } else {
89                             Toast.makeText( context: Register_Activity.this, text: "Registration failed: " + jsonObject.getString( name: "message"), Toast.LENGTH_LONG).show();
90                         }
91                     } catch (JSONException e) {
92                         e.printStackTrace();
93                         Toast.makeText( context: Register_Activity.this, text: "Error parsing response", Toast.LENGTH_LONG).show();
94                     }
95                 },
96                 new Response.ErrorListener() {
97                     no usages
98                     @Override
99                     public void onErrorResponse(VolleyError error) {
100                         progressDialog.hide();
101                         Toast.makeText( context: Register_Activity.this, text: "Registration failed: " + error.getMessage(), Toast.LENGTH_LONG).show();
102                     }
103                 }
104             },
105             new Response.ErrorListener() {
106                 no usages
107                 @Override
108                 public void onErrorResponse(VolleyError error) {
109                     progressDialog.hide();
110                     Toast.makeText( context: Register_Activity.this, text: "Registration failed: " + error.getMessage(), Toast.LENGTH_LONG).show();
111                 }
112             }
113         );
114
115         @Override
116         protected Map<String, String> getParams() {
117             Map<String, String> params = new HashMap<>();
118             params.put("username", name);
119             params.put("email", email);
120             params.put("password", password);
121             return params;
122         }
123
124         RequestQueue requestQueue = Volley.newRequestQueue( context: this);
125         requestQueue.add(stringRequest);
126     }
127
128     @Override
129     public void onClick(View v) {
130         if (v == createAccBtn) {
131             createAccount();
132         }
133     }
134 }
```

The screenshot shows the Android Studio interface with the XML code for `activity_categories.xml` on the left and a preview of the UI on the right.

XML Code:

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:id="@+id/main"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".CategoriesActivity"
    android:background="@color/black"
    tools:ignore="Orientation">

    <TextView
        android:id="@+id/slogan_category"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignParentTop="true"
        android:layout_centerHorizontal="true"
        android:layout_marginTop="27dp"
        android:layout_marginBottom="16dp"
        android:text="Select your category"
        android:textColor="@color/white"
        android:textSize="26sp"
        android:textStyle="bold" />

    <LinearLayout
        android:id="@+id/linear1_layout"
        ...>
```

Preview:

The preview window displays a black screen with a white text view in the center. The text view contains the text "Select your category" in bold, white, 26sp font. Below the text view, there is a horizontal separator line.

Component Tree:

A sidebar on the right labeled "Component Tree" shows a tree structure of the UI components defined in the XML code.

Here the images will work as buttons as shown in code below

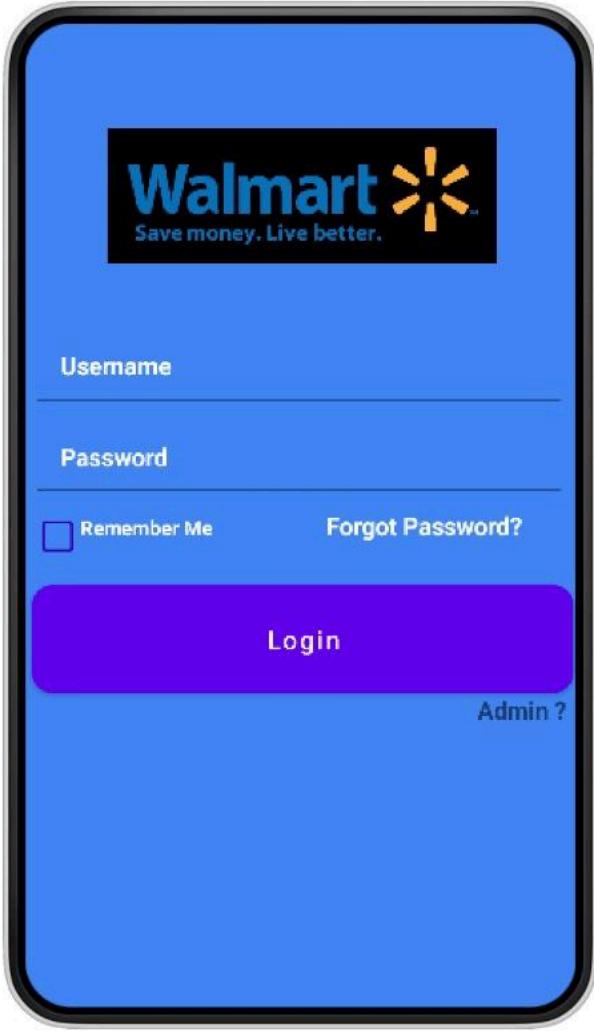
```
© CategoriesActivity.java </> activity_categories.xml : 20 ✓ 2 ^ :
```

```
1 package com.example.walmart_sales;
2
3 > import ...
4
5 D </> public class CategoriesActivity extends AppCompatActivity {
6     2 usages
7
8     private ImageView electronics, fashion, healthcare, kids, sports, luggage, kitchen, grocery, beauty, p
9
10
11     @Override
12     protected void onCreate(Bundle savedInstanceState) {
13         super.onCreate(savedInstanceState);
14         EdgeToEdge.enable( $this$enableEdgeToEdge: this);
15         setContentView(R.layout.activity_categories);
16         electronics = findViewById(R.id.electronics);
17         fashion = findViewById(R.id.fashion);
18         healthcare = findViewById(R.id.health);
19         kids = findViewById(R.id.kids);
20         sports = findViewById(R.id.sports);
21         luggage = findViewById(R.id.luggage);
22         kitchen = findViewById(R.id.kitchen);
23         grocery = findViewById(R.id.grocery);
24         beauty = findViewById(R.id.beauty);
25         petcare = findViewById(R.id.pet);
26
27         electronics.setOnClickListener(new View.OnClickListener() {
28             @Override
29             public void onClick(View v) {
30
31                 });
32
33                 fashion.setOnClickListener(new View.OnClickListener() {
34                     @Override
35                     public void onClick(View v) {
36
37                         });
38
39                         Intent intent = new Intent( packageContext: CategoriesActivity.this, Newproductactivity.class)
40                         intent.putExtra( name: "category", value: "fashion");
41                         startActivity(intent);
42
43                     );
44
45                     healthcare.setOnClickListener(new View.OnClickListener() {
46                         @Override
47                         public void onClick(View v) {
48                             Intent intent = new Intent( packageContext: CategoriesActivity.this, Newproductactivity.class)
49                             intent.putExtra( name: "category", value: "healthcare");
50                             startActivity(intent);
51
52                     );
53
54                     kids.setOnClickListener(new View.OnClickListener() {
55                         @Override
56                         public void onClick(View v) {
57                             Intent intent = new Intent( packageContext: CategoriesActivity.this, Newproductactivity.class)
58                             intent.putExtra( name: "category", value: "kids");
59                             startActivity(intent);
60
61                     );
62
63                     sports.setOnClickListener(new View.OnClickListener() {
```

Images of Our Website :



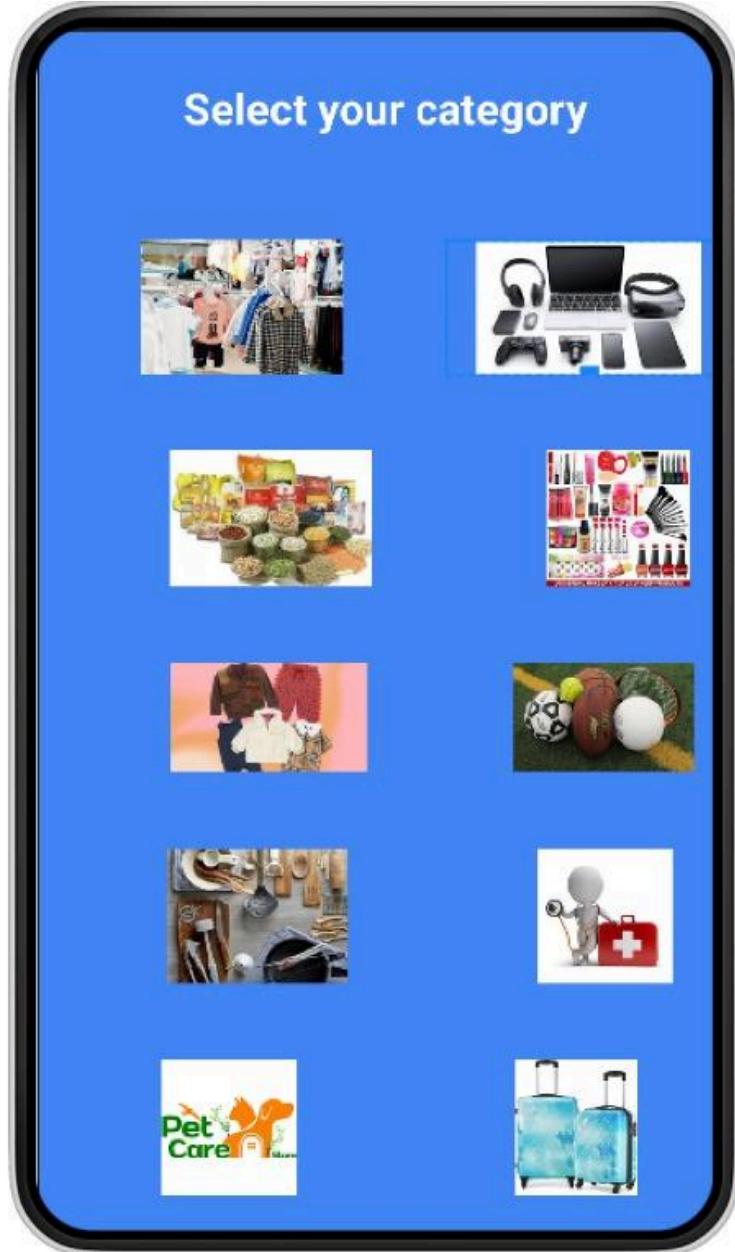
If you already have an account



If you are new user



After login in our website



11 stored procedures and functions

1) GetCustomerOrders:-

The procedure fetches all orders from the CustomerOrders table where the customer_id matches the provided input parameter in_customer_id.

