**Vijay Sales Shop**

**Database Management System**

**Overview:-**

Vijay Sales Shop is a retail business specializing in electronic products and home appliances. To effectively manage its operations, The Raw data of a SQL Database Management System (DBMS) is designed to handle essential aspects such as product inventory, sales transactions, customer data, supplier details, and purchase records.

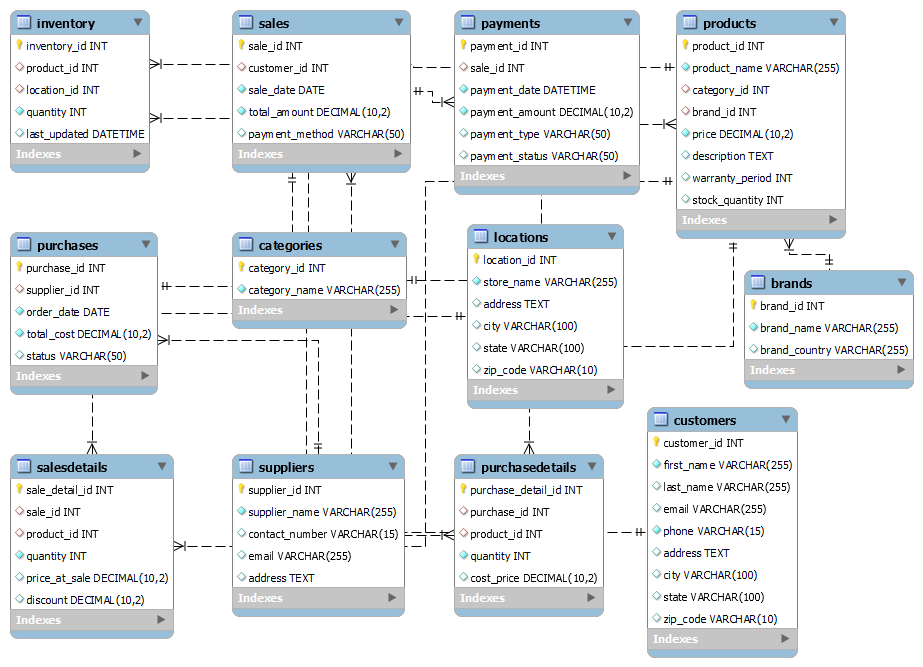
The Vijay Sales database structure includes tables for:

1. **Products** - Details of all available products, their categories, brands, prices, and stock levels.
2. **Categories** - Classification of products for easy navigation and inventory organization.
3. **Brands** - Information about various brands and their origins.
4. **Customers** - Records of customer information for tracking purchases and enhancing customer service.
5. **Sales** - Transactional data for sales, including total amounts and payment methods.
6. **SalesDetails** - Breakdown of each sale, showing product quantity, sale price, and discounts.
7. **Inventory** - Tracks product availability across different store locations.
8. **Locations** - Information about each store location, including addresses.
9. **Suppliers** - Details of suppliers providing products, including contact and ordering information.
10. **Purchases** - Purchase orders from suppliers, including costs and order status.
11. **PurchaseDetails** - Specific details for each purchase order, such as product quantity and price.
12. **Payments** - Records of customer payments for sales and their statuses.

The SQL DBMS enables Vijay Sales Shop to maintain smooth operations by accurately tracking inventory, processing transactions efficiently, and managing supplier relations. This structure helps ensure real-time data availability for decision-making, improves customer service, and supports inventory control.

**Raw Diagram**

**ER Diagram**

****

**Create Tables: -**

CReate database **vijay\_sales\_shop**;

use **vijay\_sales\_shop**;

-- 1. **Products**

CREATE TABLE **Products** (

product\_id INT PRIMARY KEY AUTO\_INCREMENT,

product\_name VARCHAR(255) NOT NULL,

category\_id INT,

brand\_id INT,

price DECIMAL(10, 2) NOT NULL,

description TEXT,

warranty\_period INT,

stock\_quantity INT,

FOREIGN KEY (category\_id) REFERENCES Categories(category\_id),

FOREIGN KEY (brand\_id) REFERENCES Brands(brand\_id)

);

-- 2. **Categories**

CREATE TABLE **Categories** (

category\_id INT PRIMARY KEY AUTO\_INCREMENT,

category\_name VARCHAR(255) NOT NULL

);

-- 3. **Brands**

CREATE TABLE **Brands** (

brand\_id INT PRIMARY KEY AUTO\_INCREMENT,

brand\_name VARCHAR(255) NOT NULL,

brand\_country VARCHAR(255)

);

-- 4. **Customers**

CREATE TABLE **Customers** (

customer\_id INT PRIMARY KEY AUTO\_INCREMENT,

first\_name VARCHAR(255) NOT NULL,

last\_name VARCHAR(255),

email VARCHAR(255) UNIQUE,

phone VARCHAR(15) NOT NULL,

address TEXT,

city VARCHAR(100),

state VARCHAR(100),

zip\_code VARCHAR(10)

);

-- 5. **Sales**

CREATE TABLE **Sales** (

sale\_id INT PRIMARY KEY AUTO\_INCREMENT,

customer\_id INT,

sale\_date DATE NOT NULL,

total\_amount DECIMAL(10, 2) NOT NULL,

payment\_method VARCHAR(50),

FOREIGN KEY (customer\_id) REFERENCES Customers(customer\_id)

);

