Online Retail Store Database Management System

Team members:
- Dev Thakkar (2020052)
- Hiren Arvind K. (2020066)
- Sahil Deshpande (2020114)
- Vishwesh Vhavle (2020156)

Project Description: Our project consists of an Online Retail Store Database Management System. It is a fully fledged model covering most necessities of businesses, ranging from information about customers, products, suppliers, orders and different stores. We have initialized various queries to check out products and stores for items, and also view the office staff while keeping in mind their privacy. Our data model also handles online deliveries making customer addresses and orders a part of our schema.

We have a UI made using HTML, CSS & PHP. Customers, suppliers, stores and admins can login using their IDs and passwords. Based on their privileges they can view data. Admins are given full authority over the data, and they can use the text field to input unique queries to manage the database.

We also have used indexing to quickly access important data from the system. Along with some sql queries we also have four embedded queries given as buttons.

Stakeholders:
- Consumers
- Retail store companies
- Suppliers
- Employees of the company

Relational Schemas:

Customer(**CustomerID: integer**, Fname: string, Lname: string, PhoneNum: string, Email: string, Pword: string)

Product(**ProductID: integer**, ProductName: string, Type: string, PricePU: real)

Store(**StoreID: integer**, Email: string, PhoneNum: string, Pword: string)

Employee(**EmployeeID: integer**, Fname: string, Lname: string, PhoneNo: string, Salary: real, ExperienceYears: integer)

Supplier(**SupplierID: integer**, SupplierName: string, PhoneNum: string, City: string, Email: string, Pword: string)

admins(**AdminID: integer**, Pword: string)

DeliveryAddress(AddressID: integer, CustomerID: integer, Pincode: string, City: string, AddressLine1: string, AddressLine2: string)

StoreAddress(AddressID: integer, StoreID: integer, Pincode: string, City: string, AddressLine1: string, AddressLine2: string)

CustomerOrder**(OrderID: integer**, CustomerID: integer, StoreID: integer, ProductID: integer, PaymentID: integer, Date: date)

Sells(**StoreID: integer, ProductID: integer**, Quantity: integer)

Supplies(**SupplierID: integer, ProductID: integer**, CostPU: real)

SupplyQuotation(**SupplyQuotationID: integer**, ProductID: integer, StoreID: integer, SupplierID: integer, QuotationDate: date, Quantity: integer)

Works(**StoreID: integer, EmployeeID: integer**)

Manages(**StoreID: integer**, EmployeeID: integer, DateOfAppointment: date)

Delivery(**OrderID: integer**, AddressID: integer, DeliveryDate: date)

SupplyDelivery(**SupplyQuotationID: integer**, StoreAddressID: integer, ReceivingDate: date)

PaymentMethod(**PaymentID: integer, CustomerID: integer**, paymentType: string)

GenericLogin(**loginID: integer, LoginType: string, Logtime: timestamp**)
CustomerLogin(**customerID: integer, Logtime: timestamp**)

SupplierLogin(**SupplierID: integer, Logtime: timestamp**)

StoreLogin(**employeeID: integer, Logtime: timestamp**)

Queries:

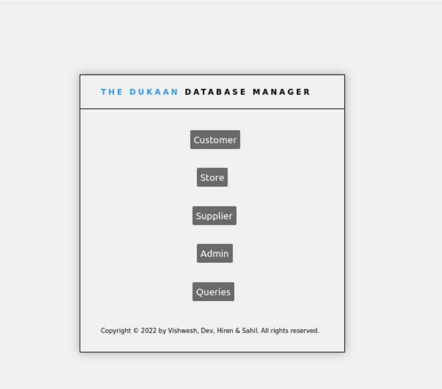
```
-- Query 1 - Show the orders, products and delivery dates of orders delivered to a particular customer
SELECT CustomerOrder.orderID, CustomerOrder.productID
FROM CustomerOrder
JOIN delivery
ON CustomerOrder.orderID = delivery.orderID
WHERE CustomerOrder.customerID = 40;


-- Query 2 -Show products and stores where the product has a price of more than 500 rupees and quantity is less than 30
SELECT product.productID, product.productName, store.storeID
FROM product
JOIN sells
ON product.productID = sells.productID
JOIN store
ON store.storeID = sells.storeID
WHERE product.PricePU > 500 AND sells.Quantity < 30;


-- Query 3 -Show how many products are electronics, i.e. 'Laptops' or 'Smart Phones'
SELECT SUM(electronicsTotal)
FROM (
    SELECT COUNT(product.productID) as electronicsTotal
    FROM product
    WHERE product.productType = 'Laptops' OR product.productType = 'Smart Phones'
)eProducts;


-- Query 4 -Create a view to show all employee details except salary and contact info
CREATE VIEW employeeList
AS SELECT EmployeeID, Fname as FirstName, Lname as LastName, Experience FROM employee;

SELECT * FROM employeeList;


-- Query 5 -Find employeeIDs', names' of managers of stores with employees with Last name as 'Scott'
```

```sql
SELECT employeeID, Fname, Lname
FROM employee
WHERE employeeID in (
    SELECT employeeID
    FROM manages
    WHERE storeID in (
        SELECT storeID
        FROM works
        WHERE employeeID in (
            SELECT employeeID
            FROM employee
            WHERE Lname = 'Scott'
        )
    )
);


-- Query 6 -Using the employeeList view show the First names of employees and how many employees share the same first name
-- and order them lexographically in reverse
SELECT FirstName, COUNT(FirstName)
FROM employeeList
GROUP BY FirstName
ORDER BY FirstName DESC;


-- Query 7 - Find median cost price per unit of products for each supplier
SELECT supplier.supplierID, AVG(costPU) as MedianCostPricePerUnit
FROM supplier
JOIN supplies
ON supplier.supplierID = supplies.supplierID
GROUP BY supplierID;


-- Query 8 - Show delivery date, store address, product type and quantity
SELECT City, ReceivingDate, Quantity, ProductType
FROM StoreAddress, SupplyDelivery, supplyQuotation, product
WHERE (StoreAddress.StoreID = supplyQuotation.StoreID)
AND (StoreAddress.AddressID = SupplyDelivery.AddressID)
AND (supplyQuotation.ProductID = product.ProductID)
ORDER BY ReceivingDate;


-- Query 9 - Show customer first name, last name, product, orderID and payment method
Select Fname, Lname, ProductName, OrderID, paymentType
FROM customer, product, CustomerOrder, paymentMethod
WHERE (customer.CustomerID = CustomerOrder.CustomerID)
```

```sql
AND (product.ProductID = CustomerOrder.ProductID)
AND (customer.CustomerID = paymentMethod.CustomerID)
ORDER BY Fname;


-- Query 10 - Show customer email address and phone number where delivery city is new
delhi
SELECT Email, PhoneNum
FROM customer, deliveryAddress
WHERE (customer.customerID = deliveryAddress.customerID)
AND (City = 'New Delhi')
ORDER BY Email;

-- Query 11 - Show the sellers that sell any Laptop under 50000
SELECT StoreID
FROM sells
WHERE EXISTS (
    SELECT ProductID
    FROM product
    WHERE product.productID = sells.productID AND ProductType = 'Laptops' AND PricePU
< 50000
);


-- Query 12 - Change a value in a row which is null
INSERT INTO customer VALUES (999,'Sumit','Kaif',
NULL,'sumitk2@gmail.com','8LHIn5ebX');
SELECT *
FROM customer
WHERE CustomerID = '999';
UPDATE customer
SET PhoneNum = '1000000000'
WHERE PhoneNum IS NULL;
SELECT *
FROM customer
WHERE CustomerID = '999';
DELETE FROM customer
WHERE CustomerID = '999';

-- Query 13 - Show the names of customers who have i as the second letter in their
First names
SELECT Fname
FROM customer
WHERE Fname LIKE '_i%';


--Query 14 - Check which items need restocking and which don't and make note
```

```
SELECT StoreID, ProductID,
CASE
    WHEN Quantity > 30 THEN 'No restock needed'
    WHEN Quantity < 10 THEN 'Restock ASAP'
    ELSE 'Restock needed'
END AS QuantityText
FROM sells;

-- Query 15 - Show the top 3 most expensive products
SELECT productID, pricePU
FROM product
ORDER BY pricePU DESC
LIMIT 3;

-- Query 16 - Show the employees and their managers
SELECT employee.employeeID, manages.employeeID AS managerID
FROM employee
LEFT JOIN works ON employee.employeeID = works.employeeID
LEFT JOIN manages ON works.storeID = manages.storeID
ORDER BY managerID;

-- Query 17 - count which type of product is maximum
Select MAX(productType), COUNT(productType)
FROM product
group by productType
ORDER BY count(productType) DESC
LIMIT 1;

-- Query 18 - calculate the standard deviation of price of all products from mean
SELECT STD(pricePU), ProductType
FROM product
GROUP BY ProductType;
```

Project Interface:


THE DUKAAN DATABASE MANAGER

Copyright © 2022 by Vishwesh, Dev, Hiren & Sahil. All rights reserved.

The interface is designed using HTML5, CSS and PHP. The project is hosted using 000webhost and can be found at https://dukaan.me/ . It uses PHPMyAdmin SQL server to store the database.

After collecting inputs from user from HTML pages, the input is posted to PHP script which connects with PHPMyAdmin server and accesses data using embedded SQL queries and echoes whatever data is returned.

There are 4 access controlled interfaces for each type of user i.e Customer, Store, Supplier and Admin. Customer, Store & Supplier interfaces can only view tables. Admin has the privilege to show, add, update and delete certains tables and interact with the entire database from the query box.