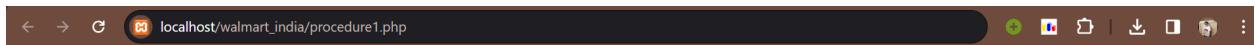
The result contains all fields from the CustomerOrders table for the specified customer.

```
DELIMITER $$  
CREATE PROCEDURE GetCustomerOrders(in_customer_id INT)  
BEGIN  
    SELECT * FROM CustomerOrders WHERE customer_id =  
in_customer_id;  
END$$  
DELIMITER ;  
  
CALL GetCustomerOrders(1);
```

Code for calling function :

```
procedure1.php X procedure2.php
C: > xampp > htdocs > walmart_india > procedure1.php
1  <?php
2  $servername = "localhost";
3  $username = "root";
4  $password = "Harsh@149";
5  $dbname = "walmart_india";
6
7  header('Content-Type: application/json');
8
9  $mysqli = new mysqli("localhost", "your_username", "your_password", "your_database_name");
10
11 if ($mysqli->connect_error) {
12     die("Connection failed: " . $mysqli->connect_error);
13 }
14
15 if ($_SERVER['REQUEST_METHOD'] === 'GET' && isset($_GET['customer_id'])) {
16     $customerId = $_GET['customer_id'];
17
18     $stmt = $mysqli->prepare("CALL GetCustomerOrders(?)");
19     $stmt->bind_param("i", $customerId);
20     $stmt->execute();
21
22     $result = $stmt->get_result();
23
24     $orders = array();
25     while ($row = $result->fetch_assoc()) {
26         $orders[] = $row;
27     }
28
29     $stmt->close();
30     $mysqli->close();
31
32     echo json_encode($orders);
33 }
```

Output :



A screenshot of a web browser window. The address bar shows the URL: localhost/walmart_india/procedure1.php. The page content displays a table with two rows of data.

order_id	unit_price	quantity	total_amount	order_date	expected_delivery_date	status	customer_id	product_id
1	1299.99	2	2599.98	2024-04-05 10:32:34	2024-04-12 10:32:34	pending	1	1
2	499.99	3	1499.97	2024-04-05 10:32:34	2024-04-12 10:32:34	pending	1	2

2) GetSupplierProducts:-

The function calculates the count of products from the Products table where the supplier_id matches the provided input parameter supplier_id_param.

```
DELIMITER $$  
CREATE FUNCTION GetSupplierProducts(supplier_id_param INT)  
RETURNS INT  
READS SQL DATA  
BEGIN  
    DECLARE product_count INT;  
    SELECT COUNT(*) INTO product_count FROM Products WHERE  
supplier_id = supplier_id_param;  
    RETURN product_count;  
END$$  
DELIMITER ;  
  
SELECT GetSupplierProducts(1) AS product_count;
```

Code for calling function :

```
C: > xampp > htdocs > walmart.india > procedure2.php
 1  <?php
 2  $servername = "localhost";
 3  $username = "root";
 4  $password = "Harsh@149";
 5  $dbname = "walmart_india";
 6
 7  $conn = new mysqli($servername, $username, $password, $dbname);
 8
 9
10 if ($conn->connect_error) {
11     die("Connection failed: " . $conn->connect_error);
12 }
13
14 $supplier_id = 1;
15 $sql = "SELECT GetSupplierProducts($supplier_id) AS product_count";
16 $result = $conn->query($sql);
17
18 if ($result->num_rows > 0) {
19     while($row = $result->fetch_assoc()) {
20         echo "Number of products supplied by supplier with ID $supplier_id: " . $row["product_count"];
21     }
22 } else {
23     echo "0 results";
24 }
25
26 $conn->close();
27 ?>
28
```

Output



3) GetProductReviews:-

The procedure fetches all reviews from the Reviews table where the product_id matches the provided input parameter prod_id.

The result contains all fields from the Reviews table for the specified product.

```
DELIMITER $$  
CREATE PROCEDURE GetProductReviews(IN prod_id INT)  
BEGIN  
    SELECT * FROM Reviews WHERE product_id = prod_id;  
END$$  
DELIMITER ;  
  
CALL GetProductReviews(123);
```

Code for calling function :

```
C: > xampp > htdocs > walmart_india > procedure3.php
1  <?php
2  $servername = "localhost";
3  $username = "root";
4  $password = "Harsh@149";
5  $dbname = "walmart_india";
6
7  $conn = new mysqli($servername, $username, $password, $dbname);
8  if ($conn->connect_error) {
9      die("Connection failed: " . $conn->connect_error);
10 }
11
12 $productId = $_GET['product_id'];
13
14 $sql = "CALL GetProductReviews($productId)";
15 $result = $conn->query($sql);
16
17 if ($result->num_rows > 0) {
18     $reviews = array();
19     while($row = $result->fetch_assoc()) {
20         $reviews[] = $row;
21     }
22     echo json_encode($reviews);
23 } else {
24     echo "0 results";
25 }
26 $conn->close();
27 ?>
28
```

Output :



review_id	product_id	review_text	rating	reviewer_name	review_date
123	123	Met my expectations.	4	Prem Patel	2024-04-05 10:32:34
124	123	Grateful	5	Haard Patel	2024-04-05 10:32:34

4) UpdateProductPrice:-

The procedure updates the product_price field in the Products table for the specified product ID with the provided new price.

```
DELIMITER //
CREATE PROCEDURE UpdateProductPrice
(
    IN product_id_param INT,
    IN new_price DECIMAL(10, 2)
)
BEGIN
    UPDATE Products SET product_price = new_price WHERE product_id =
product_id_param;
END //
DELIMITER ;
```

```
CALL UpdateProductPrice(123, 19.99);
```

Code for calling function :

```
C:\>xampp>htdocs>walmart_india>procedure4_1.php
1  <?php
2  $servername = "localhost";
3  $username = "root";
4  $password = "Harsh@149";
5  $dbname = "walmart_india";
6  $conn = new mysqli($servername, $username, $password, $dbname);
7  if ($conn->connect_error) {
8      die("Connection failed: " . $conn->connect_error);
9  }
10
11 $productId = $_GET['product_id'];
12 $newPrice = $_GET['new_price'];
13
14 $sql = "CALL UpdateProductPrice($productId, $newPrice)";
15 if ($conn->query($sql) === TRUE) {
16     echo "Product price updated successfully";
17 } else {
18     echo "Error updating product price: " . $conn->error;
19 }
20 $conn->close();
21 ?>
22
```

SleepWell

Natural Sleep Aid to promote restful and deep sleep



Quantity: 100 Price: 24.99

[Want to Update?](#)

Digest Ease

Natural Digestive Aid for a Comfortable Stomach



Quantity: 100 Price: 12.99

[Want to Update?](#)

Product Name:
SleepWell

Product Price:
24.99

New Price:
19.99

Product Quantity:
100

New Quantity:
100

UPDATE

Output Before

119	Trendy Scarf	Stylish scarf made from soft fabric, adds a touch of elegance to any outfit.	100	14.99	3	2	Fashion	NULL
120	Designer Perfume	Exquisite fragrance with a blend of floral and woody notes, perfect for special occasions.	100	89.99	3	2	Fashion	NULL
121	PainRelief	Effective pain relief medication for various types of pain.	100	15.99	3	3	HealthCare	NULL
122	VitaminBoost	A daily multivitamin to boost your overall health and immunity.	100	19.99	3	3	HealthCare	NULL
123	SleepWell	Natural sleep aid to promote restful and deep sleep.	100	24.99	3	3	HealthCare	NULL
124	DigestEase	Supports healthy digestion and relieves gastrointestinal discomfort.	100	12.99	3	3	HealthCare	NULL
125	HeartCare	Promotes cardiovascular health and maintains healthy cholesterol levels.	100	29.99	3	3	HealthCare	NULL
126	ImmuneGuard	Strengthens the immune system to fight off infections and illnesses.	100	17.99	3	3	HealthCare	NULL
127	JointFlex	Helps alleviate joint pain and stiffness for improved mobility.	100	22.99	3	3	HealthCare	NULL
128	StressLess	Reduces stress and promotes relaxation for mental well-being.	100	18.99	3	3	HealthCare	NULL
129	EyeCare	Supports healthy vision and protects against age-related eye diseases.	100	26.99	3	3	HealthCare	NULL
130	EnergyBoost	Provides a natural energy boost to combat fatigue and tiredness.	100	21.99	3	3	HealthCare	NULL
131	Cute Teddy Bear	Soft and cuddly teddy bear, perfect for kids to play and snuggle with.	100	15.99	3	4	Kids Section	NULL
132	Colorful Building Blocks	Set of colorful building blocks to encourage creativity and imagination in children.	100	12.49	3	4	Kids Section	NULL
133	Kids Coloring Book	Fun coloring book filled with engaging illustrations to keep kids entertained for hours.	100	8.99	3	4	Kids Section	NULL
134	Interactive Learning Tablet	Educational tablet for kids with interactive games and activities to enhance learning.	100	29.99	3	4	Kids Section	NULL
135	Musical Toy Piano	Miniature piano with colorful keys and musical notes to introduce kids to music.	100	19.99	3	4	Kids Section	NULL
136	Plush Animal Backpack	Adorable plush backpack designed in the shape of a favorite animal friend for kids.	100	17.99	3	4	Kids Section	NULL
137	Outdoor Play Tent	Pop-up play tent for outdoor adventures, providing hours of fun and imaginative play.	100	24.99	3	4	Kids Section	NULL
138	Remote Control Car	Remote control car with easy-to-use controls, perfect for indoor or outdoor play.	100	34.99	3	4	Kids Section	NULL
139	Educational Puzzle Set	Set of educational puzzles to develop problem-solving skills and	100	11.99	3	4	Kids Section	NULL

Output After

119	Trendy Scarf	Stylish scarf made from soft fabric, adds a touch of elegance to any outfit.	100	14.99	3	2	Fashion	NULL
120	Designer Perfume	Exquisite fragrance with a blend of floral and woody notes, perfect for special occasions.	100	89.99	3	2	Fashion	NULL
121	PainRelief	Effective pain relief medication for various types of pain.	100	15.99	3	3	HealthCare	NULL
122	VitaminBoost	A daily multivitamin to boost your overall health and immunity.	100	19.99	3	3	HealthCare	NULL
123	SleepWell	Natural sleep aid to promote restful and deep sleep.	100	19.99	3	3	HealthCare	NULL
124	DigestEase	Supports healthy digestion and relieves gastrointestinal discomfort.	100	12.99	3	3	HealthCare	NULL
125	HeartCare	Promotes cardiovascular health and maintains healthy cholesterol levels.	100	29.99	3	3	HealthCare	NULL
126	ImmuneGuard	Strengthens the immune system to fight off infections and illnesses.	100	17.99	3	3	HealthCare	NULL
127	JointFlex	Helps alleviate joint pain and stiffness for improved mobility.	100	22.99	3	3	HealthCare	NULL
128	StressLess	Reduces stress and promotes relaxation for mental well-being.	100	18.99	3	3	HealthCare	NULL
129	EyeCare	Supports healthy vision and protects against age-related eye diseases.	100	26.99	3	3	HealthCare	NULL
130	EnergyBoost	Provides a natural energy boost to combat fatigue and tiredness.	100	21.99	3	3	HealthCare	NULL
131	Cute Teddy Bear	Soft and cuddly teddy bear, perfect for kids to play and snuggle with.	100	15.99	3	4	Kids Section	NULL
132	Colorful Building Blocks	Set of colorful building blocks to encourage creativity and imagination in children.	100	12.49	3	4	Kids Section	NULL
133	Kids Coloring Book	Fun coloring book filled with engaging illustrations to keep kids entertained for hours.	100	8.99	3	4	Kids Section	NULL
134	Interactive Learning Tablet	Educational tablet for kids with interactive games and activities to enhance learning.	100	29.99	3	4	Kids Section	NULL
135	Musical Toy Piano	Miniature piano with colorful keys and musical notes to introduce kids to music.	100	19.99	3	4	Kids Section	NULL
136	Plush Animal Backpack	Adorable plush backpack designed in the shape of a favorite animal friend for kids.	100	17.99	3	4	Kids Section	NULL
137	Outdoor Play Tent	Pop-up play tent for outdoor adventures, providing hours of fun and imaginative play.	100	24.99	3	4	Kids Section	NULL
138	Remote Control Car	Remote control car with easy-to-use controls, perfect for indoor or outdoor play.	100	34.99	3	4	Kids Section	NULL