-- 6. **SalesDetails**

CREATE TABLE **SalesDetails** (

sale\_detail\_id INT PRIMARY KEY AUTO\_INCREMENT,

sale\_id INT,

product\_id INT,

quantity INT NOT NULL,

price\_at\_sale DECIMAL(10, 2),

discount DECIMAL(10, 2),

FOREIGN KEY (sale\_id) REFERENCES Sales(sale\_id),

FOREIGN KEY (product\_id) REFERENCES Products(product\_id)

);

-- 7. **Inventory**

CREATE TABLE **Inventory** (

inventory\_id INT PRIMARY KEY AUTO\_INCREMENT,

product\_id INT,

location\_id INT,

quantity INT NOT NULL,

last\_updated DATETIME,

FOREIGN KEY (product\_id) REFERENCES Products(product\_id),

FOREIGN KEY (location\_id) REFERENCES Locations(location\_id)

);

-- 8. **Locations**

CREATE TABLE **Locations** (

location\_id INT PRIMARY KEY AUTO\_INCREMENT,

store\_name VARCHAR(255) NOT NULL,

address TEXT,

city VARCHAR(100),

state VARCHAR(100),

zip\_code VARCHAR(10)

);

-- 9. **Suppliers**

CREATE TABLE **Suppliers** (

supplier\_id INT PRIMARY KEY AUTO\_INCREMENT,

supplier\_name VARCHAR(255) NOT NULL,

contact\_number VARCHAR(15),

email VARCHAR(255) UNIQUE,

address TEXT

);

-- 10. **Purchases**

CREATE TABLE **Purchases** (

purchase\_id INT PRIMARY KEY AUTO\_INCREMENT,

supplier\_id INT,

order\_date DATE NOT NULL,

total\_cost DECIMAL(10, 2) NOT NULL,

status VARCHAR(50),

FOREIGN KEY (supplier\_id) REFERENCES Suppliers(supplier\_id)

);

-- 11. **PurchaseDetails**

CREATE TABLE **PurchaseDetails** (

purchase\_detail\_id INT PRIMARY KEY AUTO\_INCREMENT,

purchase\_id INT,

product\_id INT,

quantity INT NOT NULL,

cost\_price DECIMAL(10, 2),

FOREIGN KEY (purchase\_id) REFERENCES Purchases(purchase\_id),

FOREIGN KEY (product\_id) REFERENCES Products(product\_id)

);

-- 12. **Payments**

CREATE TABLE **Payments** (

payment\_id INT PRIMARY KEY AUTO\_INCREMENT,

sale\_id INT,

payment\_date DATETIME NOT NULL,

payment\_amount DECIMAL(10, 2) NOT NULL,

payment\_type VARCHAR(50),

payment\_status VARCHAR(50),

FOREIGN KEY (sale\_id) REFERENCES Sales(sale\_id)

);

**Insert Values of all Tables**

* **Products**

INSERT INTO **Products** (product\_id, product\_name, category\_id, brand\_id, price, description, warranty\_period, stock\_quantity)

VALUES

(1, 'Samsung LED TV 43"', 1, 1, 29999.99, '43-inch LED TV with Full HD resolution', 24, 10),

(2, 'iPhone 14', 2, 2, 79999.99, '128GB, Midnight Black', 12, 15),

(3, 'Dell Inspiron Laptop', 3, 3, 45999.99, 'Core i5, 8GB RAM, 512GB SSD', 12, 8),

(4, 'Sony Headphones WH-1000XM4', 4, 4, 19999.99, 'Noise-canceling wireless headphones', 12, 20),

(5, 'LG Refrigerator 190L', 5, 5, 15999.99, '190L single-door refrigerator with smart inverter compressor', 24, 12),

(6, 'HP Pavilion Laptop 14"', 3, 6, 60999.99, 'Core i7, 16GB RAM, 1TB SSD', 12, 7),

(7, 'Canon EOS 1500D DSLR Camera', 6, 7, 35999.99, '24.1MP DSLR camera with dual-lens kit', 12, 5),

(8, 'Philips Air Fryer HD9216', 7, 8, 8499.99, 'Rapid Air Technology for healthier frying', 12, 18),

(9, 'Sony Bravia 55" LED TV', 1, 9, 54999.99, '4K UHD Smart LED TV with HDR and Android OS', 24, 15),

(10, 'Samsung Galaxy S21', 2, 2, 69999.99, '128GB storage, 8GB RAM, Exynos 2100', 12, 20),

(11, 'Dell Inspiron 15"', 3, 10, 48999.99, 'Core i5, 8GB RAM, 512GB SSD, Windows 10', 12, 10),

(12, 'Whirlpool Washing Machine 6.5kg', 5, 11, 17999.99, '6.5kg fully automatic front-load', 24, 8),

(13, 'Nikon D5600 DSLR Camera', 6, 12, 44999.99, '24.2MP DSLR with 18-55mm lens', 12, 5),

(14, 'Bajaj Microwave Oven 20L', 7, 13, 5999.99, 'Convection microwave with digital display', 12, 12);

* **Categories**

INSERT INTO **Categories** (category\_id, category\_name)

VALUES

(1, 'Television'),

(2, 'Mobile Phones'),

(3, 'Laptops'),

(4, 'Headphones'),

(5, 'Refrigerators'),

(6, 'Cameras'),

(7, 'Kitchen Appliances'),

(8, 'Microwaves'),

(9, 'Washing Machines');

* **Brands**

INSERT INTO **Brands** (brand\_id, brand\_name, brand\_country)

