**\*\*Database Documentation: Sales Tracking Database\*\***

\*\*Introduction:\*\*

The Sales Tracking Database is designed to help our company effectively track sales data, customer data, and inventory data. This database will provide valuable insights to make informed business decisions and optimize our operations. The database is implemented using SQL and includes various tables to store the necessary information.

\*\*Database Schema:\*\*

The following tables constitute the Sales Tracking Database:

1. \*\*Product table:\*\*

- Columns: product\_id, name, description, price, category

- Purpose: Stores information about products, including product details and pricing.

2. \*\*Region table:\*\*

- Columns: region\_id, name

- Purpose: Stores information about different regions where products are sold.

3. \*\*Time Period table:\*\*

- Columns: time\_period\_id, name

- Purpose: Stores information about different time periods, enabling tracking of sales over time.

4. \*\*Customer table:\*\*

- Columns: customer\_id, name, address, age, loyalty\_status

- Purpose: Stores customer data, including demographics and loyalty status.

5. \*\*Purchase table:\*\*

- Columns: purchase\_id, customer\_id, time\_period\_id, total\_amount

- Purpose: Tracks individual purchases made by customers, recording purchase details and the total amount spent.

6. \*\*Purchase Product table (junction table):\*\*

- Columns: purchase\_id, product\_id

- Purpose: Represents the many-to-many relationship between purchases and products, linking purchases to the products bought.

7. \*\*Product Region table (junction table):\*\*

- Columns: product\_id, region\_id

- Purpose: Represents the many-to-many relationship between products and regions, indicating in which regions each product is sold.

8. \*\*Inventory table:\*\*

- Columns: inventory\_id, product\_id, stock\_level, availability\_status

- Purpose: Stores inventory data, including stock levels and availability status for each product.

\*\*Data Integrity Constraints:\*\*

To maintain data integrity, the database includes the following constraints:

- Primary keys are used to uniquely identify records in each table.

- Foreign keys establish relationships between tables, ensuring valid references between related data.

- Unique constraints are applied where necessary to enforce uniqueness of certain attributes.

- Not null constraints ensure that essential attributes cannot have empty values.

**Creating Tables:**

-- Product table

CREATE TABLE Product (

product\_id INT PRIMARY KEY,

name VARCHAR(100) NOT NULL,

description TEXT,

price DECIMAL(10, 2) NOT NULL,

category VARCHAR(50)

);

-- Region table

CREATE TABLE Region (

region\_id INT PRIMARY KEY,

name VARCHAR(100) NOT NULL

);

-- Time period table

CREATE TABLE TimePeriod (

time\_period\_id INT PRIMARY KEY,

name VARCHAR(50) NOT NULL

);

-- Customer table

CREATE TABLE Customer (

customer\_id INT PRIMARY KEY,

name VARCHAR(100) NOT NULL,

address TEXT,

age INT,

loyalty\_status VARCHAR(20)

);

-- Purchase table

CREATE TABLE Purchase (

purchase\_id INT PRIMARY KEY,

customer\_id INT,

time\_period\_id INT,

total\_amount DECIMAL(10, 2),

FOREIGN KEY (customer\_id) REFERENCES Customer(customer\_id),

FOREIGN KEY (time\_period\_id) REFERENCES TimePeriod(time\_period\_id)

);

-- Purchase product table (junction table)

CREATE TABLE PurchaseProduct (

purchase\_id INT,

product\_id INT,

PRIMARY KEY (purchase\_id, product\_id),

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

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

);

-- Product region table (junction table)

CREATE TABLE ProductRegion (

product\_id INT,

region\_id INT,

PRIMARY KEY (product\_id, region\_id),

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

FOREIGN KEY (region\_id) REFERENCES Region(region\_id)

);

-- Inventory table

CREATE TABLE Inventory (

inventory\_id INT PRIMARY KEY,

product\_id INT,

stock\_level INT,

availability\_status VARCHAR(20),

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

);

\*\*Inserting Sample Data:\*\*

Here are some sample records inserted into the tables:

**Customer table :**

| **customer\_id** | **name** | **address** | **age** | **loyalty\_status** |
| --- | --- | --- | --- | --- |
| 1 | John Doe | 123 Main St, City A | 30 | Gold |
| 2 | Jane Smith | 456 Maple Rd, City B | 28 | Silver |
| 3 | Michael Johnson | 789 Oak Ave, City C | 42 | Bronze |
| 4 | Emily Williams | 567 Pine St, City D | 25 | Silver |
| 5 | Robert Brown | 987 Elm St, City E | 35 | Platinum |