5) Update Customer Information:-

The procedure updates the `customer_name`, `customer_email`, `customer_phone`, `customer_address`, and `customer_gender` fields in the `Customers` table for the specified customer ID with the provided new values.

```
DELIMITER //
CREATE PROCEDURE UpdateCustomerInfo(
    IN customerId INT,
    IN newName VARCHAR(100),
    IN newEmail VARCHAR(100),
    IN newPhone VARCHAR(20),
    IN newAddress VARCHAR(255),
    IN newGender ENUM('male', 'female', 'other')
)
BEGIN
    UPDATE Customers
    SET customer_name = newName,
        customer_email = newEmail,
        customer_phone = newPhone,
        customer_address = newAddress,
        customer_gender = newGender
    WHERE customer_id = customerId;
END //
DELIMITER ;
```

CALL UpdateCustomerInfo(1,'John Doe', 'john.doe@example.com', '1234567890', '123 Main Street', 'male');

Code for calling function :

```
1  <?php
2  $servername = "localhost";
3  $username = "root";
4  $password = "Harsh@149";
5  $dbname = "walmart_india";
6  |
7  $conn = new mysqli($servername, $username, $password, $dbname);
8  if ($conn->connect_error) {
9      die("Connection failed: " . $conn->connect_error);
10 }
11
12 $customerId = $_GET['customer_id'];
13 $newName = $_GET['new_name'];
14 $newEmail = $_GET['new_email'];
15 $newPhone = $_GET['new_phone'];
16 $newAddress = $_GET['new_address'];
17 $newGender = $_GET['new_gender'];
18
19 $sql = "CALL UpdateCustomerInfo($customerId, '$newName', '$newEmail', '$newPhone', '$newAddress', '$newGender')";
20 if ($conn->query($sql) === TRUE) {
21     echo "Customer information updated successfully";
22 } else {
23     echo "Error updating customer information: " . $conn->error;
24 }
25 $conn->close();
26 ?>
27
```

Name:

Enter Name

Email:

Enter Email

Phone:

Enter Phone

Address:

Enter Address

Gender:



Male



Female



Other

UPDATE

Output Before

customer_id	customer_name	customer_gender	customer_email	customer_phone	customer_address
1	Vatsal Shah	male	vatsalshah@gmail.com	9876543210	Customer Address1, Mumbai, Maharashtra, India
2	Sahil Shah	male	sahilshah@gmail.com	9462038571	Customer Address2, Mumbai, Maharashtra, India
3	Aarya Parekh	male	aaryaparekh@gmail.com	9503728164	Customer Address3, Mumbai, Maharashtra, India
4	Prasham Shah	male	prashamsyah@gmail.com	9537284610	Customer Address4, Mumbai, Maharashtra, India
5	Kavyan Savani	male	kavyansavani@gmail.com	9473628105	Customer Address5, Mumbai, Maharashtra, India
6	Parth Zinzuwadia	male	parthzinzuwadia@gmail.com	9492837501	Customer Address6, Mumbai, Maharashtra, India
7	Diya Kasera	female	diyakasera@gmail.com	9457283019	Customer Address7, Mumbai, Maharashtra, India
8	Saumya Shah	female	saumyashah@gmail.com	9482730615	Customer Address8, Mumbai, Maharashtra, India
9	Aarya Gogia	male	aaryagogia@gmail.com	9471928360	Customer Address9, Mumbai, Maharashtra, India
10	Om Patel	male	ompatel@gmail.com	9457839026	Customer Address10, Mumbai, Maharashtra, India
11	Het Naik	male	hetnaik@gmail.com	9876543310	Customer Address1, Ahmedabad, Gujarat, India
12	Haard Patel	male	haardpatel@gmail.com	9876543320	Customer Address2, Ahmedabad, Gujarat, India
13	Prem Patel	male	prempatel@gmail.com	9876543330	Customer Address3, Ahmedabad, Gujarat, India
14	Karan Prajapati	male	karanprajapati@gmail.com	9876543340	Customer Address4, Ahmedabad, Gujarat, India
15	Jay Raval	male	jayraval@gmail.com	9876543350	Customer Address5, Ahmedabad, Gujarat, India
16	Vedant Parikh	male	vedantparikh@gmail.com	9876543360	Customer Address6, Ahmedabad, Gujarat, India
17	Shefali Naik	female	shefalinaike@gmail.com	9876543370	Customer Address7, Ahmedabad, Gujarat, India
18	Jainam	male	jainam@gmail.com	9876543380	Customer Address8, Ahmedabad, Gujarat, India
19	Anshi	female	anshi@gmail.com	9876543390	Customer Address9, Ahmedabad, Gujarat, India
20	Varun	male	varun@gmail.com	9876543400	Customer Address10, Ahmedabad, Gujarat, India
21	Suryaraj Jadeja	male	suryarajjadeja@gmail.com	9876543110	Customer Address1, Delhi, India
22	Dwarkesh Nasit	male	dwarkeshnasit@gmail.com	9876543120	Customer Address2, Delhi, India
23	Kahan Shah	male	kahanshah@gmail.com	9876543130	Customer Address3, Delhi, India
24	Vinisha Kankariya	female	vinishakankariya@gmail.com	9876543140	Customer Address4, Delhi, India
25	Divyam Shah	male	divyamshah@gmail.com	9876543150	Customer Address5, Delhi, India
26	Heshvi Davawala	female	heshvidavawala@gmail.com	9876543160	Customer Address6, Delhi, India

Output After

customer_id	customer_name	customer_gender	customer_email	customer_phone	customer_address
1	John Doe	male	john.doe@example.com	123 Main Street	
2	Sahil Shah	male	sahilshah@gmail.com	9462038571	Customer Address2, Mumbai, Maharashtra, India
3	Aarya Parekh	male	aaryaparekh@gmail.com	9503728164	Customer Address3, Mumbai, Maharashtra, India
4	Prasham Shah	male	prashamsyah@gmail.com	9537284610	Customer Address4, Mumbai, Maharashtra, India
5	Kavyan Savani	male	kavyansavani@gmail.com	9473628105	Customer Address5, Mumbai, Maharashtra, India
6	Parth Zinzuwadia	male	parthzinzuwadia@gmail.com	9492837501	Customer Address6, Mumbai, Maharashtra, India
7	Diya Kasera	female	diyakasera@gmail.com	9457283019	Customer Address7, Mumbai, Maharashtra, India
8	Saumya Shah	female	saumyashah@gmail.com	9482730615	Customer Address8, Mumbai, Maharashtra, India
9	Aarya Gogia	male	aaryagogia@gmail.com	9471928360	Customer Address9, Mumbai, Maharashtra, India
10	Om Patel	male	ompatel@gmail.com	9457839026	Customer Address10, Mumbai, Maharashtra, India
11	Het Naik	male	hetnaik@gmail.com	9876543310	Customer Address1, Ahmedabad, Gujarat, India
12	Haard Patel	male	haardpatel@gmail.com	9876543320	Customer Address2, Ahmedabad, Gujarat, India
13	Prem Patel	male	prempatel@gmail.com	9876543330	Customer Address3, Ahmedabad, Gujarat, India
14	Karan Prajapati	male	karanprajapati@gmail.com	9876543340	Customer Address4, Ahmedabad, Gujarat, India
15	Jay Raval	male	jayraval@gmail.com	9876543350	Customer Address5, Ahmedabad, Gujarat, India
16	Vedant Parikh	male	vedantparikh@gmail.com	9876543360	Customer Address6, Ahmedabad, Gujarat, India
17	Shefali Naik	female	shefalinaike@gmail.com	9876543370	Customer Address7, Ahmedabad, Gujarat, India
18	Jainam	male	jainam@gmail.com	9876543380	Customer Address8, Ahmedabad, Gujarat, India
19	Anshi	female	anshi@gmail.com	9876543390	Customer Address9, Ahmedabad, Gujarat, India
20	Varun	male	varun@gmail.com	9876543400	Customer Address10, Ahmedabad, Gujarat, India
21	Suryaraj Jadeja	male	suryarajjadeja@gmail.com	9876543110	Customer Address1, Delhi, India
22	Dwarkesh Nasit	male	dwarkeshnasit@gmail.com	9876543120	Customer Address2, Delhi, India
23	Kahan Shah	male	kahanshah@gmail.com	9876543130	Customer Address3, Delhi, India
24	Vinisha Kankariya	female	vinishakankariya@gmail.com	9876543140	Customer Address4, Delhi, India
25	Divyam Shah	male	divyamshah@gmail.com	9876543150	Customer Address5, Delhi, India
26	Heshvi Davawala	female	heshvidavawala@gmail.com	9876543160	Customer Address6, Delhi, India

6) Search Products:-

The procedure selects all columns (*) from the Products table where the product name contains the provided search keyword, the category type matches the provided category name, and the product price falls within the specified price range.

```
DELIMITER //
CREATE PROCEDURE SearchProducts
(
    IN searchKeyword VARCHAR(100),
    IN categoryName VARCHAR(50),
    IN minPrice DECIMAL(10,2),
    IN maxPrice DECIMAL(10,2)
)
BEGIN
    SELECT *
    FROM Products
    WHERE product_name LIKE CONCAT('%', searchKeyword, '%')
        AND category_type = categoryName
        AND product_price BETWEEN minPrice AND maxPrice;
END //
DELIMITER

CALL SearchProducts('shirt', 'Fashion', 10.00, 50.00);
```

Code for calling function :

```
1  <?php
2  $servername = "localhost";
3  $username = "root";
4  $password = "Harsh@149";
5  $dbname = "walmart_india";
6
7  $conn = new mysqli($servername, $username, $password, $dbname);
8  if ($conn->connect_error) {
9  |  die("Connection failed: " . $conn->connect_error);
10 }
11
12 $searchKeyword = $_GET['search_keyword'];
13 $categoryName = $_GET['category_name'];
14 $minPrice = $_GET['min_price'];
15 $maxPrice = $_GET['max_price'];
16
17 $sql = "CALL SearchProducts('$searchKeyword', '$categoryName', $minPrice, $maxPrice)";
18 $result = $conn->query($sql);
19
20 if ($result->num_rows > 0) {
21 |  $products = array();
22 |  while($row = $result->fetch_assoc()) {
23 |  |  $products[] = $row;
24 |  }
25  echo json_encode($products);
26 } else {
27 |  echo "0 results";
28 }
29 $conn->close();
30 ?>
31
```

Output

product_id	product_name	product_description	product_quantity	product_price	supplier_id	category_id	category_type	review_id
11	Casual Cotton T-Shirt	Comfortable and stylish cotton t-shirt for everyday wear.	98	19.99	1	2	Fashion	NULL
111	Casual Cotton T-Shirt	Comfortable and stylish cotton t-shirt for everyday wear.	98	19.99	3	2	Fashion	NULL
211	Casual Cotton T-Shirt	Comfortable and stylish cotton t-shirt for everyday wear.	97	19.99	5	2	Fashion	NULL
311	Casual Cotton T-Shirt	Comfortable and stylish cotton t-shirt for everyday wear.	98	19.99	7	2	Fashion	NULL
411	Casual Cotton T-Shirt	Comfortable and stylish cotton t-shirt for everyday wear.	98	19.99	9	2	Fashion	NULL

7) Get Top Selling Products:-

The procedure selects the product ID, product name, and the total quantity sold (sum of quantities from sales) for each product.