VALUES

(1, 'Samsung', 'South Korea'),

(2, 'Apple', 'United States'),

(3, 'Dell', 'United States'),

(4, 'Sony', 'Japan'),

(5, 'LG', 'South Korea'),

(6, 'HP', 'United States'),

(7, 'Canon', 'Japan'),

(8, 'Philips', 'Netherlands'),

(9, 'Sony', 'Japan'),

(10, 'Dell', 'United States'),

(11, 'Whirlpool', 'United States'),

(12, 'Nikon', 'Japan'),

(13, 'Bajaj', 'India');

* **Customers**

INSERT INTO **Customers** (customer\_id, first\_name, last\_name, email, phone, address, city, state, zip\_code)

VALUES

(1, 'Ravi', 'Sharma', 'ravi.sharma@gmail.com', '9123456789', '123 Main Street', 'Mumbai', 'Maharashtra', '400001'),

(2, 'Anita', 'Verma', 'anita.verma@gmail.com', '9234567890', '456 Park Avenue', 'Pune', 'Maharashtra', '411001'),

(3, 'Raj', 'Patel', 'raj.patel@gmail.com', '9345678901', '789 Hill Road', 'Surat', 'Gujarat', '395007'),

(4, 'Priya', 'Kumar', 'priya.kumar@gmail.com', '9456123456', 'Apartment 42, Sector 22', 'Noida', 'Uttar Pradesh', '201301'),

(5, 'Amit', 'Singh', 'amit.singh@gmail.com', '9561234567', 'Bungalow 19, Shivaji Nagar', 'Nagpur', 'Maharashtra', '440010'),

(6, 'Neha', 'Joshi', 'neha.joshi@gmail.com', '9671234568', 'Flat 13B, Race Course Road', 'Indore', 'Madhya Pradesh', '452001'),

(7, 'Rajesh', 'Patel', 'rajesh.patel@gmail.com', '9876543210', 'House 29, MG Road', 'Pune', 'Maharashtra', '411001'),

(8, 'Suman', 'Sharma', 'suman.sharma@gmail.com', '9765432189', 'Apartment 4A, South City', 'Kolkata', 'West Bengal', '700047'),

(9, 'Akhil', 'Verma', 'akhil.verma@gmail.com', '8896541234', 'Building 21, Banjara Hills', 'Hyderabad', 'Telangana', '500034');

* **Sales**

INSERT INTO **Sales** (sale\_id, customer\_id, sale\_date, total\_amount, payment\_method)

VALUES

(1, 1, '2024-10-01', 109999.97, 'Credit Card'),

(2, 2, '2024-10-02', 79999.99, 'Debit Card'),

(3, 3, '2024-10-03', 29999.99, 'Cash'),

(4, 4, '2024-10-04', 15999.99, 'UPI'),

(5, 5, '2024-10-05', 60999.99, 'Credit Card'),

(6, 6, '2024-10-06', 8499.99, 'Debit Card'),

(7, 7, '2024-10-07', 54999.99, 'Credit Card'),

(8, 8, '2024-10-08', 69999.99, 'UPI'),

(9, 9, '2024-10-09', 17999.99, 'Cash');

* **SalesDetails**

INSERT INTO **SalesDetails** (sale\_detail\_id, sale\_id, product\_id, quantity, price\_at\_sale, discount)

VALUES

(1, 1, 1, 1, 29999.99, 0),

(2, 1, 2, 1, 79999.99, 0),

(3, 2, 2, 1, 79999.99, 0),

(4, 3, 1, 1, 29999.99, 0),

(5, 4, 5, 1, 15999.99, 500),

(6, 5, 6, 1, 60999.99, 0),

(7, 6, 8, 1, 8499.99, 200),

(8, 7, 9, 1, 54999.99, 0),

(9, 8, 10, 1, 69999.99, 2000),

(10, 9, 12, 1, 17999.99, 500);

* **Inventory**

INSERT INTO **Inventory** (inventory\_id, product\_id, location\_id, quantity, last\_updated)

VALUES

(1, 1, 1, 10, '2024-10-01'),

(2, 2, 1, 15, '2024-10-01'),

(3, 3, 2, 8, '2024-10-01'),

(4, 4, 2, 20, '2024-10-01'),

(5, 5, 1, 12, '2024-10-04'),

(6, 6, 1, 7, '2024-10-05'),

(7, 7, 2, 5, '2024-10-06'),

(8, 8, 2, 18, '2024-10-06'),

(9, 9, 3, 15, '2024-10-07'),

(10, 10, 2, 20, '2024-10-08'),

(11, 11, 1, 10, '2024-10-09'),

(12, 12, 2, 8, '2024-10-09');

* **Locations**

INSERT INTO **Locations** (location\_id, store\_name, address, city, state, zip\_code)

VALUES

(1, 'Vijay Sales - Andheri', 'Plot 21, Andheri West', 'Mumbai', 'Maharashtra', '400053'),

(2, 'Vijay Sales - Bandra', 'Shop No. 10, Linking Road', 'Mumbai', 'Maharashtra', '400050'),

(3, 'Vijay Sales - Thane', 'Shop No. 22, Ghodbunder Road', 'Thane', 'Maharashtra', '400601'),

(4, 'Vijay Sales - Vashi', 'Plot 5, Sector 17', 'Navi Mumbai', 'Maharashtra', '400703'),

(5, 'Vijay Sales - Andheri', 'Plot No. 9, Andheri West', 'Mumbai', 'Maharashtra', '400058'),

(6, 'Vijay Sales - Kandivali', 'Station Road, Kandivali East', 'Mumbai', 'Maharashtra', '400101');

* **Suppliers**

INSERT INTO **Suppliers** (supplier\_id, supplier\_name, contact\_number, email, address)

