Marhba

Database Documentation

Table of Contents

1. Introduction:

Marhba database is a comprehensive system for storing and managing data related to various aspects of hypermarket operations. It comprises multiple tables that capture detailed information about products, customers, employees, suppliers, Departments, and other relevant entities. This database plays a crucial role in supporting and streamlining hypermarket operations by facilitating efficient data retrieval, analysis, and reporting.

1.1 Purpose:

The primary purpose of the Marhba database is to provide a centralized and organized structure for storing and managing data related to the hypermarket's operations. This data can then be used for various purposes, including:

1. Manage Sales operations:

- Track sales across multiple stores.
- Analyze customer patterns.
- Manage employee information and salaries.

2. Analyze sales trends and customer behavior:

- Identify popular products, categories and subcategories.
- Understand customer preferences.
- Providing store locations in the best-selling areas.

3. Optimize product sourcing:

- Track supplier performance and product availability.
- Identify cost-effective sourcing options.

4. Enhance customer service:

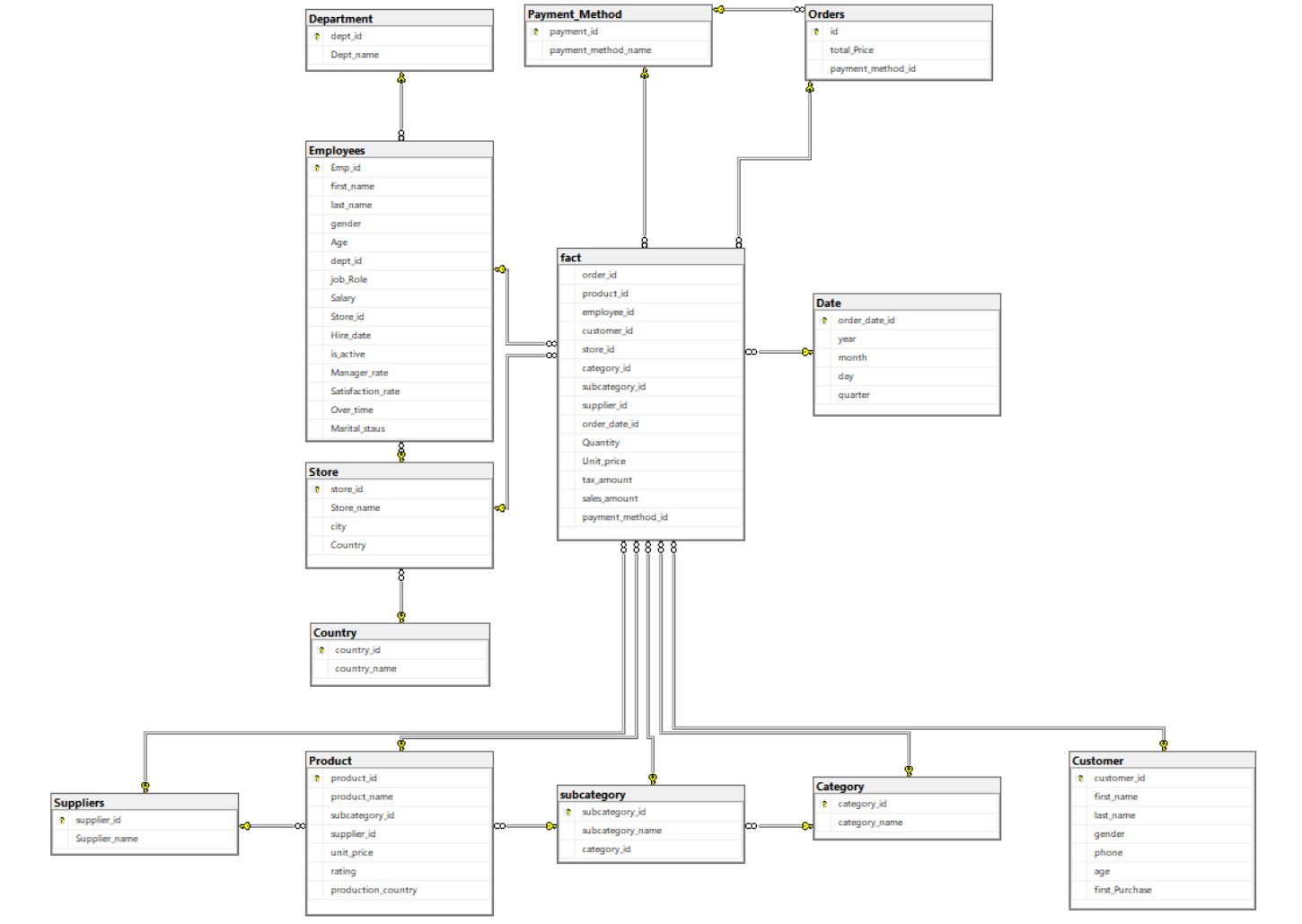
- Track customer purchase history and preferences.
- Resolve customer issues efficiently.