It joins the Sales table with the Products table using the product_id as the common field.

It groups the results by product ID and product name.

It orders the results by the total quantity sold in descending order.

It limits the number of results returned based on the limit_count parameter.

```
DELIMITER //
CREATE PROCEDURE GetTopSellingProducts(limit_count INT)
BEGIN
    SELECT p.product_id, p.product_name, SUM(s.quantity) AS total_sales
    FROM Sales s
    INNER JOIN Products p ON s.product_id = p.product_id
    GROUP BY p.product_id, p.product_name
    ORDER BY total_sales DESC
    LIMIT limit_count;
END//
DELIMITER ;
```

```
CALL GetTopSellingProducts(10);
```

Code for calling function :

```
C: > xampp > htdocs > walmart_india > procedure7.php
1  <?php
2  $conn = new mysqli($servername, $username, $password, $dbname);
3  if ($conn->connect_error) {
4  |   die("Connection failed: " . $conn->connect_error);
5  }
6
7  $limitCount = $_GET['limit_count'];
8
9  $sql = "CALL GetTopSellingProducts($limitCount)";
10 $result = $conn->query($sql);
11
12 if ($result->num_rows > 0) {
13 |   $products = array();
14 |   while($row = $result->fetch_assoc()) {
15 |     |   $products[] = $row;
16 |   }
17 |   echo json_encode($products);
18 } else {
19 |   echo "0 results";
20 }
21 $conn->close();
22 ?>
23
```

Output



The screenshot shows a web browser window with the URL `http://localhost/walmart_india/procedure7.php` in the address bar. The page displays a table with three columns: `product_id`, `product_name`, and `total_sales`. The table contains 10 rows of data, each with a product ID, name, and sales count of 5.

product_id	product_name	total_sales
352	Explorer Backpack	5
162	Non-Stick Frying Pan	5
192	Feline Delight Cat Litter	5
182	Refreshing Facial Cleanser	5
372	Pasta	5
362	Non-Stick Frying Pan	5
172	Pasta	5
152	Explorer Backpack	5
382	Refreshing Facial Cleanser	5
392	Feline Delight Cat Litter	5

8) Update Product Quantity:-

The procedure updates the product_quantity field in the Products table for the specified product ID with the provided new quantity.

```
DELIMITER //
CREATE PROCEDURE UpdateProductQuantity
(
    IN product_id_param INT,
    IN new_quantity INT
)
BEGIN
    UPDATE Products SET product_quantity = new_quantity WHERE
product_id = product_id_param;
END //
DELIMITER ;
```

```
CALL UpdateProductQuantity(123, 50);
```

Code for calling function :

```
C: > xampp > htdocs > walmart_india > procedure8_1.php
 1  <?php
 2  $conn = new mysqli($servername, $username, $password, $dbname);
 3  if ($conn->connect_error) {
 4  |   die("Connection failed: " . $conn->connect_error);
 5  }
 6
 7  $product_id = $_GET['product_id'];
 8  $new_quantity = $_GET['new_quantity'];
 9
10 $sql = "CALL UpdateProductQuantity($product_id, $new_quantity)";
11 if ($conn->query($sql) === TRUE) {
12 |   echo "Product quantity updated successfully";
13 } else {
14 |   echo "Error updating product quantity: " . $conn->error;
15 }
16 $conn->close();
17 ?>
18
```

SleepWell

Natural Sleep Aid to promote restful and deep sleep



Quantity: 100 Price: 24.99

[Want to Update?](#)

Digest Ease

Natural Digestive Aid for a Comfortable Stomach



Quantity: 100 Price: 12.99

[Want to Update?](#)

Product Name:

SleepWell

Product Price:

24.99

New Price:

24.99

Product Quantity:

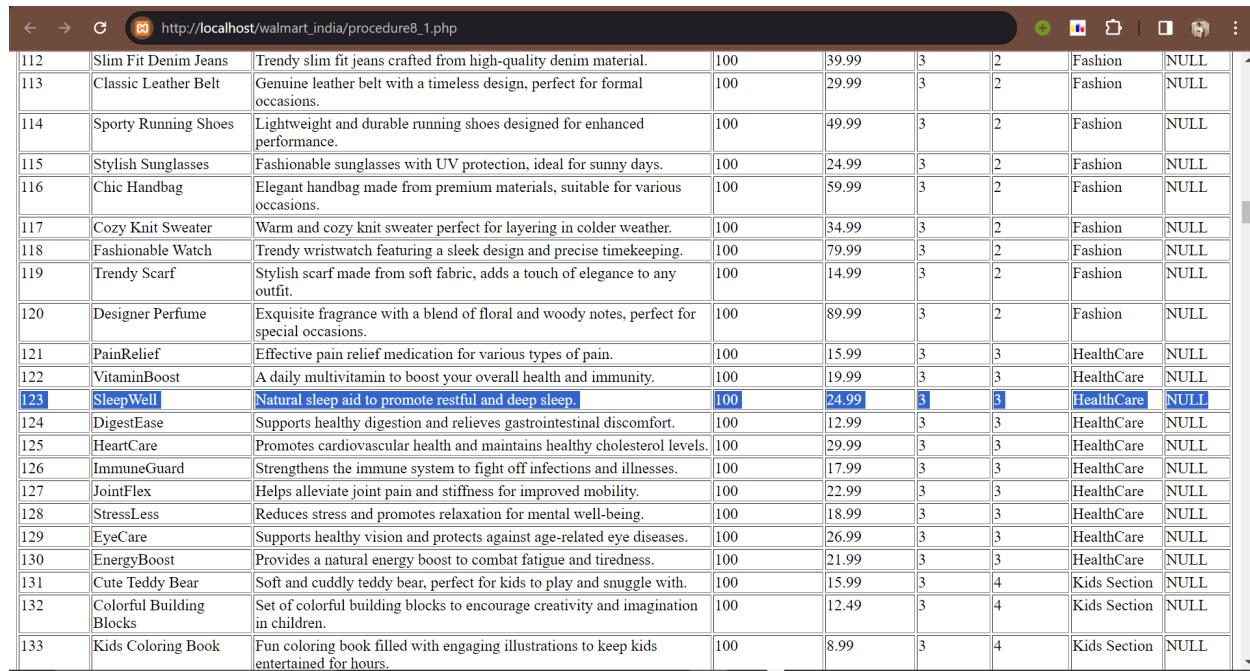
100

New Quantity:

50

UPDATE

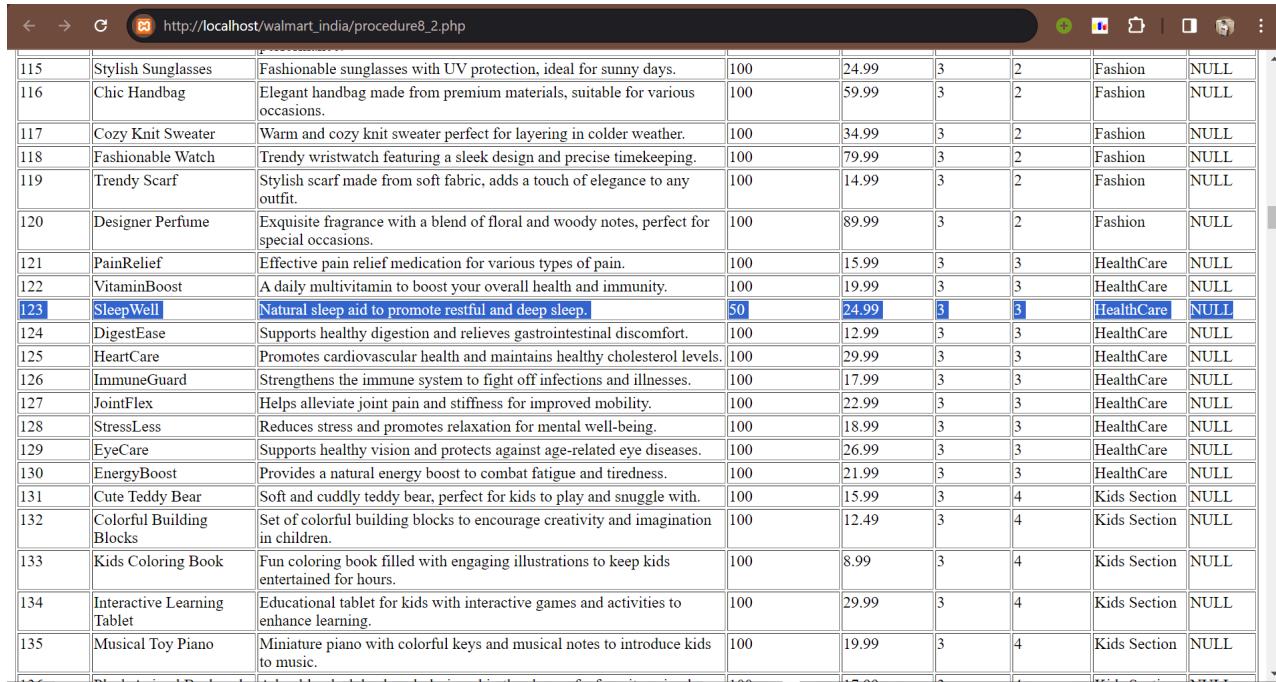
Output Before



The screenshot shows a web browser window with the URL http://localhost/walmart_india/procedure8_1.php. The page displays a table of product information. The columns are: ID, Product Name, Description, Stock Level, Price, Rating, Reviews, Category, and Status. The table contains 33 rows of data.