VALUES

(1, 'Samsung Electronics', '9876543210', 'contact@samsung.com', 'Seoul, South Korea'),

(2, 'Apple Inc.', '9765432109', 'sales@apple.com', 'Cupertino, CA, USA'),

(3, 'Dell India', '9654321098', 'info@dell.com', 'Bangalore, Karnataka, India'),

(4, 'LG Electronics', '9988776655', 'lg@lgelectronics.com', 'Seoul, South Korea'),

(5, 'HP India', '9876543212', 'info@hp.com', 'Bengaluru, Karnataka, India'),

(6, 'Canon India', '9765432107', 'support@canon.in', 'Noida, Uttar Pradesh, India'),

(7, 'Philips India', '9654321096', 'service@philips.com', 'Gurgaon, Haryana, India'),

(8, 'Sony India', '9898989898', 'support@sony.co.in', 'Tokyo, Japan'),

(9, 'Whirlpool India', '9797979797', 'contact@whirlpool.com', 'Pune, Maharashtra, India'),

(10, 'Nikon India', '9696969696', 'service@nikon.in', 'Tokyo, Japan'),

(11, 'Bajaj Electricals', '9595959595', 'support@bajajelectricals.com', 'Mumbai, Maharashtra, India');

* **Purchases**

INSERT INTO **Purchases** (purchase\_id, supplier\_id, order\_date, total\_cost, status)

VALUES

(1, 1, '2024-09-01', 250000.00, 'Ordered'),

(2, 2, '2024-09-10', 500000.00, 'Received'),

(3, 3, '2024-09-15', 150000.00, 'Ordered'),

(4, 4, '2024-10-04', 120000.00, 'Ordered'),

(5, 5, '2024-10-05', 250000.00, 'Received'),

(6, 6, '2024-10-06', 180000.00, 'Ordered'),

(7, 8, '2024-10-07', 400000.00, 'Ordered'),

(8, 9, '2024-10-08', 150000.00, 'Received'),

(9, 10, '2024-10-09', 250000.00, 'Pending');

* **PurchaseDetails**

INSERT INTO **PurchaseDetails** (purchase\_detail\_id, purchase\_id, product\_id, quantity, cost\_price)

VALUES

(1, 1, 1, 10, 25000.00),

(2, 2, 2, 5, 75000.00),

(3, 3, 3, 3, 50000.00),

(4, 4, 5, 10, 12000.00),

(5, 5, 6, 5, 50000.00),

(6, 6, 7, 3, 60000.00),

(7, 7, 9, 8, 45000.00),

(8, 8, 12, 6, 12000.00),

(9, 9, 13, 4, 40000.00);

* **Payments**

INSERT INTO **Payments** (payment\_id, sale\_id, payment\_date, payment\_amount, payment\_type, payment\_status)

VALUES

(1, 1, '2024-10-01', 109999.97, 'Credit Card', 'Completed'),

(2, 2, '2024-10-02', 79999.99, 'Debit Card', 'Completed'),

(3, 3, '2024-10-03', 29999.99, 'Cash', 'Completed'),

(4, 4, '2024-10-04', 15999.99, 'UPI', 'Completed'),

(5, 5, '2024-10-05', 60999.99, 'Credit Card', 'Completed'),

(6, 6, '2024-10-06', 8499.99, 'Debit Card', 'Completed'),

(7, 7, '2024-10-07', 54999.99, 'Credit Card', 'Completed'),

(8, 8, '2024-10-08', 69999.99, 'UPI', 'Completed'),

(9, 9, '2024-10-09', 17999.99, 'Cash', 'Completed');

\*\*\* **Select & Desc** \*\*\*

A black and white text

Description automatically generatedselect \* from **Products**;

select \* from **Categories**;A screenshot of a computer program

Description automatically generated

select \* from **Brands**;A screenshot of a computer code

Description automatically generated

select \* from **Customers**;A close-up of a table

Description automatically generated

select \* from **Sales**;A table of numbers with numbers

Description automatically generated with medium confidence

select \* from **SalesDetails**;A table of numbers and lines

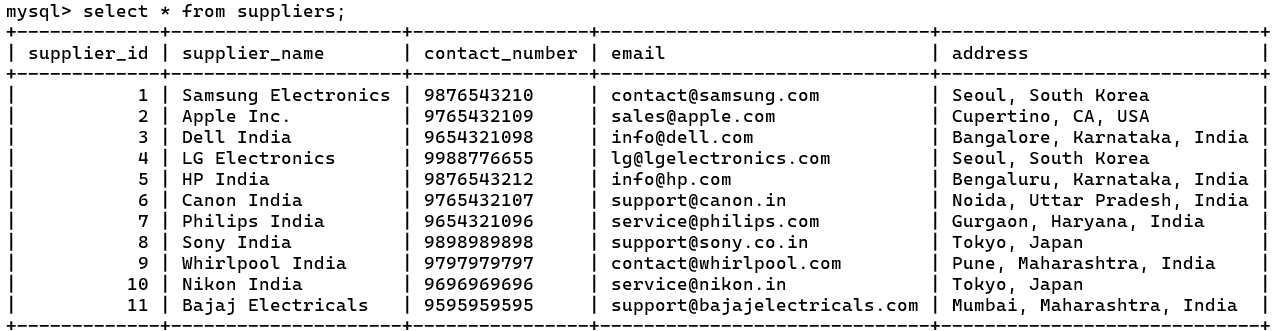
Description automatically generated with medium confidence

