

MIS686: Term Project

Enterprise Database Management

San Diego State University

Parth Desai, Roshini Padmanabha, Sharad Parmar

STEP 1: Topic/Domain Selection

Topic - Retail Store Database Management

Introduction

In the ever-evolving landscape of retail, modern retailers seek to thrive among their competitors. Hence, efficient management of large-scale data has become paramount for success in the retail domain. By efficiently managing and leveraging data, retail businesses can position themselves for success, adapt to market dynamics, and provide exceptional experiences to their customers. The Retail Stores Database Management system is designed to streamline and organize various aspects of retail operations. This comprehensive database encompasses entities such as customers, products, transactions, employees, and store locations, providing a centralized platform for data storage and retrieval. With the ability to track inventory and stockouts of products, customer interactions, and sales transactions, this database empowers retail businesses to make informed decisions, enhance customer experiences, and optimize overall operational efficiency of the business.

Advantages of Retail DBMS

Inventory Control:

- The database facilitates real-time tracking of product stock levels, preventing overstock or stockouts. This, in turn, minimizes losses due to excess inventory or missed sales opportunities.

Customer Relationship Management (CRM):

- Customer profiles and transaction history enable personalized services, targeted promotions, and improved customer satisfaction. Loyalty points and membership data contribute to effective CRM strategies.

Sales Analytics:

- Analyzing sales data allows retailers to identify top-performing products, track sales trends, and anticipate consumer preferences. This data-driven approach aids in optimizing pricing and marketing strategies.

Employee Management:

- Employee information, including positions, hire dates, and performance metrics, facilitates efficient workforce management. This ensures appropriate staffing levels, proper scheduling, and improved employee performance.

Operational Efficiency:

- Centralized data management streamlines various retail processes, reducing redundancies and errors. This contributes to a more efficient and organized retail operation.

Uses Cases

Multi-Branch Retail Chains:

- Large retail chains with multiple branches can utilize the database to maintain consistency in inventory, pricing, and customer experiences across various locations.

Specialty Stores:

- Specialty stores focusing on specific product categories benefit from tailored data management, allowing them to cater to niche markets and optimize product offerings.

Franchise Operations:

- Franchise businesses can implement the database to ensure standardized inventory control, pricing strategies, and customer interactions across different franchise locations.

E-commerce Platforms:

- Online retailers can integrate the database to manage product listings, customer orders, and fulfillment processes. This allows for seamless coordination between online and physical store inventories.

Local Shops and Boutiques:

- Even small local shops can adopt a simplified version of the database to manage customer transactions, monitor inventory levels, and enhance customer engagement.

Data-Driven Decision Making:

- Retailers can leverage the database to analyze data trends, identify market opportunities, and make informed decisions that contribute to business growth and sustainability.

The Retail Stores Database Management system is a versatile tool that adapts to diverse retail models, providing a solid foundation for data-driven strategies and operational excellence in the retail industry.

STEP 2: Conceptual Data Modeling and Database Design

Business rules

Employee Accountability:

- Employees are accountable for the transactions they handle.
- An employee cannot be associated with more than one transaction at the same time.
- An employee must be assigned to a store before being involved in any transactions.

Invoice Consistency:

- An invoice must have at least one product line.
- The total quantity and price in the invoice must be the sum of the product lines.

Product Line Validation:

- Product lines must have a positive quantity.
- The line price in the product line should be the product of quantity and the unit price of the associated product.

Stock Availability:

- The quantity of a product in stock cannot be negative.
- Product lines in an invoice cannot exceed the available stock for each product.

Supplier Integrity:

- Each product must be associated with a valid supplier.
- A supplier must exist in the supplier table before being associated with any product.

Product Price Consistency:

- The price of a product in the product table must be consistent with the prices in the product lines of invoices.

Employee Contact Information:

- Employee phone numbers must be unique.

Customer Address Completeness:

- Customer addresses must be complete, including street, city, state, ZIP code.
- No customer should have incomplete address details.

Store Information Completeness:

- Store names and addresses must be complete and unique.
- No store should have incomplete or duplicate information.