112	Slim Fit Denim Jeans	Trendy slim fit jeans crafted from high-quality denim material.	100	39.99	3	2	Fashion	NULL
113	Classic Leather Belt	Genuine leather belt with a timeless design, perfect for formal occasions.	100	29.99	3	2	Fashion	NULL
114	Sporty Running Shoes	Lightweight and durable running shoes designed for enhanced performance.	100	49.99	3	2	Fashion	NULL
115	Stylish Sunglasses	Fashionable sunglasses with UV protection, ideal for sunny days.	100	24.99	3	2	Fashion	NULL
116	Chic Handbag	Elegant handbag made from premium materials, suitable for various occasions.	100	59.99	3	2	Fashion	NULL
117	Cozy Knit Sweater	Warm and cozy knit sweater perfect for layering in colder weather.	100	34.99	3	2	Fashion	NULL
118	Fashionable Watch	Trendy wristwatch featuring a sleek design and precise timekeeping.	100	79.99	3	2	Fashion	NULL
119	Trendy Scarf	Stylish scarf made from soft fabric, adds a touch of elegance to any outfit.	100	14.99	3	2	Fashion	NULL
120	Designer Perfume	Exquisite fragrance with a blend of floral and woody notes, perfect for special occasions.	100	89.99	3	2	Fashion	NULL
121	PainRelief	Effective pain relief medication for various types of pain.	100	15.99	3	3	HealthCare	NULL
122	VitaminBoost	A daily multivitamin to boost your overall health and immunity.	100	19.99	3	3	HealthCare	NULL
123	SleepWell	Natural sleep aid to promote restful and deep sleep.	100	24.99	3	3	HealthCare	NULL
124	DigestEase	Supports healthy digestion and relieves gastrointestinal discomfort.	100	12.99	3	3	HealthCare	NULL
125	HeartCare	Promotes cardiovascular health and maintains healthy cholesterol levels.	100	29.99	3	3	HealthCare	NULL
126	ImmuneGuard	Strengthens the immune system to fight off infections and illnesses.	100	17.99	3	3	HealthCare	NULL
127	JointFlex	Helps alleviate joint pain and stiffness for improved mobility.	100	22.99	3	3	HealthCare	NULL
128	StressLess	Reduces stress and promotes relaxation for mental well-being.	100	18.99	3	3	HealthCare	NULL
129	EyeCare	Supports healthy vision and protects against age-related eye diseases.	100	26.99	3	3	HealthCare	NULL
130	EnergyBoost	Provides a natural energy boost to combat fatigue and tiredness.	100	21.99	3	3	HealthCare	NULL
131	Cute Teddy Bear	Soft and cuddly teddy bear, perfect for kids to play and snuggle with.	100	15.99	3	4	Kids Section	NULL
132	Colorful Building Blocks	Set of colorful building blocks to encourage creativity and imagination in children.	100	12.49	3	4	Kids Section	NULL
133	Kids Coloring Book	Fun coloring book filled with engaging illustrations to keep kids entertained for hours.	100	8.99	3	4	Kids Section	NULL

Output After



The screenshot shows a web browser window with the URL http://localhost/walmart_india/procedure8_2.php. The page displays a table of product information. The table has columns for Product ID, Name, Description, Stock Level, Price, Rating, Reviews, Category, and Status. A specific row, number 123, is highlighted with a blue background. The table data is as follows:

115	Stylish Sunglasses	Fashionable sunglasses with UV protection, ideal for sunny days.	100	24.99	3	2	Fashion	NULL	
116	Chic Handbag	Elegant handbag made from premium materials, suitable for various occasions.	100	59.99	3	2	Fashion	NULL	
117	Cozy Knit Sweater	Warm and cozy knit sweater perfect for layering in colder weather.	100	34.99	3	2	Fashion	NULL	
118	Fashionable Watch	Trendy wristwatch featuring a sleek design and precise timekeeping.	100	79.99	3	2	Fashion	NULL	
119	Trendy Scarf	Stylish scarf made from soft fabric, adds a touch of elegance to any outfit.	100	14.99	3	2	Fashion	NULL	
120	Designer Perfume	Exquisite fragrance with a blend of floral and woody notes, perfect for special occasions.	100	89.99	3	2	Fashion	NULL	
121	PainRelief	Effective pain relief medication for various types of pain.	100	15.99	3	3	HealthCare	NULL	
122	VitaminBoost	A daily multivitamin to boost your overall health and immunity.	100	19.99	3	3	HealthCare	NULL	
123	SleepWell	Natural sleep aid to promote restful and deep sleep.	50	24.99	3	3	HealthCare	NULL	
124	DigestEase	Supports healthy digestion and relieves gastrointestinal discomfort.	100	12.99	3	3	HealthCare	NULL	
125	HeartCare	Promotes cardiovascular health and maintains healthy cholesterol levels.	100	29.99	3	3	HealthCare	NULL	
126	ImmuneGuard	Strengthens the immune system to fight off infections and illnesses.	100	17.99	3	3	HealthCare	NULL	
127	JointFlex	Helps alleviate joint pain and stiffness for improved mobility.	100	22.99	3	3	HealthCare	NULL	
128	StressLess	Reduces stress and promotes relaxation for mental well-being.	100	18.99	3	3	HealthCare	NULL	
129	EyeCare	Supports healthy vision and protects against age-related eye diseases.	100	26.99	3	3	HealthCare	NULL	
130	EnergyBoost	Provides a natural energy boost to combat fatigue and tiredness.	100	21.99	3	3	HealthCare	NULL	
131	Cute Teddy Bear	Soft and cuddly teddy bear, perfect for kids to play and snuggle with.	100	15.99	3	4	Kids Section	NULL	
132	Colorful Building Blocks	Set of colorful building blocks to encourage creativity and imagination in children.	100	12.49	3	4	Kids Section	NULL	
133	Kids Coloring Book	Fun coloring book filled with engaging illustrations to keep kids entertained for hours.	100	8.99	3	4	Kids Section	NULL	
134	Interactive Learning Tablet	Educational tablet for kids with interactive games and activities to enhance learning.	100	29.99	3	4	Kids Section	NULL	
135	Musical Toy Piano	Miniature piano with colorful keys and musical notes to introduce kids to music.	100	19.99	3	4	Kids Section	NULL	

9) Check Stock Availability:-

The function reads the current quantity available in stock (product_quantity) for the specified product ID from the Products table.

It compares the available quantity with the required quantity.

If the available quantity is greater than or equal to the required quantity, it returns TRUE, indicating that the required quantity is available in stock. Otherwise, it returns FALSE.

```
DELIMITER $$  
CREATE FUNCTION CheckStockAvailability(product_id_param INT,  
required_quantity INT) RETURNS BOOLEAN  
READS SQL DATA  
BEGIN  
DECLARE available_quantity INT;  
SELECT product_quantity INTO available_quantity FROM Products WHERE  
product_id = product_id_param;  
IF available_quantity >= required_quantity THEN  
    RETURN TRUE;  
ELSE  
    RETURN FALSE;  
END IF;  
END$$  
DELIMITER ;
```

```
SELECT CheckStockAvailability(123, 20) AS is_available;  
SELECT CheckStockAvailability(122, 120) AS is_available;
```

Code for calling function :

```
C:\xampp\htdocs\walmart_india>function1.php
1  <?php
2  $conn = new mysqli($servername, $username, $password, $dbname);
3  if ($conn->connect_error) {
4  |   die("Connection failed: " . $conn->connect_error);
5  }
6
7  $product_id = $_GET['product_id'];
8  $required_quantity = $_GET['required_quantity'];
9
10 $sql = "SELECT CheckStockAvailability($product_id, $required_quantity) AS availability";
11 $result = $conn->query($sql);
12
13 if ($result->num_rows > 0) {
14 |   $row = $result->fetch_assoc();
15 |   $availability = $row['availability'];
16 |   echo json_encode(["availability" => (bool)$availability]);
17 } else {
18 |   echo "0 results";
19 }
20 $conn->close();
21 ?>
22
```

Output





10) GetMonthlySalesByStore:-

The function calculates the total sales amount by summing up the total_amount column from the Sales table for the specified store, month, and year.

```
DELIMITER $$  
CREATE FUNCTION GetMonthlySalesByStore(store_number int,  
month_number int, year_number int) RETURNS DECIMAL(10, 2)  
READS SQL DATA  
BEGIN  
    DECLARE total_sales DECIMAL(10, 2);  
  
    SELECT SUM(total_amount) INTO total_sales  
    FROM Sales  
    WHERE store_id=store_number and month(sale_date)=month_number and  
year(sale_date)=year_number;  
  
    RETURN total_sales;  
END$$  
DELIMITER ;
```

```
SELECT GetMonthlySalesByStore(1,4,2024) AS monthly_sales;  
SELECT GetMonthlySalesByStore(2,4,2024) AS monthly_sales;  
SELECT GetMonthlySalesByStore(3,4,2024) AS monthly_sales;  
SELECT GetMonthlySalesByStore(4,4,2024) AS monthly_sales;  
SELECT GetMonthlySalesByStore(5,4,2024) AS monthly_sales;
```

Code for calling function :

```
C:\>xampp>htdocs>walmart_india>function2.php
1  <?php
2  $conn = new mysqli($servername, $username, $password, $dbname);
3  if ($conn->connect_error) {
4      die("Connection failed: " . $conn->connect_error);
5  }
6
7  $store_number = $_GET['store_number'];
8  $month_number = $_GET['month_number'];
9  $year_number = $_GET['year_number'];
10
11 $sql = "SELECT GetMonthlySalesByStore($store_number, $month_number, $year_number) AS monthly_sales";
12 $result = $conn->query($sql);
13
14 if ($result->num_rows > 0) {
15     $row = $result->fetch_assoc();
16     $monthly_sales = $row['monthly_sales'];
17     echo json_encode(["monthly_sales" => $monthly_sales]);
18 } else {
19     echo "0 results";
20 }
21 $conn->close();
22 ?>
23
```

Output





monthly_sales

4015.69



monthly_sales

3791.08



11) GetMonthlyTransactionsByStore:-

The function calculates the total number of transactions by counting the records in the Transactions table for the specified store, month, and year.

```
DELIMITER $$  
CREATE FUNCTION GetMonthlyTransactionsByStore(store_number INT,  
month_number INT, year_number INT) RETURNS INT  
READS SQL DATA  
BEGIN  
    DECLARE total_transactions INT;  
  
    SELECT COUNT(*) INTO total_transactions  
    FROM Transactions  
    WHERE store_id = store_number AND MONTH(transaction_date) =  
month_number AND YEAR(transaction_date) = year_number;  
  
    RETURN total_transactions;  
END$$  
DELIMITER ;  
  
SELECT GetMonthlyTransactionsByStore(1, 4, 2024);  
SELECT GetMonthlyTransactionsByStore(2, 4, 2024);  
SELECT GetMonthlyTransactionsByStore(3, 4, 2024);  
SELECT GetMonthlyTransactionsByStore(4, 4, 2024);  
SELECT GetMonthlyTransactionsByStore(5, 4, 2024);
```

Code for calling function :

```
C: > xampp > htdocs > walmart_india > function3.php
1  <?php
2  $conn = new mysqli($servername, $username, $password, $dbname);
3  if ($conn->connect_error) {
4      die("Connection failed: " . $conn->connect_error);
5  }
6
7  $store_number = $_GET['store_number'];
8  $month_number = $_GET['month_number'];
9  $year_number = $_GET['year_number'];
10
11 $sql = "SELECT GetMonthlyTransactionsByStore($store_number, $month_number, $year_number) AS total_transactions";
12 $result = $conn->query($sql);
13
14 if ($result->num_rows > 0) {
15     $row = $result->fetch_assoc();
16     $total_transactions = $row['total_transactions'];
17     echo json_encode([ "total_transactions" => $total_transactions]);
18 } else {
19     echo "0 results";
20 }
21 $conn->close();
22 ?>
23
```

Output







← → ⌛ http://localhost/walmart_india/procedure11_4.php

GetMonthlyTransactionsByStore(4, 4, 2024)
20



← → ⌛ http://localhost/walmart_india/procedure11_5.php

GetMonthlyTransactionsByStore(5, 4, 2024)
20

5 Triggers

→ CalculateTaxCOGSGrossMarginIncomeTrigger:-

The trigger fires before a new row is inserted into the Transactions table.

```
DELIMITER //
CREATE TRIGGER CalculateTaxCOGSGrossMarginIncomeTrigger BEFORE
INSERT ON
Transactions
FOR EACH ROW
BEGIN
    DECLARE tax_val DECIMAL(10, 2);
    DECLARE cogs_val DECIMAL(10, 2);
    DECLARE gross_margin_val DECIMAL(10, 2);
    DECLARE gross_income_val DECIMAL(10, 2);

    -- Calculate tax
    SET tax_val = NEW.unit_price * NEW.quantity * 0.18;
    -- Calculate COGS
    SET cogs_val = NEW.unit_price * NEW.quantity;

    -- Calculate total amount (unit price * quantity + tax)
    SET NEW.total_amount = cogs_val + tax_val;

    -- Calculate gross income
    SET gross_income_val = NEW.total_amount - cogs_val;

    -- Calculate gross margin
    IF NEW.total_amount > 0 THEN
        SET gross_margin_val = ((NEW.total_amount - cogs_val) /
        NEW.total_amount) * 100;
    ELSE
        SET gross_margin_val = 0;
```