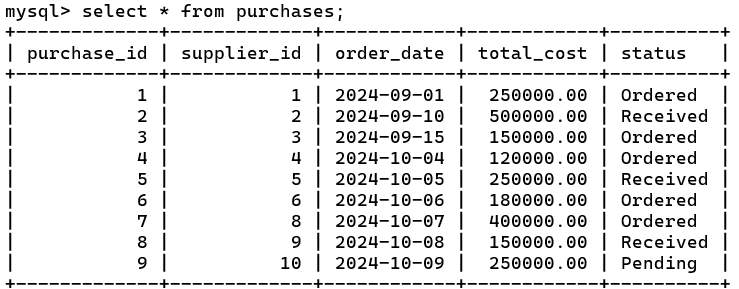
select \* from **Inventory**;A screenshot of a data

Description automatically generated

select \* from **Locations**; A close-up of a document

Description automatically generated

select \* from **Suppliers**;

select \* from **Purchases**;

select \* from **PurchaseDetails**;A screenshot of a screen

Description automatically generated

select \* from **Payments**;A screenshot of a document

Description automatically generated

**DESC**

DESC **Products**; A table with black text

Description automatically generated with medium confidence

DESC **Categories**;A close up of a grid

Description automatically generated

DESC **Brands**;A grid of black text

Description automatically generated with medium confidence

DESC **Customers**;A table with numbers and letters

Description automatically generated with medium confidence

DESC **Sales**;A close-up of a grid

Description automatically generated

DESC **SalesDetails**;A table of text and numbers

Description automatically generated with medium confidence

DESC **Inventory**;A close-up of a chart

Description automatically generated

DESC **Locations**;A table of numbers and letters

Description automatically generated with medium confidence

DESC **Suppliers**;A close-up of a grid

Description automatically generated

DESC **Purchases**;A close-up of a table

Description automatically generated

DESC **PurchaseDetails**;A white sheet with black text

Description automatically generated

DESC **Payments**;A close-up of a chart

Description automatically generated

\*\*\* **Questions** \*\*\*

-- How many SHop are in Mumbai?

SELECT store\_name, address ,city

FROM locations

WHERE city = 'Mumbai';A close-up of a address

Description automatically generated

-- How many customers are from Mumbai?

SELECT first\_name, last\_name ,city,state

FROM Customers

WHERE state = 'Maharashtra';A white paper with black text

Description automatically generated

-- How many Brands are from Japan?

SELECT Brand\_name

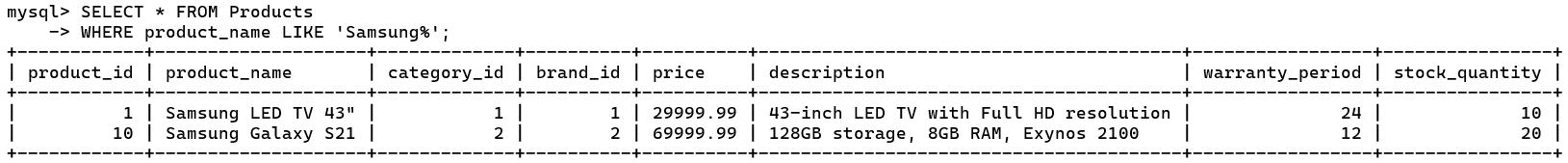
FROM Brands

A screen shot of a computer

Description automatically generatedWHERE brand\_country = 'Japan'

-- find all products with names that start with "Samsung":

SELECT \* FROM Products

WHERE product\_name LIKE 'Samsung%';

-- find all products where the second character in the name is "a":

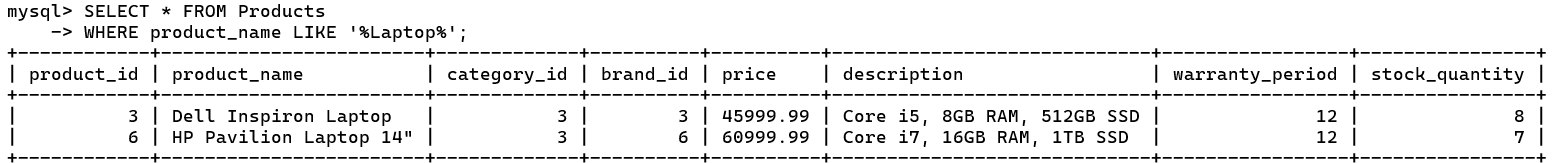
SELECT \* FROM Products

WHERE product\_name LIKE '\_a%';A close up of a white paper

Description automatically generated with medium confidence

-- find all products that contain "Laptop" in their name:

SELECT \* FROM Products

WHERE product\_name LIKE '%Laptop%';

-- What are the first names, last names, and email addresses of customers

-- whose first names start with the letter "A"?

SELECT first\_name, last\_name, email

FROM Customers

WHERE first\_name LIKE 'A%';A white background with black text

Description automatically generated

-- What is the total revenue generated from sales?

SELECT SUM(total\_amount) AS total\_revenue

FROM Sales;A white background with black text

Description automatically generated

-- What is the total number of customers and the total sales amount?

SELECT (SELECT COUNT(\*) FROM Customers) AS total\_customers,

(SELECT SUM(total\_amount) FROM Sales) AS total\_sales;A close-up of a number

Description automatically generated

-- What is the list of all products along with their category, brand, price, and stock quantity?