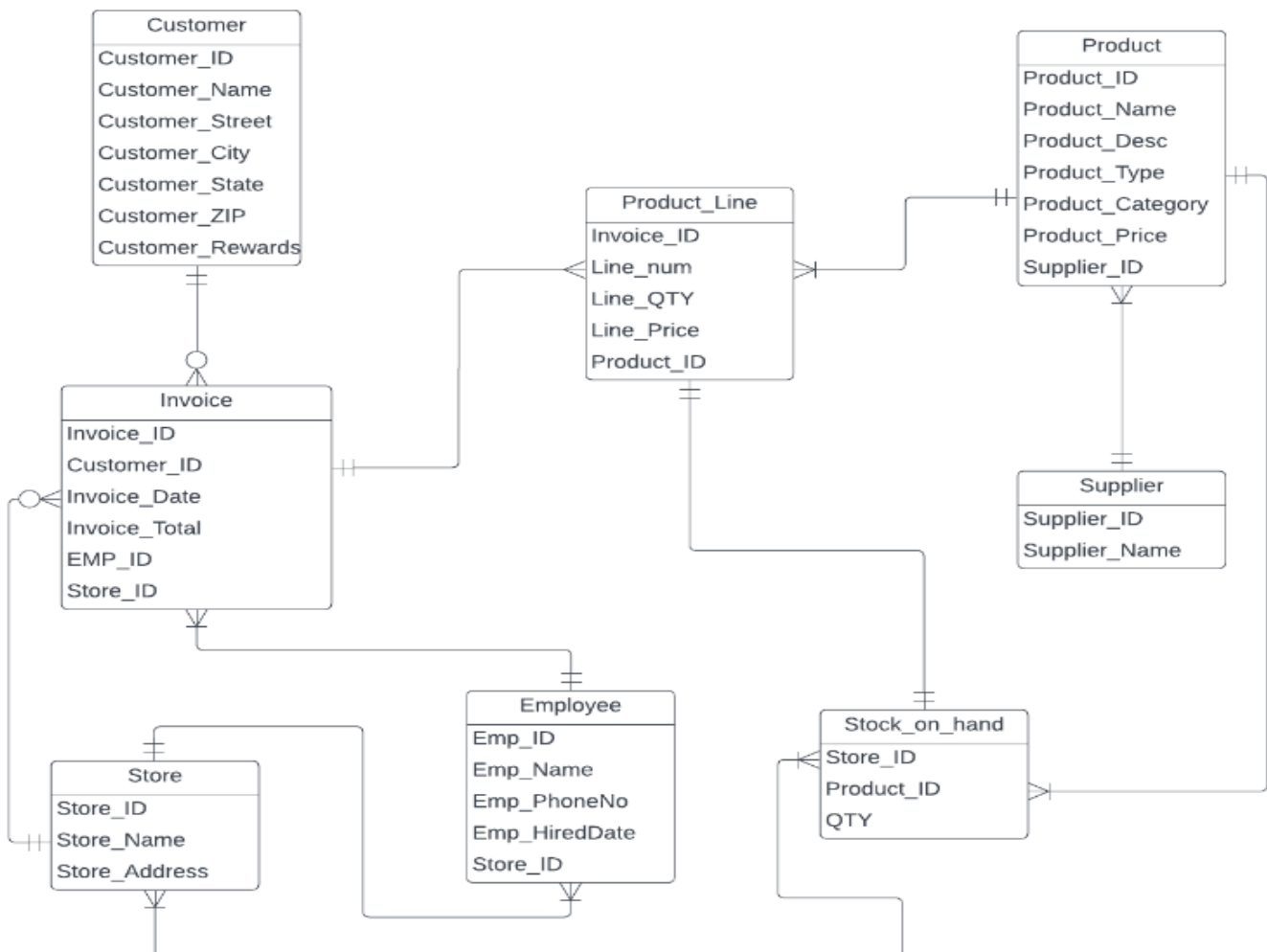
Invoice Date Validity:

- The invoice date must be on or after the transaction date of the employee associated with the invoice.
- The invoice date should not be in the future.

Invoice Total Calculation:

- The invoice total should be the sum of all product line prices in the invoice.

ER/EER DIAGRAM:



Relational Model :



STEP 3: Database Implementation

SQL Commands to Create Tables for Database:

Below are the SQL commands that were used to create tables for the database

USE EDM_Project;

CREATE TABLE Customer (

Customer_ID INT AUTO_INCREMENT PRIMARY KEY,

Customer_Name VARCHAR(255) NOT NULL,

Customer_Street VARCHAR(255) NOT NULL,

Customer_City VARCHAR(255) NOT NULL,

Customer_State VARCHAR(255) NOT NULL,

Customer_ZIP VARCHAR(10) NOT NULL,

Customer_Rewards INT DEFAULT 0

);

```
CREATE TABLE Store (  
    Store_ID INT AUTO_INCREMENT PRIMARY KEY,  
    Store_Name VARCHAR(255) NOT NULL,  
    Store_Address VARCHAR(255) NOT NULL  
);
```

```
CREATE TABLE Employee (  
    Emp_ID INT AUTO_INCREMENT PRIMARY KEY,  
    Emp_Name VARCHAR(255) NOT NULL,  
    Emp_PhoneNo VARCHAR(20),  
    Emp_HiredDate DATE,  
    Store_ID INT,  
    CONSTRAINT Employee_FK FOREIGN KEY (Store_ID) REFERENCES  
Store(Store_ID)  
);
```

```
CREATE TABLE Invoice (  
    Invoice_ID INT AUTO_INCREMENT PRIMARY KEY,  
    Customer_ID INT,  
    Invoice_Date DATE,  
    Invoice_Total DECIMAL(10, 2),  
    Emp_ID INT,  
    Store_ID INT,  
    CONSTRAINT INVOICE_FK1 FOREIGN KEY (Customer_ID) REFERENCES  
Customer(Customer_ID),  
    CONSTRAINT INVOICE_FK2 FOREIGN KEY (Emp_ID) REFERENCES  
Employee (Emp_ID),  
    CONSTRAINT INVOICE_FK3 FOREIGN KEY (Store_ID) REFERENCES  
Store(Store_ID)  
);
```

```
CREATE TABLE Supplier (  
    Supplier_ID INT AUTO_INCREMENT PRIMARY KEY,  
    Supplier_Name VARCHAR(255) NOT NULL  
);
```

```
CREATE TABLE Product (  
    Product_ID INT AUTO_INCREMENT PRIMARY KEY,  
    Product_Name VARCHAR(255) NOT NULL,  
    Product_Desc TEXT,  
    Product_Type VARCHAR(50),  
    Product_Category VARCHAR(50),  
    Product_Price DECIMAL(10, 2),  
    Supplier_ID INT,  
    CONSTRAINT PRODUCT_FK FOREIGN KEY (Supplier_ID) REFERENCES  
Supplier(Supplier_ID)  
);
```

```
CREATE TABLE Product_Line (  
    Invoice_ID INT,  
    Line_num INT,  
    Line_QTY INT,  
    Line_Price DECIMAL(10, 2),  
    Product_ID INT,  
    CONSTRAINT Product_Line_FK1 FOREIGN KEY (Invoice_ID) REFERENCES  
Invoice (Invoice_ID),  
    CONSTRAINT Product_Line_FK2 FOREIGN KEY (Product_ID) REFERENCES  
Product(Product_ID )  
);
```

```
CREATE TABLE Stock_on_hand (  
    Store_ID INT,  
    Product_ID INT,  
    QTY INT,  
    CONSTRAINT Stock_on_hand_FK1 FOREIGN KEY (Store_ID ) REFERENCES  
Store(Store_ID ),  
    CONSTRAINT Stock_on_hand_FK2 FOREIGN KEY (Product_ID ) REFERENCES  
Product(Product_ID )  
);
```