END IF;

-- Update tax

SET NEW.tax = tax_val;

-- Update COGS

SET NEW.cogs = cogs_val;

-- Update gross margin

SET NEW.gross_margin = gross_margin_val;

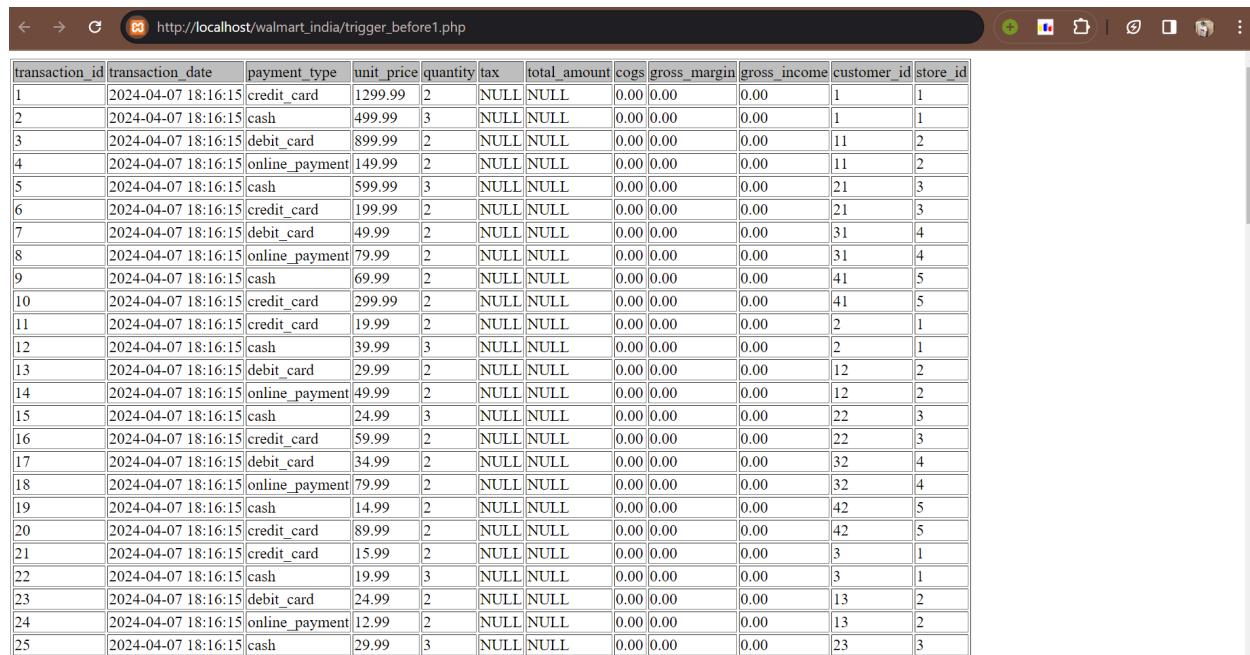
-- Update gross income

SET NEW.gross_income = gross_income_val;

END //

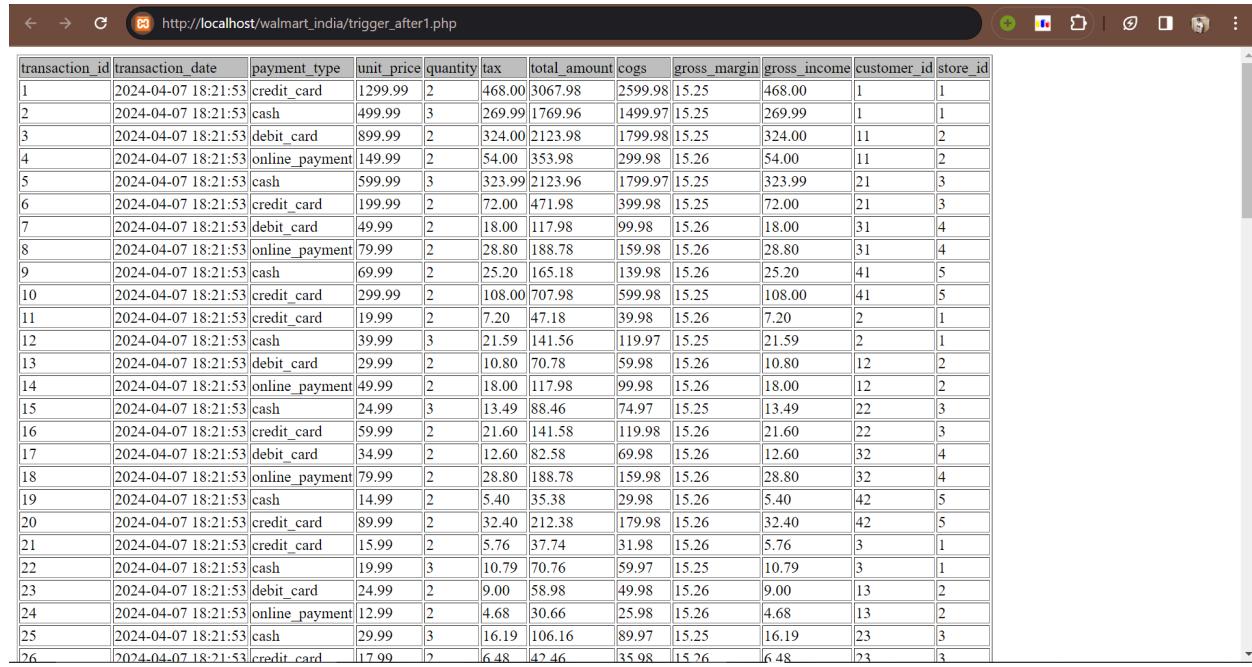
DELIMITER ;

Output Before



transaction_id	transaction_date	payment_type	unit_price	quantity	tax	total_amount	cogs	gross_margin	gross_income	customer_id	store_id
1	2024-04-07 18:16:15	credit_card	1299.99	2	NULL	NULL	0.00	0.00	0.00	1	1
2	2024-04-07 18:16:15	cash	499.99	3	NULL	NULL	0.00	0.00	0.00	1	1
3	2024-04-07 18:16:15	debit_card	899.99	2	NULL	NULL	0.00	0.00	0.00	11	2
4	2024-04-07 18:16:15	online_payment	149.99	2	NULL	NULL	0.00	0.00	0.00	11	2
5	2024-04-07 18:16:15	cash	599.99	3	NULL	NULL	0.00	0.00	0.00	21	3
6	2024-04-07 18:16:15	credit_card	199.99	2	NULL	NULL	0.00	0.00	0.00	21	3
7	2024-04-07 18:16:15	debit_card	49.99	2	NULL	NULL	0.00	0.00	0.00	31	4
8	2024-04-07 18:16:15	online_payment	79.99	2	NULL	NULL	0.00	0.00	0.00	31	4
9	2024-04-07 18:16:15	cash	69.99	2	NULL	NULL	0.00	0.00	0.00	41	5
10	2024-04-07 18:16:15	credit_card	299.99	2	NULL	NULL	0.00	0.00	0.00	41	5
11	2024-04-07 18:16:15	credit_card	19.99	2	NULL	NULL	0.00	0.00	0.00	2	1
12	2024-04-07 18:16:15	cash	39.99	3	NULL	NULL	0.00	0.00	0.00	2	1
13	2024-04-07 18:16:15	debit_card	29.99	2	NULL	NULL	0.00	0.00	0.00	12	2
14	2024-04-07 18:16:15	online_payment	49.99	2	NULL	NULL	0.00	0.00	0.00	12	2
15	2024-04-07 18:16:15	cash	24.99	3	NULL	NULL	0.00	0.00	0.00	22	3
16	2024-04-07 18:16:15	credit_card	59.99	2	NULL	NULL	0.00	0.00	0.00	22	3
17	2024-04-07 18:16:15	debit_card	34.99	2	NULL	NULL	0.00	0.00	0.00	32	4
18	2024-04-07 18:16:15	online_payment	79.99	2	NULL	NULL	0.00	0.00	0.00	32	4
19	2024-04-07 18:16:15	cash	14.99	2	NULL	NULL	0.00	0.00	0.00	42	5
20	2024-04-07 18:16:15	credit_card	89.99	2	NULL	NULL	0.00	0.00	0.00	42	5
21	2024-04-07 18:16:15	credit_card	15.99	2	NULL	NULL	0.00	0.00	0.00	3	1
22	2024-04-07 18:16:15	cash	19.99	3	NULL	NULL	0.00	0.00	0.00	3	1
23	2024-04-07 18:16:15	debit_card	24.99	2	NULL	NULL	0.00	0.00	0.00	13	2
24	2024-04-07 18:16:15	online_payment	12.99	2	NULL	NULL	0.00	0.00	0.00	13	2
25	2024-04-07 18:16:15	cash	29.99	3	NULL	NULL	0.00	0.00	0.00	23	3

Output After



The screenshot shows a web browser window with the URL http://localhost/walmart_india/trigger_after1.php. The page displays a table of transaction data with the following columns: transaction_id, transaction_date, payment_type, unit_price, quantity, tax, total_amount, cogs, gross_margin, gross_income, customer_id, and store_id. The data consists of 26 rows, each representing a transaction record. The table is styled with alternating row colors.