5. Business Intelligence and Analytics:

- The Marhba database provides a robust platform for generating reports, analyzing trends, and extracting valuable insights.
- Identify areas for improvement and make data-driven decisions.
- By leveraging business intelligence and analytics, hypermarkets can optimize their operations, maximize their marketing efforts, and achieve sustainable growth in a competitive marketplace.

2. **Entity Relationship Diagram:**

The ERD for Marhba database illustrates a comprehensive data model that captures the relationships between various entities involved in hypermarket operations.



3. **Data Dictionary:**

- Table: Country

Column Name	Data Type	Description
country_id	Integer	Unique identifier for the country
country_name	String	Name of the country

- Table: Store

Column Name	Data Type	Description
store_id	Integer	Unique identifier for the store
store_name	String	Name of the store
City	String	Name of the city
country_id	Integer	Foreign key to the Country table

- Table: Payment_Method

Column Name	Data Type	Description
payment_method_id	Integer	Unique identifier for the payment method
payment_method_name	String	Name of the payment method

- Table: Employees

Column Name	Data Type	Description
employee_id	Integer	Unique identifier for the employee
first_name	String	First name of the employee
last_name	String	Last name of the employee
gender	String	Gender of the employee
age	integer	Age of the employee
dept_id	Integer	Foreign key to the Department table
job_role	String	The job role of the employee
salary	String	Salary of the employee
store_id	Integer	Foreign key to the Store table

hire_date	Date	The hire date of the employee
is_active	String	Employment status (Is employee currently working or not)
manager_rate	String	Manager's evaluation of the employee
satisfaction_rate	String	Employee satisfaction rate
over_time	String	Is the employee working overtime or not
marital_status	String	Marital status of the employee

- Table: Department

Column Name	Data Type	Description
dept_id	Integer	Unique identifier for the department
dept_name	String	Name of the department

- Table: Fact

Column Name	Data Type	Description
order_id	Integer	Key to the orders
product_id	Integer	Foreign key to the Product table
employee_id	Integer	Foreign key to the Employees table
customer_id	Integer	Foreign key to the Customer table
store_id	Integer	Foreign key to the Store table
category_id	Integer	Foreign key to the category table
subcategory_id	Integer	Foreign key to the subcategory table
supplier_id	Integer	Foreign key to the suppliers table
order_date_id	Integer	Foreign key to the Date table
quantity	Integer	Quantity of the product purchased
unit_price	Integer	Price per unit of the product
tax_amount	float	Product tax amount
sales_amount	Integer	Product sales amount
payment_method_id	Integer	Foreign key to the payment method table

- Table: Date

Column Name	Data Type	Description
order_date_id	Integer	Unique identifier for the date
year	Integer	Year of the date
month	Integer	Month of the date
day	Integer	Day of the date