SQL Commands to Insert Data Into Database:

USE EDM_Project;

INSERT INTO Customer (Customer_ID, Customer_Name, Customer_Street, Customer_City, Customer_State, Customer_ZIP, Customer_Rewards) VALUES

(2000, 'Alice Johnson', '456 Oak Lane', 'Another City', 'NY', '67890', 100),
(2001, 'Bob Anderson', '789 Maple St', 'Someville', 'TX', '54321', 150),
(2002, 'Charlie Smith', '321 Elm Ave', 'New City', 'FL', '98765', 200),
(2003, 'David White', '654 Pine St', 'Smalltown', 'WA', '13579', 300),
(2004, 'Emma Davis', '987 Cedar St', 'Big City', 'IL', '24680', 500),
(2005, 'Frank Wilson', '111 Fir St', 'Littleville', 'OH', '97531', 120),
(2006, 'Grace Taylor', '222 Walnut Blvd', 'Hometown', 'GA', '86420', 80),
(2007, 'Henry Brown', '333 Spruce St', 'Metropolis', 'MI', '25846', 70),
(2008, 'Isabel Turner', '444 Birch St', 'Village', 'NC', '36987', 250),
(2009, 'Jack Miller', '555 Pineapple St', 'Townsville', 'AZ', '14703', 180),
(2010, 'Sophia Davis', '123 Oak Lane', 'Another City', 'NY', '54321', 50),
(2011, 'Oliver Smith', '456 Maple St', 'Someville', 'TX', '98765', 90),
(2012, 'Mia White', '789 Elm Ave', 'New City', 'FL', '13579', 120),
(2013, 'Liam Brown', '987 Pine St', 'Smalltown', 'WA', '24680', 200),
(2014, 'Emma Wilson', '111 Cedar St', 'Big City', 'IL', '97531', 300),
(2015, 'Noah Taylor', '222 Fir St', 'Littleville', 'OH', '86420', 80),
(2016, 'Ava Turner', '333 Walnut Blvd', 'Hometown', 'GA', '25846', 70),
(2017, 'Lucas Anderson', '444 Spruce St', 'Metropolis', 'MI', '36987', 150),
(2018, 'Isabella Johnson', '555 Birch St', 'Village', 'NC', '14703', 250),
(2019, 'Jackson Davis', '666 Pineapple St', 'Townsville', 'AZ', '96321', 180),
(2020, 'Sophie Wilson', '777 Peach St', 'Cityville', 'CA', '25873', 300),
(2021, 'Oliver Taylor', '888 Cedar St', 'Mountain View', 'CO', '78945', 120),
(2022, 'Mia Smith', '999 Oak Lane', 'Riverdale', 'GA', '12345', 50),
(2023, 'Ethan White', '111 Maple St', 'Sunnydale', 'FL', '56789', 90),
(2024, 'Ava Taylor', '222 Elm Ave', 'Hilltop', 'IL', '98765', 120),
(2025, 'Liam Johnson', '333 Pine St', 'Valley City', 'OH', '24680', 200),
(2026, 'Isabella Davis', '444 Cedar St', 'Sunset Hills', 'MO', '97531', 300),
(2027, 'Jackson Brown', '555 Fir St', 'Lakeside', 'MI', '86420', 80),
(2028, 'Sophia Taylor', '666 Walnut Blvd', 'Highland', 'IN', '25846', 70),
(2029, 'Lucas Turner', '777 Spruce St', 'Meadowview', 'VA', '36987', 150),
(2030, 'Olivia Anderson', '888 Birch St', 'Lakeshore', 'WA', '14703', 250),
(2031, 'Liam Turner', '999 Pineapple St', 'Riverside', 'CA', '96321', 180),
(2032, 'Ava Smith', '123 Peach St', 'Hilltop', 'CO', '25873', 300),
(2033, 'Oliver Wilson', '456 Oak Lane', 'Sunnydale', 'FL', '12345', 50),
(2034, 'Mia Brown', '789 Maple St', 'Valley City', 'OH', '56789', 90),
(2035, 'Jackson White', '321 Elm Ave', 'Sunset Hills', 'MO', '98765', 120),
(2036, 'Sophie Davis', '654 Pine St', 'Lakeside', 'MI', '24680', 200),
(2037, 'Oliver Smith', '987 Cedar St', 'Highland', 'IN', '97531', 300),
(2038, 'Isabella Taylor', '111 Fir St', 'Meadowview', 'VA', '86420', 80),
(2039, 'Lucas Anderson', '222 Walnut Blvd', 'Lakeshore', 'WA', '25846', 70),
(2040, 'Olivia Turner', '333 Spruce St', 'Riverside', 'CA', '36987', 150),
(2041, 'Ethan Johnson', '444 Birch St', 'Hilltop', 'CO', '14703', 250),
(2042, 'Ava Davis', '555 Pineapple St', 'Sunnydale', 'FL', '96321', 180),
(2043, 'Liam Wilson', '666 Peach St', 'Valley City', 'OH', '25873', 300),
(2044, 'Isabella White', '777 Oak Lane', 'Sunset Hills', 'MO', '12345', 50),
(2045, 'Jackson Taylor', '888 Maple St', 'Lakeside', 'MI', '56789', 90),
(2046, 'Sophia Brown', '999 Elm Ave', 'Highland', 'IN', '98765', 120),