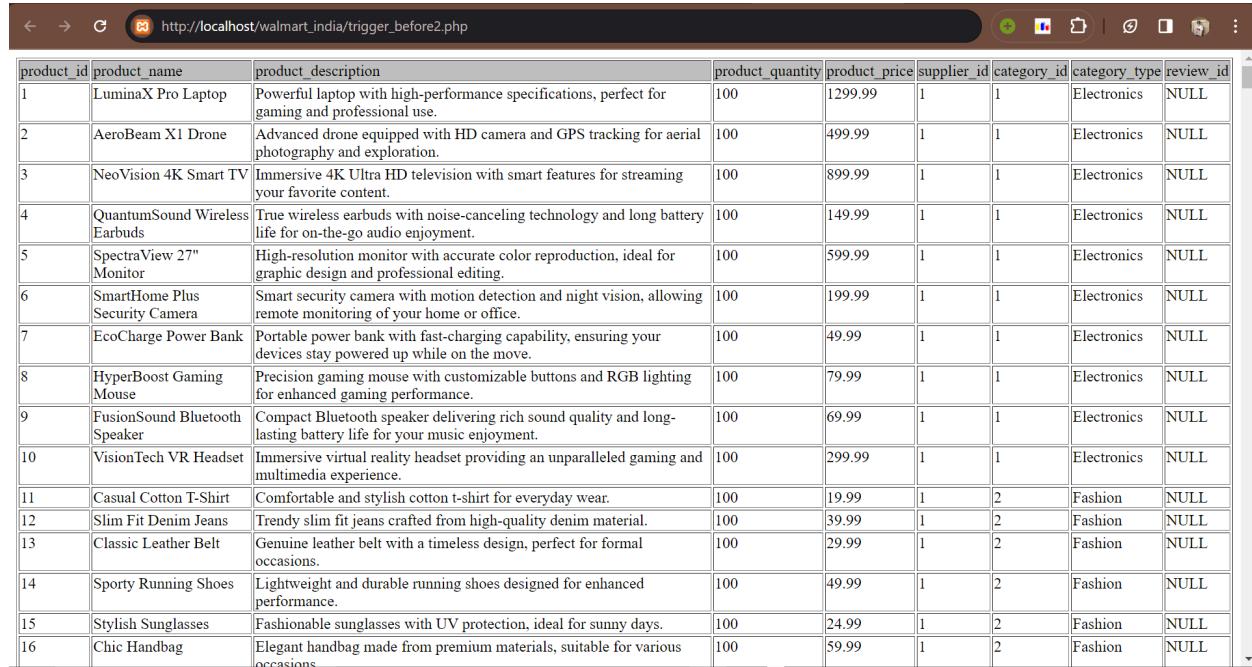
transaction_id	transaction_date	payment_type	unit_price	quantity	tax	total_amount	cogs	gross_margin	gross_income	customer_id	store_id
1	2024-04-07 18:21:53	credit_card	1299.99	2	468.00	3067.98	2599.98	15.25	468.00	1	1
2	2024-04-07 18:21:53	cash	499.99	3	269.99	1769.96	1499.97	15.25	269.99	1	1
3	2024-04-07 18:21:53	debit_card	899.99	2	324.00	2123.98	1799.98	15.25	324.00	11	2
4	2024-04-07 18:21:53	online_payment	149.99	2	54.00	353.98	299.98	15.26	54.00	11	2
5	2024-04-07 18:21:53	cash	599.99	3	323.99	2123.96	1799.97	15.25	323.99	21	3
6	2024-04-07 18:21:53	credit_card	199.99	2	72.00	471.98	399.98	15.25	72.00	21	3
7	2024-04-07 18:21:53	debit_card	49.99	2	18.00	117.98	99.98	15.26	18.00	31	4
8	2024-04-07 18:21:53	online_payment	79.99	2	28.80	188.78	159.98	15.26	28.80	31	4
9	2024-04-07 18:21:53	cash	69.99	2	25.20	165.18	139.98	15.26	25.20	41	5
10	2024-04-07 18:21:53	credit_card	299.99	2	108.00	707.98	599.98	15.25	108.00	41	5
11	2024-04-07 18:21:53	credit_card	19.99	2	7.20	47.18	39.98	15.26	7.20	2	1
12	2024-04-07 18:21:53	cash	39.99	3	15.99	141.56	119.97	15.25	21.59	2	1
13	2024-04-07 18:21:53	debit_card	29.99	2	10.80	70.78	59.98	15.26	10.80	12	2
14	2024-04-07 18:21:53	online_payment	49.99	2	18.00	117.98	99.98	15.26	18.00	12	2
15	2024-04-07 18:21:53	cash	24.99	3	13.49	88.46	74.97	15.25	13.49	22	3
16	2024-04-07 18:21:53	credit_card	59.99	2	21.60	141.58	119.98	15.26	21.60	22	3
17	2024-04-07 18:21:53	debit_card	34.99	2	12.60	82.58	69.98	15.26	12.60	32	4
18	2024-04-07 18:21:53	online_payment	79.99	2	28.80	188.78	159.98	15.26	28.80	32	4
19	2024-04-07 18:21:53	cash	14.99	2	5.40	35.38	29.98	15.26	5.40	42	5
20	2024-04-07 18:21:53	credit_card	89.99	2	32.40	212.38	179.98	15.26	32.40	42	5
21	2024-04-07 18:21:53	credit_card	15.99	2	5.76	37.74	31.98	15.26	5.76	3	1
22	2024-04-07 18:21:53	cash	19.99	3	10.79	70.76	59.97	15.25	10.79	3	1
23	2024-04-07 18:21:53	debit_card	24.99	2	9.00	58.98	49.98	15.26	9.00	13	2
24	2024-04-07 18:21:53	online_payment	12.99	2	4.68	30.66	25.98	15.26	4.68	13	2
25	2024-04-07 18:21:53	cash	29.99	3	16.19	106.16	89.97	15.25	16.19	23	3
26	2024-04-07 18:21:53	credit_card	17.99	2	6.48	42.46	35.98	15.26	6.48	23	3

→ UpdateProductQuantity:-

The trigger fires after a new row is inserted into the Sales table.

```
DELIMITER $$  
CREATE TRIGGER AfterSaleInsert  
AFTER INSERT ON Sales  
FOR EACH ROW  
BEGIN  
    UPDATE Products  
    SET product_quantity = product_quantity - NEW.quantity  
    WHERE product_id = NEW.product_id;  
END$$  
DELIMITER ;
```

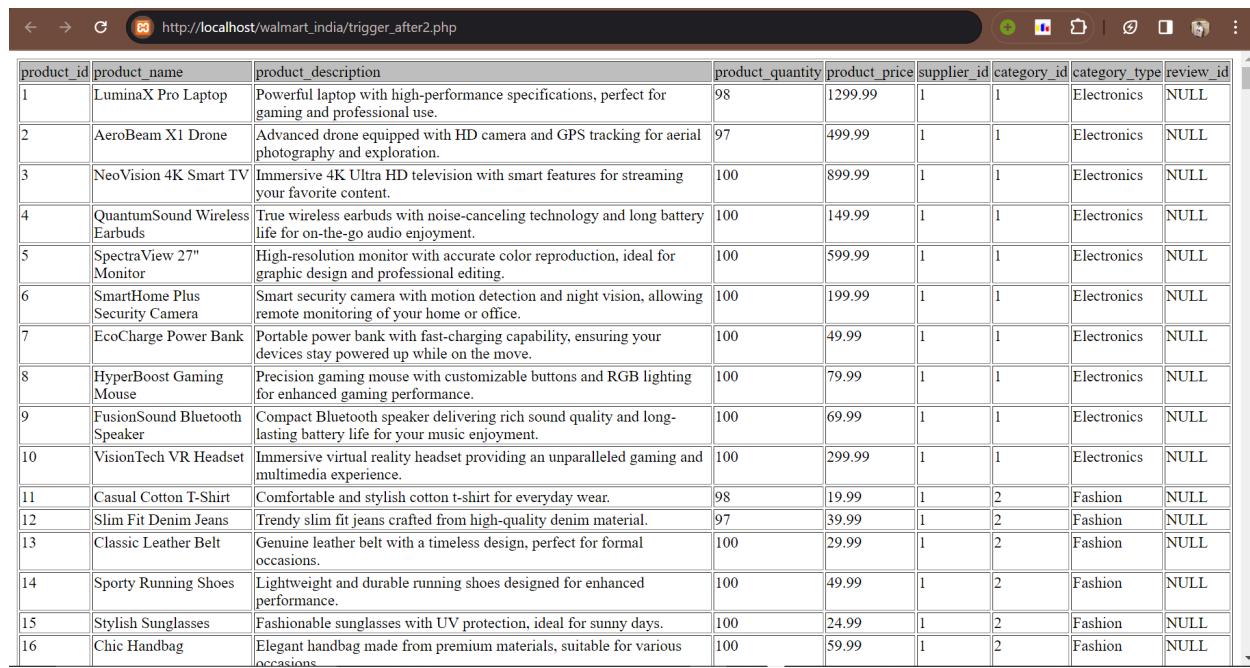
Output Before



A screenshot of a web browser window displaying a table of product data. The URL in the address bar is http://localhost/walmart_india/trigger_before2.php. The table has 9 columns: product_id, product_name, product_description, product_quantity, product_price, supplier_id, category_id, category_type, and review_id. The data consists of 16 rows, each representing a different product with its details.

product_id	product_name	product_description	product_quantity	product_price	supplier_id	category_id	category_type	review_id
1	LuminaX Pro Laptop	Powerful laptop with high-performance specifications, perfect for gaming and professional use.	100	1299.99	1	1	Electronics	NULL
2	AeroBeam X1 Drone	Advanced drone equipped with HD camera and GPS tracking for aerial photography and exploration.	100	499.99	1	1	Electronics	NULL
3	NeoVision 4K Smart TV	Immersive 4K Ultra HD television with smart features for streaming your favorite content.	100	899.99	1	1	Electronics	NULL
4	QuantumSound Wireless Earbuds	True wireless earbuds with noise-cancelling technology and long battery life for on-the-go audio enjoyment.	100	149.99	1	1	Electronics	NULL
5	SpectraView 27" Monitor	High-resolution monitor with accurate color reproduction, ideal for graphic design and professional editing.	100	599.99	1	1	Electronics	NULL
6	SmartHome Plus Security Camera	Smart security camera with motion detection and night vision, allowing remote monitoring of your home or office.	100	199.99	1	1	Electronics	NULL
7	EcoCharge Power Bank	Portable power bank with fast-charging capability, ensuring your devices stay powered up while on the move.	100	49.99	1	1	Electronics	NULL
8	HyperBoost Gaming Mouse	Precision gaming mouse with customizable buttons and RGB lighting for enhanced gaming performance.	100	79.99	1	1	Electronics	NULL
9	FusionSound Bluetooth Speaker	Compact Bluetooth speaker delivering rich sound quality and long-lasting battery life for your music enjoyment.	100	69.99	1	1	Electronics	NULL
10	VisionTech VR Headset	Immersive virtual reality headset providing an unparalleled gaming and multimedia experience.	100	299.99	1	1	Electronics	NULL
11	Casual Cotton T-Shirt	Comfortable and stylish cotton t-shirt for everyday wear.	100	19.99	1	2	Fashion	NULL
12	Slim Fit Denim Jeans	Trendy slim fit jeans crafted from high-quality denim material.	100	39.99	1	2	Fashion	NULL
13	Classic Leather Belt	Genuine leather belt with a timeless design, perfect for formal occasions.	100	29.99	1	2	Fashion	NULL
14	Sporty Running Shoes	Lightweight and durable running shoes designed for enhanced performance.	100	49.99	1	2	Fashion	NULL
15	Stylish Sunglasses	Fashionable sunglasses with UV protection, ideal for sunny days.	100	24.99	1	2	Fashion	NULL
16	Chic Handbag	Elegant handbag made from premium materials, suitable for various occasions.	100	59.99	1	2	Fashion	NULL

Output After



A screenshot of a web browser window displaying a table of product data. The URL in the address bar is http://localhost/walmart_india/trigger_after2.php. The table has 16 rows and 9 columns. The columns are: product_id, product_name, product_description, product_quantity, product_price, supplier_id, category_id, category_type, and review_id. The data includes various products like laptops, drones, TVs, earbuds, monitors, cameras, power banks, mice, speakers, VR headsets, clothing, belts, shoes, sunglasses, and handbags, categorized into Electronics and Fashion.