- Table: Product

Column Name	Data Type	Description
product_id	Integer	Unique identifier for the product
product_name	String	Name of the product
subcategory_id	Integer	Foreign key to the subcategory table
supplier_id	Integer	Foreign key to the Suppliers table
unit_price	String	Price per unit of the product
rating	String	rating assigned to product
production_country	String	The country producing the product

- Table: Suppliers

Column Name	Data Type	Description
supplier_id	Integer	Unique identifier for the supplier
supplier_name	String	Name of the supplier

- Table: Category

Column Name	Data Type	Description
category_id	Integer	Unique identifier for the category
category_name	String	Name of the category

- Table: Subcategory

Column Name	Data Type	Description
subcategory_id	Integer	Unique identifier for the subcategory
subcategory_name	String	Name of the subcategory
category_id	Integer	Foreign key to the Category table

- Table: Customer

Column Name	Data Type	Description
customer_id	Integer	Unique identifier for the customer
first_name	String	First name of the customer
last_name	String	Last name of the customer
gender	String	Gender of the customer
age	integer	Age of the customer
phone	String	Customer phone number
first_Purchase	String	Customer first purchase year

- Table: Orders

Column Name	Data Type	Description
id	Integer	Unique identifier for the order
total_price	Integer	Order total price
payment_method_id	Integer	Foreign key to the payment method table

4. **Constraints:**

- Primary Keys:
  - Country Table: country_id
  - Store Table: store_id
  - Payment_method Table: payment_method_id
  - Employees Table: employee_id
  - Date Table: date_id
  - Department Table: dept_id
  - Product Table: product_id
  - Suppliers Table: supplier_id
  - Category Table: category_id
  - Subcategory Table: subcategory_id
  - Customer Table: customer_id
  - Orders Table: id
- Foreign Keys:
  - supplier_id references Suppliers (supplier_id)
  - product_id references Product (product_id)
  - customer_id references Customer (customer_id)
  - employee_id references Employees (employee_id)
  - order_date_id references Date (order_date_id)
  - payment_method_id references Payment_method (payment_method_id)
  - store_id references Store (store_id)
  - category_id references Category (category_id)
  - subcategory_id references Subcategory (subcategory_id)
  - order_id references Orders (id)

5. **Relationships:**

- country table is related to store table via the column (country_id)
- store table is related to employees table via the column (store_id)
- store table is related to fact table via the column (store_id)
- payment_method table is related to fact table via the column (payment_id)
- date table is related to fact table via the column (order_date_id)
- employee table is related to fact table via the column (emp_id)
- employee table is related to department table via the column (dept_id)
- suppliers table is related to fact table via the column (supplier_id)
- product table is related to fact table via the column (product_id)
- subcategory table is related to fact table via the column (subcategory_id)
- category table is related to fact table via the column (category_id)
- customer table is related to fact table via the column (customer_id)
- suppliers table is related to product table via the column (supplier_id)
- product table is related to subcategory table via the column (subcategory_id)
- subcategory table is related to category table via the column (category_id)
- payment_method table is related to orders table via the column (payment_id)
- orders table is related to fact table via the column (order_id)

6. Stored Procedures:

1. **Calculate Total Price**

   Description: calculate the total price of a product based on its name and the quantity ordered.
   Input Parameters:
   - @Product_Name: The name of the product the total price is calculated.
   - @Quantity: The quantity of the product ordered.
   Output:
   - Category: The category of the product.
   - Subcategory: The subcategory of the product.
   - Product Name: The name of the product.
   - Quantity: The quantity of the product.
   - Unit Price: The unit price of the product.
   - Total Price: The calculated total price of the ordered quantity.

