**Topic: Retail Store Management System**

**Project Team:** 4

**Team Members:**

Diksha Vinodkumar Desai

Sai Suresh Kale

Shiva Mahitha Maddi

Sayali Sambhaji Pathare

Abhilash Sunil Wase

**Database Purpose:**

The purpose of the Retail Database Management System is to amplify the efficiency, sales, and quality of the Retail Store services. It will simplify the process of maintaining data related to the inventory trends, orders, customers, employee management, and finances. This system will not only help to overcome the manual errors but also keeps the store organized for better customer service.

**Business Problems Addressed:**

* Allow retail store administration to check in-store stock availability
* To improve customer experience by providing an ability to check details of the customers which helps to serve the customers better (as this data gives an awareness of the tastes and preferences of the customers)
* Maintains history of the orders so that the store administrators can anticipate the inventory needs of the goods supplied by the store
* Helps to maintain Customer Acquisition and Retention by managing asset maintenance and repair
* Provides metrics about employee performance management to ensure business targets are met
* Generates order reports to know which products are driving revenue so that sales supervisors can plan their stock orders accordingly

**Business Rules:**

* Each store can have one or more employees.
* Each order can have one or more products.
* Each product category can have one or more products.
* Each employee will have one timesheet
* Each supplier can have one or more products
* Each invoice has one order
* Each delivery has one order

**Design Requirements:**

* Use Crow’s Foot Notation
* Mention Primary Key fields for each table by specifying PK next to that field
* Show relationship among the tables by pointing to the fields involved in the relationship
* Mention the appropriate relationship signs for the tables. E.g.: crow’s foot for many sides of the relationship and one for the one side relationship
* Maintain Business rules if the relationship is non-identifying

**DESIGN DECISIONS:**

|  |  |  |
| --- | --- | --- |
| **Entity Name** | **Why Entity is Included** | **How Entity is Related to Another** |
| Store | The application is designed to maintain information about the store and serves as one of the important entities of the database. | The entity is linked to several other entities like invoices, assets, promotions through its primary key so that we can particularly improve customer experience and attract more customers. For many-to-many relationship we create associative entities. |
| Customer | A store wont exists if there are no customers, hence maintaining customer information is necessary for the functioning of store. With the help of customer information, we can provide them better service and can also offer reward points, promotions, and memberships. Store managers can track the numbers of customers, and the top customers. Also, based on the customer feedback store performance can be improved. | Each customer places zero or more orders over a period. The order is further linked to invoice. Each order has one or more product with the help of which we can track the top products of a customer. |
| Employee | Another important entity of a store are its employees. Maintaining employee information is necessary from store management point of view as they can gain insights about the employee’s performance and can track employee’s record. | Every employee has a timesheet through which we can track the number of hours he/she worked.  The performance of employees can be tracked through 5 performance indicators mentioned in EmployeePerformanceIndicator table where the employee id serves as foreign key.  Employee is also related to delivery entity as there can be zero or more employees who would assist in delivering orders. Every employee will also have credentials to login into the system. |
| Product | The product entity maintains product information like its availability, price, supplier details, product category, and the rack number for quick access. Through the product information we can track the sales of products (e.g.: bestselling products), when the stocks need to be updated. | One or more products can be part of one order. We can track/obtain information of the products and its quantities present in an order through it. The product entity is linked to supplier entity through which we can track the supplier details for a product. There is also a many-to-many relationship between product and promotions and hence we have created an associative entity |
| Order | The Order entity is used to track the orders of customer, which includes product information, quantity, order type, date of order and order-item status. Through order details we can track in-store/online orders record. Also, we can track employee information who processed the order. | One order will be consisting of one or many products. We can search for particular order or individual order item through orders table. The order entity is linked with Delivery and Invoice table which would help to track payment information of customer and order delivery status. |
| Supplier | The Supplier entity is used to maintain the information of the supplier and their products. | Supplier entity is linked with the Product entity which would help to track the products availability and stock information. |
| Promotions | The promotions entity is to increase the sale of the store by offering multiple discounts to attract more customers. | Promotion entity is linked with Store and Products through a many to many relationships which is managed by creating associated entities. We can track promotion information of individual products as well as its store. |

**ERD:**

**Diagram, engineering drawing, schematic

Description automatically generated**