product_id	product_name	product_description	product_quantity	product_price	supplier_id	category_id	category_type	review_id
1	LuminaX Pro Laptop	Powerful laptop with high-performance specifications, perfect for gaming and professional use.	98	1299.99	1	1	Electronics	NULL
2	AeroBeam X1 Drone	Advanced drone equipped with HD camera and GPS tracking for aerial photography and exploration.	97	499.99	1	1	Electronics	NULL
3	NeoVision 4K Smart TV	Immersive 4K Ultra HD television with smart features for streaming your favorite content.	100	899.99	1	1	Electronics	NULL
4	QuantumSound Wireless Earbuds	True wireless earbuds with noise-canceling technology and long battery life for on-the-go audio enjoyment.	100	149.99	1	1	Electronics	NULL
5	SpectraView 27" Monitor	High-resolution monitor with accurate color reproduction, ideal for graphic design and professional editing.	100	599.99	1	1	Electronics	NULL
6	SmartHome Plus Security Camera	Smart security camera with motion detection and night vision, allowing remote monitoring of your home or office.	100	199.99	1	1	Electronics	NULL
7	EcoCharge Power Bank	Portable power bank with fast-charging capability, ensuring your devices stay powered up while on the move.	100	49.99	1	1	Electronics	NULL
8	HyperBoost Gaming Mouse	Precision gaming mouse with customizable buttons and RGB lighting for enhanced gaming performance.	100	79.99	1	1	Electronics	NULL
9	FusionSound Bluetooth Speaker	Compact Bluetooth speaker delivering rich sound quality and long-lasting battery life for your music enjoyment.	100	69.99	1	1	Electronics	NULL
10	VisionTech VR Headset	Immersive virtual reality headset providing an unparalleled gaming and multimedia experience.	100	299.99	1	1	Electronics	NULL
11	Casual Cotton T-Shirt	Comfortable and stylish cotton t-shirt for everyday wear.	98	19.99	1	2	Fashion	NULL
12	Slim Fit Denim Jeans	Trendy slim fit jeans crafted from high-quality denim material.	97	39.99	1	2	Fashion	NULL
13	Classic Leather Belt	Genuine leather belt with a timeless design, perfect for formal occasions.	100	29.99	1	2	Fashion	NULL
14	Sporty Running Shoes	Lightweight and durable running shoes designed for enhanced performance.	100	49.99	1	2	Fashion	NULL
15	Stylish Sunglasses	Fashionable sunglasses with UV protection, ideal for sunny days.	100	24.99	1	2	Fashion	NULL
16	Chic Handbag	Elegant handbag made from premium materials, suitable for various occasions.	100	59.99	1	2	Fashion	NULL

→ EnforceMinimumProductPrice:-

The trigger fires before a new row is inserted into the Products table.

```
DELIMITER //
CREATE TRIGGER EnforceMinimumProductPrice BEFORE INSERT ON
Products
FOR EACH ROW
BEGIN
    IF NEW.product_price < 0 THEN
        SIGNAL SQLSTATE '45000' SET MESSAGE_TEXT = 'Product price
cannot be negative';
    END IF;
END;
//
DELIMITER ;
```

Output

The screenshot shows the MySQL Workbench interface with a query editor window titled "SQL File 3". The code in the editor is the trigger definition provided above. A specific line of code, `INSERT INTO Products (product_name, product_description, product_quantity, product_price, supplier_id, category_id, category_type) VALUES ('ABC', 'ABC', 100, -0.1, 1, 1, 1);` is highlighted in blue, indicating it is being executed. Below the editor, the "Output" pane displays the results of the execution. The log shows several rows of activity, including the creation of the trigger, the insertion attempt, and an error message indicating that the product price cannot be negative. The status bar at the bottom right shows the date and time as 07-04-2024 06:33 PM.

#	Time	Action	Message	Duration / Fetch
80	18:21:53	INSERT INTO Transactions (payment_type, unit_price, quantity, customer_id, store_id) VALUES ('credit_card', 1299.99, 100, 1, 1)	100 row(s) affected Records: 100 Duplicates: 0 Warnings: 0	0.016 sec
81	18:22:11	select * from Transactions LIMIT 0, 5000	100 row(s) returned	0.000 sec / 0.000 sec
82	18:24:19	select * from Transactions LIMIT 0, 5000	100 row(s) returned	0.000 sec / 0.000 sec
83	18:29:19	CREATE TRIGGER AfterSaleInsert AFTER INSERT ON Sales FOR EACH ROW BEGIN UPDATE Products SET ... WHERE ... END	1 row(s) affected	0.015 sec
84	18:29:42	INSERT INTO Sales (unit_price, quantity, product_id, store_id, transaction_id) VALUES (1299.99, 2, 1, 1, 1)	(499.99, 100 row(s) affected Records: 100 Duplicates: 0 Warnings: 0	0.031 sec
85	18:29:57	select * from Products LIMIT 0, 5000	500 row(s) returned	0.000 sec / 0.000 sec
86	18:32:34	CREATE TRIGGER EnforceMinimumProductPrice BEFORE INSERT ON Products FOR EACH ROW BEGIN IF NEW.product_price < 0 THEN SIGNAL SQLSTATE '45000' SET MESSAGE_TEXT = 'Product price cannot be negative'; END IF; END	Error Code: 1644: Product price cannot be negative	0.000 sec

→ UniqueProductName:-

The trigger fires before a new row is inserted into the Store_Products table.

```
DELIMITER //
CREATE TRIGGER EnforceUniqueProductNamePerStore BEFORE INSERT
ON Store_Products
FOR EACH ROW
BEGIN
    DECLARE count_products INT;

    -- Count the number of products with the same name within the same store
    SELECT COUNT(*) INTO count_products
    FROM Store_Products sp
    WHERE sp.product_id = NEW.product_id
    AND sp.store_id = NEW.store_id;

    -- If a product with the same name already exists in the same store, raise an
    error
    IF count_products > 0 THEN
        SIGNAL SQLSTATE '45000' SET MESSAGE_TEXT = 'Product name must
be unique within the store';
    END IF;
END;
//  
DELIMITER ;
```

Output

The screenshot shows the MySQL Workbench interface with the following details:

- File**: Local instance MySQL80
- Navigator**: Schemas (sakila, sys, world)
- Toolbars**: Standard MySQL Workbench toolbar.
- Connections**: College_3, College, bank, store, Walmart-India*
- Current Connection**: Walmart-India*
- Procedure Editor**: A code editor containing a stored procedure named `Don't Limit`. The code checks if a product name already exists in the same store and prevents insertion if it does, returning an error message.
- Output Window**: Shows the execution history and results of the procedure:

Time	Action	Message	Duration / Fetch
88	18:28:01 select* from Products	500 row(s) returned	0.000 sec / 0.000 sec
89	18:28:46 select* from sales	100 row(s) returned	0.000 sec / 0.000 sec
90	18:31:10 INSERT INTO Products (product_name, product_description, product_quantity, product_price, supplier_id, category_id)	Error Code: 1452 Cannot add or update a child row: a foreign key constraint fails ('walmart_india`.`products', CONSTRAINT `FK_products_sales` FOREIGN KEY (`supplier_id`) REFERENCES `sales` (`id`))	0.011 sec
91	18:32:14 CREATE TRIGGER EnforceMinimumProductPrice BEFORE INSERT ON Products FOR EACH ROW BEGIN	Error Code: 1452 Cannot add or update a child row: a foreign key constraint fails ('walmart_india`.`products', CONSTRAINT `FK_products_sales` FOREIGN KEY (`supplier_id`) REFERENCES `sales` (`id`))	0.031 sec
92	18:32:17 INSERT INTO Products (product_name, product_description, product_quantity, product_price, supplier_id, category_id)	Error Code: 1644 Product price cannot be negative	0.000 sec
93	18:34:42 CREATE TRIGGER EnforceUniqueProductNamePerStore BEFORE INSERT ON Store_Products FOR EACH ROW BEGIN	Error Code: 1644 Product name must be unique within the store	0.015 sec
94	18:34:49 INSERT INTO Store_Products (store_id, product_id) VALUES (1, 100)	Error Code: 1644 Product name must be unique within the store	0.000 sec
- Object Info**: Shows the selected object as `Don't Limit`.
- Session**: Shows the current session status.
- System Icons**: Includes a 37°C icon, battery level, network signal, and system tray icons.

→ Ensure Review Rating is between 1 and 5:-

The trigger fires before a new row is inserted into the Reviews table.

```
DELIMITER //
CREATE TRIGGER validate_review_rating BEFORE INSERT ON Reviews
FOR EACH ROW
BEGIN
    IF NEW.rating < 1 OR NEW.rating > 5 THEN
        SIGNAL SQLSTATE '45000' SET MESSAGE_TEXT = 'Review rating
must be between 1 and 5';
    END IF;
END //
DELIMITER ;
```

Output

The screenshot shows the MySQL Workbench interface with the following details:

- File Bar:** File, Edit, View, Query, Database, Server, Tools, Scripting, Help.
- Schemas:** Local instance MySQL80, sakila, sys, world.
- Query Editor:** Shows the trigger creation script and an attempt to insert a review with a rating of 10.

```
409 • INSERT INTO Store_Products (store_id, product_id) VALUES (1, 100);
410
411 -- Ensure Review Rating is between 1 and 5
412 DELIMITER //
413 • CREATE TRIGGER validate_review_rating BEFORE INSERT ON Reviews
414 FOR EACH ROW
415 BEGIN
416     IF NEW.rating < 1 OR NEW.rating > 5 THEN
417         SIGNAL SQLSTATE '45000' SET MESSAGE_TEXT = 'Review rating must be between 1 and 5';
418     END IF;
419 END //
420 DELIMITER ;
421
422 • INSERT INTO Reviews (product_id, review_text, rating, reviewer_name)
423     VALUES (1, 'Example review', 10, 'John Doe');
424
425
426 -- Inserting data into Sunnire
```

- Output Window:** Displays the execution log with the following entries:

Action	Time	Message	Duration / Fetch
90	18:31:10	INSERT INTO Products (product_name, product_description, product_quantity, product_price, supplier_id, cat... Error Code: 1452. Cannot add or update a child row: a foreign key constraint fails ('walmart_india'.products', C...	0.016 sec
91	18:32:14	CREATE TRIGGER EnforceMinimumProductPrice BEFORE INSERT ON Products FOR EACH ROW BEGIN ... 0 row(s) affected	0.031 sec
92	18:32:17	INSERT INTO Products (product_name, product_description, product_quantity, product_price, supplier_id, cat... Error Code: 1644. Product price cannot be negative	0.000 sec
93	18:34:42	CREATE TRIGGER EnforceUniqueProductNamePerStore BEFORE INSERT ON Store_Products FOR EACH ... 0 row(s) affected	0.015 sec
94	18:34:49	INSERT INTO Products (product_name, product_description, product_quantity, product_price, supplier_id, cat... Error Code: 1644. Product name must be unique within the store	0.000 sec
95	18:35:39	CREATE TRIGGER validate_review_rating BEFORE INSERT ON Reviews FOR EACH ROW BEGIN IF NE... 0 row(s) affected	0.031 sec
96	18:35:43	INSERT INTO Reviews (product_id, review_text, rating, reviewer_name) VALUES (1, 'Example review', 10, 'John Doe') Err Code: 1644. Review rating must be between 1 and 5.	0.000 sec