(2047, 'Oliver Taylor', '123 Pine St', 'Meadowview', 'VA', '24680', 200),
(2048, 'Ava Anderson', '456 Cedar St', 'Riverside', 'CA', '97531', 300),
(2049, 'Lucas Turner', '789 Fir St', 'Hilltop', 'CO', '86420', 80);

INSERT INTO Store (Store_ID, Store_Name, Store_Address) VALUES
(200, 'Main Street Electronics', '123 Main St'),
(201, 'City Center Appliances', '456 Center Ave'),
(202, 'Southern Hardware', '789 Southern Blvd'),
(203, 'Tech Hub Superstore', '321 Tech Lane'),
(204, 'Northern Tools and More', '654 Northern St'),
(205, 'Green Living Supplies', '987 Green Ave'),
(206, 'Downtown Electronics Emporium', '111 Downtown St'),
(207, 'Hometown Hardware', '222 Hometown Blvd'),
(208, 'Metro Electronics', '333 Metro St'),
(209, 'Village Home Goods', '444 Village Ave');

INSERT INTO Employee (Emp_ID, Emp_Name, Emp_PhoneNo, Emp_HiredDate, Store_ID) VALUES
(1, 'John Smith', '123-456-7890', '2023-01-01', 200),
(2, 'Jane Doe', '234-567-8901', '2023-02-01', 201),
(3, 'Bob Johnson', '345-678-9012', '2023-03-01', 202),
(4, 'Alice Williams', '456-789-0123', '2023-04-01', 203),
(5, 'Charlie Brown', '567-890-1234', '2023-05-01', 204),
(6, 'Eva Davis', '678-901-2345', '2023-06-01', 205),
(7, 'David Miller', '789-012-3456', '2023-07-01', 206),
(8, 'Grace Taylor', '890-123-4567', '2023-08-01', 207),
(9, 'Frank Wilson', '901-234-5678', '2023-09-01', 208),
(10, 'Sophie Turner', '012-345-6789', '2023-10-01', 209);

INSERT INTO Invoice (Invoice_ID, Customer_ID, Invoice_Date, Invoice_Total, Emp_ID, Store_ID)
VALUES
(100, 2000, '2023-01-01', 500.00, 1, 200),
(101, 2001, '2023-02-15', 750.25, 2, 201),
(102, 2002, '2023-03-20', 100.00, 3, 202),
(103, 2003, '2023-04-10', 2000.50, 4, 203),
(104, 2004, '2023-05-05', 350.75, 5, 204),
(105, 2005, '2023-06-12', 50.00, 6, 205),
(106, 2006, '2023-07-08', 1200.25, 7, 206),
(107, 2007, '2023-08-25', 600.00, 8, 207),
(108, 2008, '2023-09-30', 800.75, 9, 208),
(109, 2009, '2023-10-15', 950.00, 10, 209),
(110, 2010, '2023-11-01', 300.00, 1, 200),
(111, 2011, '2023-11-15', 450.25, 2, 201),
(112, 2012, '2023-12-01', 75.50, 3, 202),
(113, 2013, '2024-01-10', 1200.75, 4, 203),
(114, 2014, '2024-02-05', 250.50, 5, 204),
(115, 2015, '2024-03-12', 30.00, 6, 205),
(116, 2016, '2024-04-08', 900.25, 7, 206),
(117, 2017, '2024-05-25', 480.00, 8, 207),
(118, 2018, '2024-06-30', 620.75, 9, 208),
(119, 2019, '2024-07-15', 780.00, 10, 209),
(120, 2020, '2024-08-01', 150.00, 1, 200),