SELECT

p.product\_name,

c.category\_name,

b.brand\_name,

p.price,

p.stock\_quantity

FROM Products p

JOIN Categories c ON p.category\_id = c.category\_id

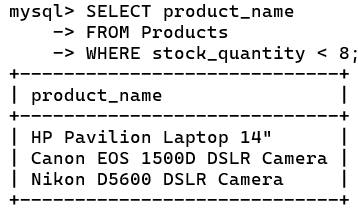
JOIN Brands b ON p.brand\_id = b.brand\_id;A screenshot of a computer

Description automatically generated

-- Which products are currently below 8 stocks?

SELECT product\_name

FROM Products

WHERE stock\_quantity < 8;

-- What is the total amount spent by each customer?

SELECT

c.first\_name,

c.last\_name,

SUM(s.total\_amount) AS total\_spent

FROM Sales s

JOIN Customers c ON s.customer\_id = c.customer\_id

GROUP BY c.customer\_id, c.first\_name, c.last\_name;A screenshot of a computer

Description automatically generated

-- What is the total quantity sold and total revenue for each product?

SELECT

p.product\_name,

SUM(sd.quantity) AS total\_quantity\_sold,

SUM(sd.price\_at\_sale \* sd.quantity) AS total\_revenue

FROM SalesDetails sd

JOIN Products p ON sd.product\_id = p.product\_id

GROUP BY p.product\_name;A white paper with black text

Description automatically generated

-- Who are the top 5 customers based on total purchase amount?

SELECT

c.first\_name,

c.last\_name,

SUM(s.total\_amount) AS total\_spent

FROM Sales s

JOIN Customers c ON s.customer\_id = c.customer\_id

GROUP BY c.customer\_id, c.first\_name, c.last\_name

ORDER BY total\_spent DESC

LIMIT 5;

A screenshot of a computer code

Description automatically generated

-- What is the inventory count for each product across all store locations?

SELECT

l.store\_name,

p.product\_name,

i.quantity AS stock\_quantity

FROM Inventory i

JOIN Products p ON i.product\_id = p.product\_id

A screenshot of a computer

Description automatically generatedJOIN Locations l ON i.location\_id = l.location\_id;

-- How much revenue is generated by each payment method?

SELECT

payment\_method,

SUM(total\_amount) AS total\_revenue

FROM Sales

GROUP BY payment\_method;A screenshot of a computer code

Description automatically generated

-- What is the average discount offered on each product?

SELECT

p.product\_name,

AVG(sd.discount) AS average\_discount

FROM SalesDetails sd

JOIN Products p ON sd.product\_id = p.product\_id

GROUP BY p.product\_name;

A screenshot of a computer program

Description automatically generated

-- List the suppliers and the status of their latest purchase orders.

SELECT s.supplier\_name,

p.order\_date,

p.status

FROM Purchases p

JOIN Suppliers s ON p.supplier\_id = s.supplier\_id

A screenshot of a computer

Description automatically generatedORDER BY s.supplier\_name, p.order\_date DESC;

-- Which products have stock below 10 units?

SELECT

product\_name,

stock\_quantity

FROM Products

WHERE stock\_quantity < 10;A close-up of a computer code

Description automatically generated

-- What are the sales made between specific dates?

SELECT sale\_id, customer\_id, sale\_date, total\_amount

FROM Sales

A screenshot of a computer code

Description automatically generatedWHERE sale\_date BETWEEN '2024-10-01' AND '2024-10-31';

-- What products has a specific customer purchased?

SELECT

c.first\_name,

c.last\_name,

p.product\_name,

sd.quantity,

sd.price\_at\_sale

FROM SalesDetails sd

JOIN Sales s ON sd.sale\_id = s.sale\_id

JOIN Customers c ON s.customer\_id = c.customer\_id

JOIN Products p ON sd.product\_id = p.product\_id

WHERE c.customer\_id = 1;A screenshot of a computer code

Description automatically generated

-- What are the contact details of all customers?

SELECT first\_name, last\_name, email, phone FROM Customers;A close-up of a computer screen

Description automatically generated

-- Which products have generated the highest revenue?

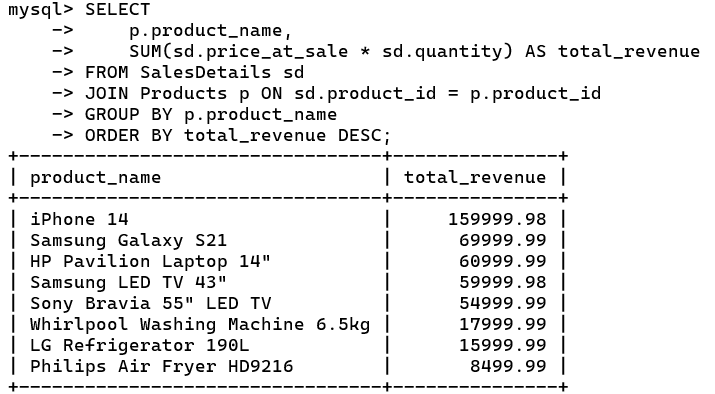
SELECT p.product\_name,

SUM(sd.price\_at\_sale \* sd.quantity) AS total\_revenue

FROM SalesDetails sd

JOIN Products p ON sd.product\_id = p.product\_id

GROUP BY p.product\_name

ORDER BY total\_revenue DESC;

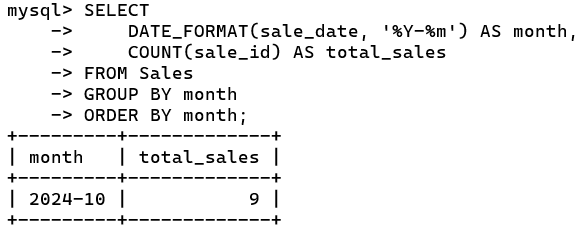
-- How many sales were made each month?