**Inventory table:**

| **inventory\_id** | **product\_id** | **stock\_level** | **availability\_status** |
| --- | --- | --- | --- |
| 1 | 1 | 100 | In Stock |
| 2 | 2 | 50 | In Stock |
| 3 | 3 | 75 | In Stock |
| 4 | 4 | 20 | Low Stock |
| 5 | 5 | 200 | In Stock |

**Product table:**

| **product\_id** | **name** | **description** | **price** | **category** |
| --- | --- | --- | --- | --- |
| 1 | t-shirt | comfortable cotton t-shirt | 20 | apparel |
| 2 | smartphone | advanced features | 699 | Electronics |
| 3 | running shoes | lightweight | 90 | footwear |
| 4 | headphones | wireless | 150 | electronics |
| 5 | Backpack | spacious | 45 | Accessories |

**Product region table:**

| **product\_id** | **region\_id** |
| --- | --- |
| 1 | 1 |
| 3 | 1 |
| 2 | 2 |
| 2 | 3 |
| 4 | 4 |

**Purchase table:**

| **purchase\_id** | **customer\_id** | **time\_period\_id** | **total\_amount** |
| --- | --- | --- | --- |
| **1** | **1** | **2** | **39.98** |
| **2** | **3** | **3** | **178.49** |
| **3** | **2** | **1** | **67.98** |
| **4** | **4** | **4** | **249** |
| **5** | **5** | **5** | **89.99** |

**Purchase product table:**

| **purchase\_id** | **product\_id** |
| --- | --- |
| **1** | **1** |
| **3** | **1** |
| **2** | **2** |
| **1** | **3** |
| **2** | **4** |

**Region table:**

| **region\_id** | **name** |
| --- | --- |
| **1** | **North America** |
| **2** | **Europe** |
| **3** | **Asia** |
| **4** | **South America** |
| **5** | **Australia** |

**Time period table:**

| **time\_period\_id** | **name** |
| --- | --- |
| **1** | **Q1 2023** |
| **2** | **Q2 2023** |
| **3** | **Q3 2023** |
| **4** | **Q4 2023** |
| **5** | **Q1 2024** |

**\*\*Data Analysis:\*\***

Once the database is populated with actual data, you can use SQL queries to perform various data analyses. Here are some example queries:

1. Total Sales Amount by Region:

SELECT r.name AS region\_name, SUM(pp.total\_amount) AS total\_sales\_amount

FROM Region r

JOIN ProductRegion pr ON r.region\_id = pr.region\_id

JOIN PurchaseProduct pp ON pr.product\_id = pp.product\_id

GROUP BY r.name;

**Output:**

| region\_name | total\_sales\_amount | |
| --- | --- | --- |
| North America | 147.94 |  |
| Europe | 178.49 |  |
| Asia | 178.49 |  |
| South America | 178.49 |  |

2. Top Loyal Customers:

SELECT c.name AS customer\_name, c.loyalty\_status, COUNT(pu.purchase\_id) AS purchase\_count

FROM Customer c

JOIN Purchase pu ON c.customer\_id = pu.customer\_id

GROUP BY c.name, c.loyalty\_status

ORDER BY purchase\_count DESC

LIMIT 5;

**Output:**

| customer\_name | loyalty\_status | purchase\_count | |
| --- | --- | --- | --- |
| John Doe | Gold | 1 |  |
| Jane Smith | Silver | 1 |  |
| Michael Johnson | Bronze | 1 |  |
| Emily Williams | Silver | 1 |  |
| Robert Brown | Platinum | 1 |  |

3. Product Availability:

SELECT p.name AS product\_name, i.stock\_level, i.availability\_status

FROM Product p

JOIN Inventory i ON p.product\_id = i.product\_id;

**Output:**

| product\_name | stock\_level | availability\_status | |
| --- | --- | --- | --- |
| t-shirt | 100 | In Stock |  |
| smartphone | 50 | In Stock |  |
| running shoes | 75 | In Stock |  |
| headphones | 20 | Low Stock | |
| Backpack | 200 | In Stock |  |