(121, 2021, '2024-08-15', 200.25, 2, 201),
 (122, 2022, '2024-09-01', 90.50, 3, 202),
 (123, 2023, '2024-10-10', 1500.75, 4, 203),
 (124, 2024, '2024-11-05', 300.50, 5, 204),
 (125, 2025, '2024-12-12', 45.00, 6, 205),
 (126, 2026, '2025-01-08', 800.25, 7, 206),
 (127, 2027, '2025-02-25', 350.00, 8, 207),
 (128, 2028, '2025-03-30', 500.75, 9, 208),
 (129, 2029, '2025-04-15', 680.00, 10, 209),
 (130, 2030, '2025-05-01', 120.00, 1, 200),
 (131, 2031, '2025-05-15', 175.25, 2, 201),
 (132, 2032, '2025-06-01', 60.50, 3, 202),
 (133, 2033, '2025-07-10', 900.75, 4, 203),
 (134, 2034, '2025-08-05', 200.50, 5, 204),
 (135, 2035, '2025-09-12', 35.00, 6, 205),
 (136, 2036, '2025-10-08', 700.25, 7, 206),
 (137, 2037, '2025-11-25', 400.00, 8, 207),
 (138, 2038, '2025-12-30', 550.75, 9, 208),
 (139, 2039, '2026-01-15', 720.00, 10, 209),
 (140, 2040, '2026-02-01', 100.00, 1, 200),
 (141, 2041, '2026-02-15', 150.25, 2, 201),
 (142, 2042, '2026-03-01', 80.50, 3, 202),
 (143, 2043, '2026-04-10', 1300.75, 4, 203),
 (144, 2044, '2026-05-05', 250.50, 5, 204),
 (145, 2045, '2026-06-12', 40.00, 6, 205),
 (146, 2046, '2026-07-08', 1000.25, 7, 206),
 (147, 2047, '2026-08-25', 550.00, 8, 207),
 (148, 2048, '2026-09-30', 700.75, 9, 208),
 (149, 2049, '2026-10-15', 880.00, 10, 209);

INSERT INTO Supplier (Supplier_ID, Supplier_Name) VALUES

(1000, 'ABC Electronics'),
 (1001, 'XYZ Manufacturing'),
 (1002, 'Smith & Co. Supplies'),
 (1003, 'Johnson Industrial Solutions'),
 (1004, 'Global Tech Parts'),
 (1005, 'Quality Suppliers Ltd.'),
 (1006, 'Green Energy Systems'),
 (1007, 'Innovate Technologies'),
 (1008, 'Star Components Inc.'),
 (1009, 'Pinnacle Manufacturing');

INSERT INTO Product (Product_Name, Product_Desc, Product_Type, Product_Category, Product_Price, Supplier_ID) VALUES

('Tablet', 'Android tablet with HD display', 'Electronics', 'Computers', 299.99, 1000),
 ('Bluetooth Speakers', 'Wireless Bluetooth speakers', 'Electronics', 'Audio', 49.99, 1001),
 ('Circular Saw', 'Powerful circular saw for woodworking', 'Tools', 'Power Tools', 129.99, 1000),
 ('Industrial Drill', 'Heavy-duty industrial drill', 'Tools', 'Power Tools', 179.99, 1003),
 ('Graphics Card', 'High-end graphics card for gaming', 'Electronics', 'Computer Components', 499.99, 1004),
 ('Smart Thermostat', 'Wi-Fi-enabled smart thermostat', 'Home and Living', 'Smart Home', 89.99, 1005),
 ('LED Desk Lamp', 'Adjustable LED desk lamp', 'Home and Living', 'Lighting', 19.99, 1005),

('Wireless Security Camera', 'Outdoor wireless security camera', 'Electronics', 'Security', 129.99, 1007),
('Arduino Starter Kit', 'Starter kit for Arduino projects', 'Electronics', 'DIY Electronics', 49.99, 1008),
('3D Printer', 'Desktop 3D printer for prototyping', 'Electronics', '3D Printers', 899.99, 1009);

```
INSERT INTO Product_Line (Invoice_ID, Line_num, Line_QTY, Line_Price, Product_ID) VALUES
(100, 1, 2, 599.98, 1),
(100, 2, 1, 49.99, 2),
(101, 1, 3, 224.97, 3),
(101, 2, 2, 359.98, 4),
(102, 1, 1, 499.99, 5),
(102, 2, 4, 359.96, 6),
(103, 1, 2, 99.98, 7),
(103, 2, 1, 49.99, 8),
(104, 1, 3, 1499.97, 9),
(104, 2, 2, 1799.98, 10),
(110, 3, 2, 599.98, 3),
(110, 4, 1, 49.99, 4),
(111, 3, 3, 224.97, 5),
(111, 4, 2, 359.98, 6),
(112, 3, 1, 499.99, 7),
(112, 4, 4, 359.96, 8),
(113, 3, 2, 99.98, 9),
(113, 4, 1, 49.99, 10),
(114, 3, 3, 1499.97, 1),
(114, 4, 2, 1799.98, 2),
(115, 3, 1, 99.98, 3),
(116, 3, 3, 149.97, 3),
(116, 4, 1, 99.99, 4),
(117, 3, 2, 799.98, 5),
(117, 4, 1, 399.99, 6),
(118, 3, 3, 449.97, 7),
(118, 4, 2, 259.98, 8),
(119, 3, 1, 99.99, 9),
(119, 4, 4, 239.96, 10),
(120, 3, 2, 899.98, 1),
(120, 4, 1, 179.99, 2),
(121, 3, 3, 299.97, 3),
(121, 4, 2, 359.98, 4),
(122, 3, 1, 499.99, 5),
(122, 4, 4, 719.96, 6),
(123, 3, 2, 199.98, 7),
(123, 4, 1, 99.99, 8),
(124, 3, 3, 899.97, 9),
(124, 4, 2, 699.98, 10),
(125, 3, 1, 499.99, 1),
(125, 4, 4, 719.96, 2),
(126, 3, 2, 599.98, 3),
(126, 4, 1, 49.99, 4),
(127, 3, 3, 224.97, 5),
(127, 4, 2, 359.98, 6),
(128, 3, 1, 499.99, 7),
```