SELECT DATE\_FORMAT(sale\_date, '%Y-%m') AS month,

COUNT(sale\_id) AS total\_sales

FROM Sales

GROUP BY month

ORDER BY month;

-- What is the total sales amount for each product category?

SELECT c.category\_name,

SUM(s.total\_amount) AS total\_sales

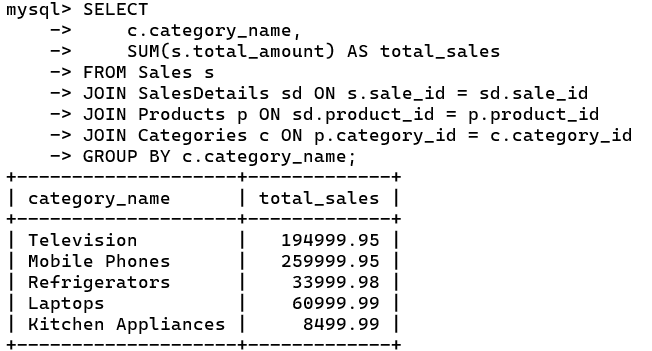
FROM Sales s

JOIN SalesDetails sd ON s.sale\_id = sd.sale\_id

JOIN Products p ON sd.product\_id = p.product\_id

JOIN Categories c ON p.category\_id = c.category\_id

GROUP BY c.category\_name;



-- How many purchases has each customer made?

SELECT

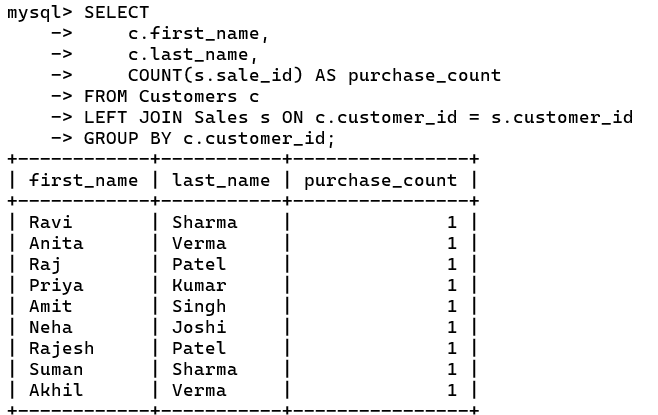
c.first\_name,

c.last\_name,

COUNT(s.sale\_id) AS purchase\_count

FROM Customers c

LEFT JOIN Sales s ON c.customer\_id = s.customer\_id

GROUP BY c.customer\_id;

-- Which products are currently being offered at a discount?

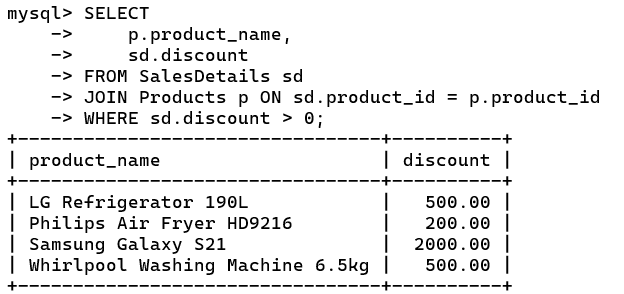
SELECT

p.product\_name,

sd.discount

FROM SalesDetails sd

JOIN Products p ON sd.product\_id = p.product\_id

WHERE sd.discount > 0;

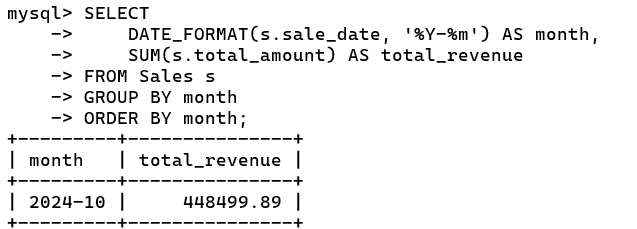
-- What is the total revenue generated each month?

SELECT DATE\_FORMAT(s.sale\_date, '%Y-%m') AS month,

SUM(s.total\_amount) AS total\_revenue

FROM Sales s

GROUP BY month

ORDER BY month;

-- Which suppliers have pending purchase orders?

SELECT

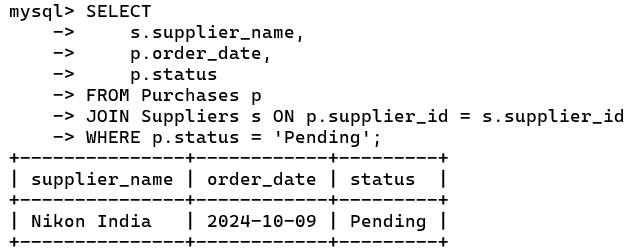
s.supplier\_name,

p.order\_date,

p.status

FROM Purchases p

JOIN Suppliers s ON p.supplier\_id = s.supplier\_id

WHERE p.status = 'Pending';

-- What is the total sales amount for each brand?