   Execution:

```
Exec [CalculateTotalPrice] @product_name ='pants', @quantity = 4;
```

2. **Get Subcategories**

   Description: retrieve a list of subcategories based on the specified category name.
   Input Parameters:
   - @category_name: The name of the category to retrieve subcategories.
   Output:
   - Subcategory: The names of the subcategories that belong to the specified category.
   Execution:

```
Exec GetSubcategories @category_name = 'Electronics'
```

3. **Get Products By Subcategory**

   Description: retrieve a list of products belonging to a specified subcategory.
   Input Parameters:
   - @subcategory_name: The name of the subcategory to retrieve products.
   Output:
   - Product Name: The name of the product.
   - Unit Price: The unit price of the product.
   - Rate: The rating of the product.
   Execution:

```
Exec GetProductsBySubcategory @subcategory_name = 'smartphone'
```

4. **City Top Product**

   Description: retrieve the top-selling products in a specific city for a given store. It list the top 5 products based on the number of items sold.
   Input Parameters:
   - @id (INT): The store_id to retrieve the top-selling products in a specific city.
   Output:
   - City: The name of the city where the store is located.
   - Category: The name of the product category.
   - Subcategory: The name of the product subcategory.
   - Product: The name of the product.
   - Unit Price: The unit price of the product.
   - Number of Items Sold: The total number of items sold for each product.
   Execution:

```
Exec CityTopProduct @id = 5;
```

5. **Get Sales**

   Description: sales-related information for a specific employee based on the provided id.

   Input Parameters:

   - @Id: The unique identifier for the employee to retrieve sales-related information.

   Output:

   - ID: The employee's id.
   - Full Name: The full name of the employee.
   - Gender: The gender of the employee.
   - Age: The age of the employee.
   - Hire Date: The date when the employee was hired.
   - Store: The name of the store where the employee works.
   - Country: The country where the store is located.
   - Satisfaction Rate: The satisfaction rate of the employee.
   - Salary: The employee's salary.
   - Total Sales: The total sales amount attributed to the employee.

   Execution:

   ```
   Exec GetSales @Id = 500;
   ```

6. **Sales By Year**

   Description: retrieve total sales for a specific year and category.

   Input Parameters:

   - @year: The year to retrieve sales data.
   - @category: The category identifier to filter the sales data.

   Output:

   - Year: The year for which sales data is being retrieved.
   - Month: The month within the specified year.
   - Total Sales: The total sales amount for the given category and year, broken down by month.

   Execution: `Exec SalesByYear @year = 2019, @category = 1`

7. **Store Sales**

Description: retrieve total sales for stores in a specific country, identified by the country_id.

Input Parameters:

- @id: The unique identifier of the country to retrieve store sales data.

Output:

- Store: The name of the store.
- City: The city where the store is located.
- Total Sales: The total sales amount for each store in the specified country.

Execution:

```
Exec StoreSales @id = 3;
```

8. **Supplier info**

Description: retrieve information about products supplied by a specific supplier.

Input Parameters:

- @id: The unique identifier of the supplier to retrieve product info.

Output:

- Supplier: The name of the supplier.
- Category: The product category to which the product belongs.
- Subcategory: The product subcategory to which the product belongs.
- Product: The name of the product.
- Unit Price: The unit price of the product.

Execution:

```
Exec SupplierInfo @id = 11;
```

9. **Top Customer Sales**

Description: retrieve total purchase amount for the top 10 customers in a specific country.

Input Parameters:

@id: The unique identifier of the country to retrieve sales data.

Output:

- Full Name: The full name of the customer.
- Age: The age of the customer.
- Gender: The gender of the customer.
- City: The city where the store is located.
- Total Purchase Amount: The total purchase amount made by each customer in the specified country.

Execution:

```
Exec TopCustomerSales @id = 2;
```

The other stored procedures were for select, insert, update, and delete operations on all database tables.

Example:

```
Create proc selectStore @id int
As begin
  Select * from Store
  Where store_id=@id
  End
GO

Exec selectStore @id = 1;
```

7. Backup info:

- **Backup Date and Time:**
  - 10/21/2023 4:54 AM
- **Database Information:**
  - Database Name: Project (Marhba)
  - Database Size: 9.5 MB
- **Backup Type:**
  - Full Backup