(128, 4, 4, 359.96, 8),
(129, 3, 2, 99.98, 9),
(129, 4, 1, 49.99, 10),
(130, 3, 3, 1499.97, 1),
(130, 4, 2, 1799.98, 2),
(131, 3, 1, 99.98, 3),
(131, 4, 3, 224.97, 4),
(132, 3, 2, 359.98, 5),
(132, 4, 1, 499.99, 6),
(133, 3, 4, 359.96, 7),
(133, 4, 2, 99.98, 8),
(134, 3, 1, 49.99, 9),
(134, 4, 4, 1499.97, 10),
(135, 3, 2, 1799.98, 1),
(135, 4, 1, 99.98, 2),
(136, 3, 3, 149.97, 3),
(136, 4, 1, 99.99, 4),
(137, 3, 2, 799.98, 5),
(137, 4, 1, 399.99, 6),
(138, 3, 3, 449.97, 7),
(138, 4, 2, 259.98, 8),
(139, 3, 1, 99.99, 9),
(139, 4, 4, 239.96, 10),
(140, 3, 2, 899.98, 1),
(140, 4, 1, 179.99, 2),
(141, 3, 3, 299.97, 3),
(141, 4, 2, 359.98, 4),
(142, 3, 1, 499.99, 5),
(142, 4, 4, 719.96, 6),
(143, 3, 2, 199.98, 7),
(143, 4, 1, 99.99, 8),
(144, 3, 3, 899.97, 9),
(144, 4, 2, 699.98, 10),
(145, 3, 1, 499.99, 1),
(145, 4, 4, 719.96, 2),
(146, 3, 2, 599.98, 3),
(146, 4, 1, 49.99, 4),
(147, 3, 3, 224.97, 5),
(147, 4, 2, 359.98, 6),
(148, 3, 1, 499.99, 7),
(148, 4, 4, 359.96, 8),
(149, 3, 2, 99.98, 9),
(149, 4, 1, 49.99, 10);

INSERT INTO Stock_on_hand (Store_ID, Product_ID, QTY) VALUES

(200, 1, 50),
(201, 2, 75),
(202, 3, 100),
(203, 4, 200),
(204, 5, 35),
(205, 6, 120),
(206, 7, 60),
(207, 8, 50),

```
(208, 9, 75),  
(209, 10, 95);
```

Triggers

1) Trigger - To update the stock on hand quantity of products in the store based on Invoice created.

```
-- Create an 'after insert' trigger on the 'Invoice' table
```

```
DELIMITER //
```

```
CREATE TRIGGER after_insert_invoice_stock
```

```
AFTER INSERT ON Invoice
```

```
FOR EACH ROW
```

```
BEGIN
```

```
-- Log the update action to stock_on_hand table
```

```
UPDATE stock_on_hand AS S
```

```
SET S.QTY = S.QTY - (
```

```
SELECT PL.Line_QTY
```

```
FROM Product_Line AS PL, Invoice I
```

```
WHERE I.Invoice_ID = PL.Invoice_ID AND S.Product_ID = PL.Product_ID AND  
S.Store_ID = I.Store_ID
```

```
);
```

```
END;
```

```
//
```

```
DELIMITER ;
```

2) Trigger - To update Customer Reward Points based on Total Invoice value of Customer.

-- Create an 'after insert' trigger on the 'Invoice' table

DELIMITER //

CREATE TRIGGER after_insert_invoice_rewards

AFTER INSERT ON Invoice

FOR EACH ROW

BEGIN

-- Log the update action to customer table

UPDATE Customer AS C

SET C.Customer_Rewards = C.Customer_Rewards + (NEW.Invoice_Total / 10)

WHERE C.Customer_ID = NEW.Customer_ID;

END;

//DELIMITER ;

Deploying The Database on AWS Cloud

Database Name: EDM_Project

The MySQL Database was deployed on AWS Cloud.

EndPoint : database-1.cdbv9bf7nxth.us-east-2.rds.amazonaws.com

Analytical Questions and SQL Queries

1. List the top 10 customers based on Maximum bill value in Stores IDs: 202,208,205?

```
SELECT
  C.customer_name,
  C.Customer_ID,
  C.Customer_city,C.Customer_State
  C.Customer_Rewards,
  I.Store_ID, S.Store_Name
  I.Invoice_ID,
  count(*) as purchases
FROM Customer C, Invoice I, Store S
WHERE C.Customer_ID=I.Customer_ID AND I.Store_ID=S.Store_ID AND I.Store_ID
in(202,208,205)
GROUP BY 1 having purchases>0
ORDER BY purchases desc
LIMIT 10
```

	A customer_name	Customer_ID	Customer_Rewards	Store_ID	Invoice_ID	purchases
1	Charlie Smith	2002	200	202	102	1
2	Mia White	2012	120	202	112	1
3	Mia Smith	2022	50	202	122	1
4	Ava Smith	2032	300	202	132	1
5	Ava Davis	2042	180	202	142	1
6	Frank Wilson	2005	120	205	105	1
7	Noah Taylor	2015	80	205	115	1
8	Liam Johnson	2025	200	205	125	1
9	Jackson White	2035	120	205	135	1
1...	Jackson Taylor	2045	90	205	145	1