SELECT

b.brand\_name,

SUM(s.total\_amount) AS total\_sales

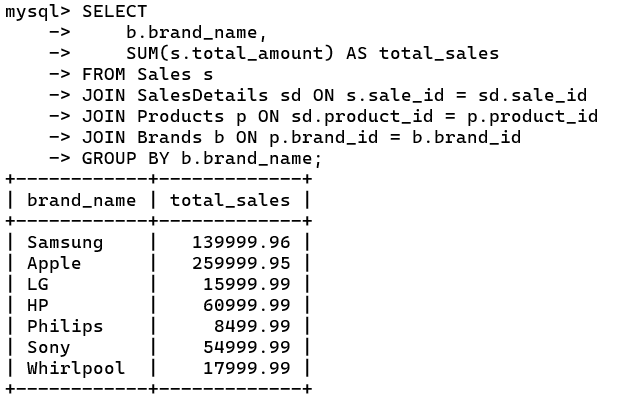
FROM Sales s

JOIN SalesDetails sd ON s.sale\_id = sd.sale\_id

JOIN Products p ON sd.product\_id = p.product\_id

JOIN Brands b ON p.brand\_id = b.brand\_id

GROUP BY b.brand\_name;



-- Which products have the highest number of sales?

SELECT

p.product\_name,

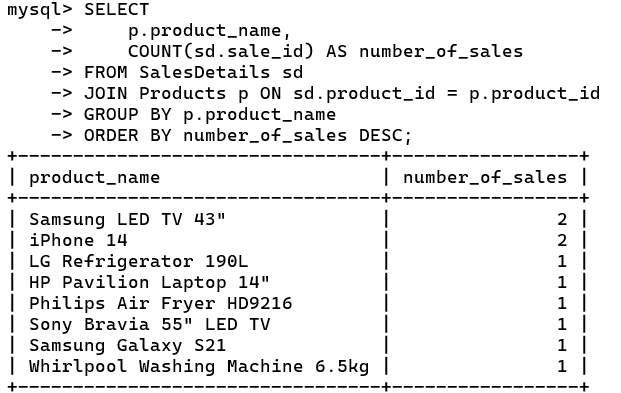
COUNT(sd.sale\_id) AS number\_of\_sales

FROM SalesDetails sd

JOIN Products p ON sd.product\_id = p.product\_id

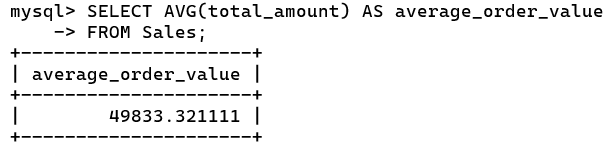
GROUP BY p.product\_name

ORDER BY number\_of\_sales DESC;



-- What is the average order value across all sales?

SELECT AVG(total\_amount) AS average\_order\_value

FROM Sales;

-- What is the spending trend of customers over the last six months?

SELECT

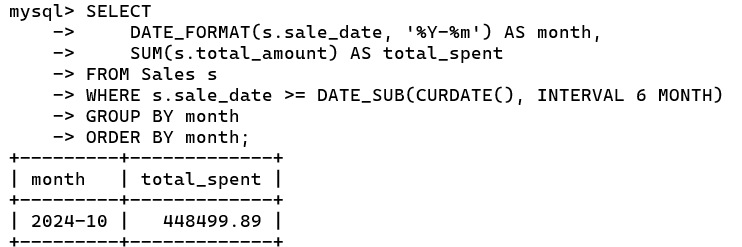
DATE\_FORMAT(s.sale\_date, '%Y-%m') AS month,

SUM(s.total\_amount) AS total\_spent

FROM Sales s

WHERE s.sale\_date >= DATE\_SUB(CURDATE(), INTERVAL 6 MONTH)

GROUP BY month

ORDER BY month;

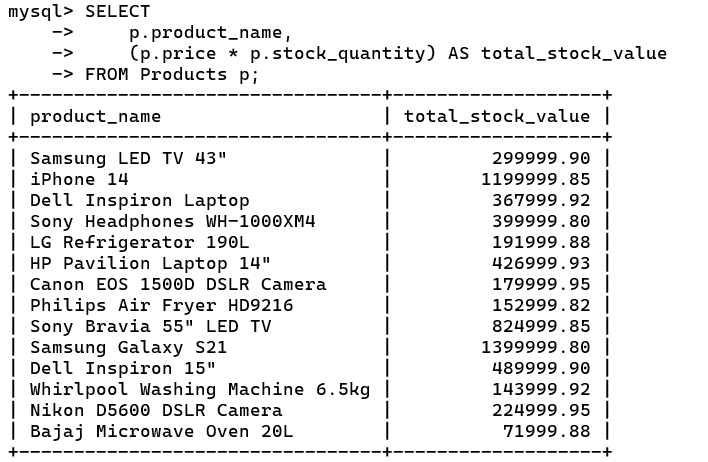
-- What is the total value of stock for each product?

SELECT

p.product\_name,

(p.price \* p.stock\_quantity) AS total\_stock\_value

FROM Products p;



**Conclusion** : -

The **Vijay Sales** SQL Database Management System (DBMS) provides a structured and efficient way to manage the essential operations of a retail business specializing in electronics and home appliances. By organizing data into well-defined tables—such as Products, Categories, Brands, Customers, Sales, Inventory, Locations, Suppliers, Purchases, and Payments—the database facilitates streamlined management of inventory, customer transactions, supplier relations, and financial records.

This robust system allows for real-time tracking of product availability, detailed sales analysis, customer insights, and efficient handling of supplier orders. The database enables Vijay Sales to make data-driven decisions, improving operational efficiency, customer service, and profitability. Additionally, with easy access to comprehensive transaction data and inventory control, Vijay Sales can respond quickly to market demands and customer needs, positioning it well for continued growth and success in a competitive retail environment.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

**\*\*\* THE END \*\*\***

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*