2. Which are the products supplied by the Supplier - Innovate Technologies?

```
SELECT S.Supplier_ID,S.Supplier_Name,P.Product_Id, P.Product_Name, P.Product_Desc,
P.Product_Category, P.Product_Price FROM Supplier S, Product P WHERE
S.Supplier_Name LIKE "%Innovate Technologies%" AND
S.Supplier_ID=P.Supplier_ID
ORDER BY P.Product_Id
```

	Supplier_ID	A Supplier_Name	Product_Id	A Product_Name	A Product_Desc	A Product_Category	Product_Price
1	1007	Innovate Technologies	8	Wireless Security Camera	Outdoor wireless sec...	Security	129.99






3. What are the 5 least billed products in the past 6 months?

```
SELECT
    P.Product_Name,
    SUM(PL.Line_QTY) AS TotalQuantity
FROM
    Invoice I
JOIN Product_Line PL ON I.Invoice_ID = PL.Invoice_ID
JOIN Product P ON P.Product_ID = PL.Product_ID
WHERE
    I.Invoice_date >= DATE_SUB(CURDATE(), INTERVAL 6 MONTH)
GROUP BY
    P.Product_Name
ORDER BY
    TotalQuantity ASC
LIMIT 5
```

 Product_Name	 TotalQuantity
Industrial Drill	12
Tablet	14
Bluetooth Speakers	15
Arduino Starter Kit	15
Graphics Card	17

4. What are the products that have stock on hand higher than 70 units?

```
SELECT S.Store_ID,P.Product_Desc, S.Product_ID, P.Product_Category,S.QTY
FROM
    Stock_on_hand S, Product P WHERE P.Product_ID = S.Product_ID
    AND S.QTY>70
```

 Store_ID	 Product_Desc	 Product_ID	 Product_Category	 QTY
201	Wireless Bluetooth speakers	2	Audio	75
202	Powerful circular saw for woodworki...	3	Power Tools	100
203	Heavy-duty industrial drill	4	Power Tools	200
205	Wi-Fi-enabled smart thermostat	6	Smart Home	120
208	Starter kit for Arduino projects	9	DIY Electronics	75
209	Desktop 3D printer for prototyping	10	3D Printers	95

5. List the Items sold in 202,208,205 Stores and priced in the range \$300-\$600?








SELECT DISTINCT

P.Product_ID, P.Product_Name, P.Product_Desc, P.Product_Price, P.Product_Category, I.Invoice_ID, I.Store_ID, S.Store_Name

FROM Invoice I, Store S, Product P, Product_Line PL

WHERE I.Store_ID=S.Store_ID AND PL.Product_ID=P.Product_ID AND I.Store_ID in(202,208,205) AND P.Product_Price BETWEEN 300 AND 600

ORDER BY P.Product_ID

 Product_ID	 Product_Name	 Product_Desc	 Product_Price	 Product_Category	 Invoice_ID	 Store_ID
1	Tablet	Android tablet with HD di...	299.99	Computers	102	202
1	Tablet	Android tablet with HD di...	299.99	Computers	112	202
1	Tablet	Android tablet with HD di...	299.99	Computers	122	202
1	Tablet	Android tablet with HD di...	299.99	Computers	132	202
5	Graphics Card	High-end graphics card f...	499.99	Computer Components	102	202
5	Graphics Card	High-end graphics card f...	499.99	Computer Components	135	205
5	Graphics Card	High-end graphics card f...	499.99	Computer Components	125	205
5	Graphics Card	High-end graphics card f...	499.99	Computer Components	145	205
5	Graphics Card	High-end graphics card f...	499.99	Computer Components	132	202
5	Graphics Card	High-end graphics card f...	499.99	Computer Components	108	208

6. What are the details of the customers who are from California State who billed during the years of 2024-26 ?

SELECT

C.Customer_Name, C.Customer_street, C.Customer_State, I.Invoice_ID, I.Invoice_Date
FROM

Invoice I






JOIN Product_Line PL ON I.Invoice_ID = PL.Invoice_ID

JOIN Product P ON P.Product_ID = PL.Product_ID

JOIN Customer C ON C.Customer_ID = I.Customer_ID

WHERE C.customer_state='CA'

order by I.invoice_date

	 Customer_Name	 Customer_street	 Customer_State	 Invoice_ID	 Invoice_Date
1	Sophie Wilson	777 Peach St	CA	120	2024-08-01
2	Sophie Wilson	777 Peach St	CA	120	2024-08-01
3	Liam Turner	999 Pineapple St	CA	131	2025-05-15
4	Liam Turner	999 Pineapple St	CA	131	2025-05-15
5	Olivia Turner	333 Spruce St	CA	140	2026-02-01
6	Olivia Turner	333 Spruce St	CA	140	2026-02-01
7	Ava Anderson	456 Cedar St	CA	148	2026-09-30
8	Ava Anderson	456 Cedar St	CA	148	2026-09-30

7. Who are the Top 5 Performing Store Employees ?

```
SELECT
  E.Emp_ID,E.Emp_Name
  E.Store_ID, S.Store_Name, SUM(I.Invoice_Total) As "Total_Billed_Amount"
  FROM Customer C, Invoice I, Employee E
WHERE I.Emp_ID = E.Emp_ID AND I.Invoice_date between
GROUP BY E.EMP_ID
ORDER by SUM(I.Invoice_Total) desc
LIMIT 5
```

	Emp_ID	Emp_Name	Store_ID	Store_Name	Total_Bill_Amount
1	4	Alice Williams	203	Village Home Goods	3451750
2	7	David Miller	206	Village Home Goods	2300625
3	10	Sophie Turner	209	Village Home Goods	2005000
4	9	Frank Wilson	208	Village Home Goods	1586875
5	8	Grace Taylor	207	Village Home Goods	1190000

8. Which State has the most Customers across all stores in the USA?

```
Select C.State, count(C.Customer_State) as 'Number of Customers'
from Customer C
Group by C.Customer_State
Order by count(C.Customer_State) desc
Limit 1;
```

Customer_State	Number of Customers
FL	5

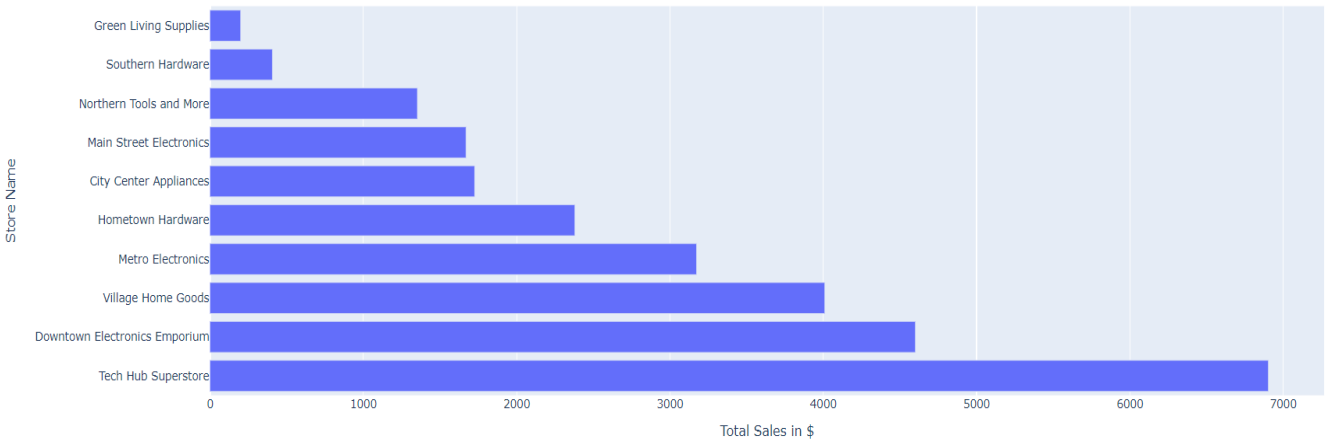
STEP 4: Enterprise (web) Database Dashboard

The analytical group dashboard can be found under the following link:

[Retail_EDM_Project.ipynb](#)

Dashboards for Sales generated in each of the stores, based on invoices generated, have been built. The Python visualization library - Plotly has been used to build the bar graph and Pie charts. Also, Number of Customers in Each State has been visualized. Furthermore, No. of Products by Category have been depicted using a Pie chart.

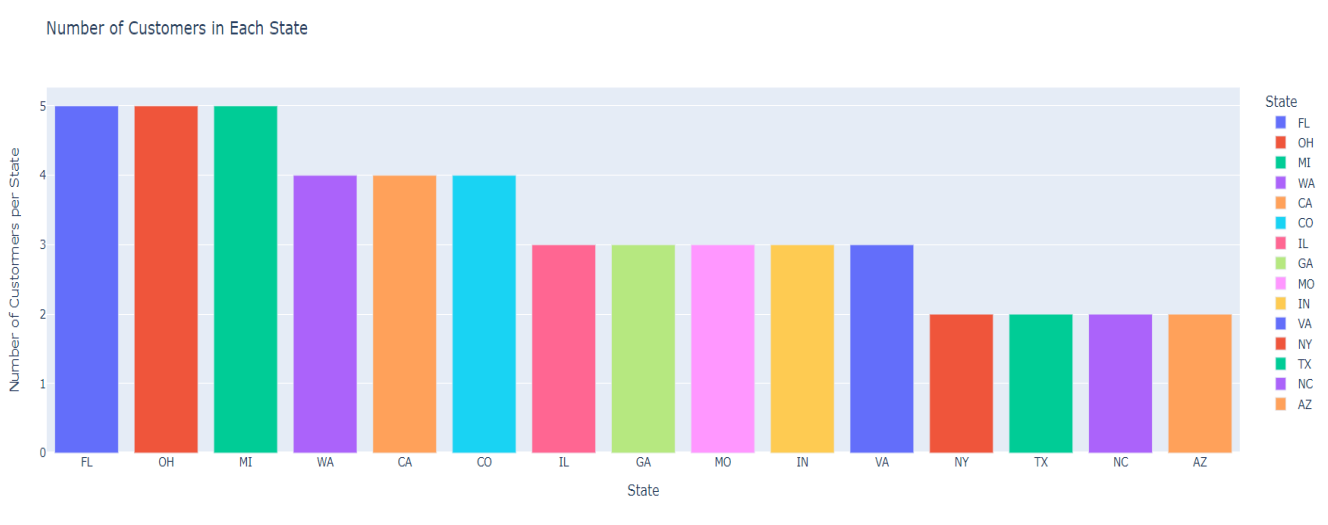
1) Store-wise Total Sales Generated



2) Percentage of Products Available in Each Product Category



3) Number of Customers in each State of USA



4) Product Category-wise Sales Generated

