**Group members:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Group No** | **Student ID** | **Name** | **Project Topic** |
| 7 | **181805051** | **Şevval Şimşek** | **Supermarket Stock Control System** |
| **181805067** | **Buse Latife Beker** |
| **181805077** | **Emine Ece Coşkunçay** |

**1:** What is the purpose of your project? Explain.

The Supermarket stock control database management system, it allows the user (or the owner / Authorized Person) to see the product stocks at any time and to follow up in detail.

The basis of supermarket stock control is keeping a record of stock that comes into the store and stock that leaves the store. Every time an item is sold, the information will be sent to a computer and the sale will be deducted from the stock control table. This system should be able to tell you stock status there is of any one product.

It can also show information like identify products that have expired, the number of stocks, products that have decreased stock, and product details, personnel information, orders received from customers, customer information in a detailed is recorded.

**2:** Draw your E-R diagram (Specifiy all entities-relationships-attributes). Send it as a separate file (.pdf, jpg..). Show (explain in detail) that the database is at least in 3rd NF.

* More than one information cannot be in a single column.
* Information within a field should not be separated by special characters.
* Each non-key column defined in a table must depend on key columns defined as primary keys. it must contain the information needed by the key column.
* If the key column consists of a combination of more than one column, the data in the table must be linked to both columns. If dependent on a single column, it must be kept in a separate table.
* The column that is not a key in a table must be related to the key column of another table or the columns of the table in which it is located, or ​the column that is not a key for the table cannot be dependent on any other column that is not a key.

Using these rules, we drew our diagram and database.

Since the address attribute in Customer and Staff attributes is a very valuable attribute, we have defined a new entity called Address.

The categories entity holds the categories created for the products in the supermarket.

There is no dependency between the **user\_password** and **staffID**. That's why we made a separate table for the user.

There is a dependency between the **amount** and **price** attributes but any of them do not follow the 3NF rule as they are not keys. It complies with the 3NF rule in the new table (table of Order Detail) created. Likewise, we did it for **stock\_status** and **numberOf\_stock**.

There is an **n-m** relationship between **Category** and **Product** entities. Since n-m relationships require a new entity, the Product Category (Has) relationship has been defined and this relationship will be used when converting the ER model to a database table. For example,

|  |  |
| --- | --- |
| Product Category Table | |
| **PF** | **productID** |
| **PF** | **categoryID** |

With 3NF, we divided the tables in the most efficient way. In this way, unnecessary information repetitions in the database will be prevented, while the amount of data in the database goes to infinity, the data will be kept more efficiently.

**3:** Explain your tables and attributes of tables. Explain your data types of attributes of tables. Why you choose that type?

Address Table: This is a table containing address information of staff and customers.

Contains **addressID, street, neighborhood, building\_no, city, post\_code, country** attributes.

We used an **integer** type for the **addressID** and **post\_code** attributes because it can hold integers and the data range is wide. When id is mentioned, numbers such as identity number, school number come to mind first and every data in this field must be unique to avoid confusion.

We used the **varchar** type for **all other attributes** because the address information contains both numbers and text and also varchar data type is better suited for these fields because it takes up as much space as the entered data.

Staff Table: This is the table that contains staff information.

Contains the attributes of **staffID, staff\_name, staff\_surname, staff\_phone, staff\_salary, addressID(FK).**

We used the **staffID** and **AddressID** attribute as **integer** types because it can hold integers and the data range is wide. Like adressID in the Address table.

We used an **money** type for the attribute **staff\_salary** because it is a monetary value and money is used to hold monetary values ​​of about -2⁶⁴ to 2⁶⁴. It stores data of decimal type sensitive up to 4 digits..

We used **varchar** for **staff\_name** and **staff\_surname** attributes because varchar is used for fields that hold alpha numeric data and it takes up as much space as the entered data.

Attributes of **staff\_phone** is a **varchar** type because it may contain phone numbers of different countries. So varchar is used for fields that hold alpha numeric data and it takes up as much space as the entered data

User Table: This table contains information for logging into the system.

Contains the attiributes **userID, user\_password, staff\_name(FK).**

We used **integer** type for other **userID** because it can hold integers and the data range is wide.

We used **varchar** for **staff\_name** attributes because varchar is used for fields that hold alpha numeric data and it takes up as much space as the entered data.

We used **varchar** for **user\_password** because the password can contain any number, letter sign.

Customer Table: This is the table containing customer information.

Contains the attiributes **CustomerID, customer\_name, customer\_surname, customer\_phone, addressID(FK)**.

We used **integer** type for other **customerID** and **addressID** because it can hold integers and the data range is wide.

We used a **varchar** type for **other attributes** because varchar is used for fields that hold alpha numeric data and it takes up as much space as the entered data.

Attributes of **customer\_phone** is a **varchar** type because it may contain phone numbers of different countries. So varchar is used for fields that hold alpha numeric data and it takes up as much space as the entered data.

Order Table: This is the table containing the **orderID, customerID(FK), order\_date** attributes for orders placed by customers.

We used **integer** type for **orderID** and **customerID** because it can hold integers and the data range is wide.

We used an **date** type for **order\_date** because date is the data type that can store dates Thanks to the many static methods and properties of its structure, basic practical operations related to date applications can be performed.

Order Detail Table: This is the table containing the order details.

Contains the attiributes **orderdetailID, orderID(FK), productID(FK), amount, price** .

We used **integer** type for **orderID**, **orderdetailID**, **productID**, **amount** because it can hold integers and the data range is wide.

we used a **money** type for **price** because it is used to hold monetary values ​​of about -2⁶⁴ to 2⁶⁴. It stores data of decimal type sensitive up to 4 digits.

Product Table: This is the table containing product information.

It contains the attiributes **productID, product\_name, production\_date, consumption\_date, purchase\_price, sale\_price, brandID(FK), stockID(FK)**.

We used an **integer** type for **productID, brandID, stockID** because it can hold integers and the data range is wide.

We used a **varchar** type for **product\_name** because varchar is used for fields that hold alpha numeric data and it takes up as much space as the entered data.

We used an **date** type for **production\_date** ve **consumption\_date** because date is the data type that can store dates Thanks to the many static methods and properties of its structure, basic practical operations related to date applications can be performed.

We used **money** type for the attributes **purchase\_price** and **sale\_price**(so we can see the difference / profit between them.) because it is used to hold monetary values ​​of about -2⁶⁴ to 2⁶⁴. It stores data of decimal type sensitive up to 4 digits.

Brand Table: This is a table containing all brand names.

Contains the attributes **brandID** **ve brand\_name**.

We used **integer** type for **brandID** because it can hold integers and the data range is wide.

We used a **varchar** type for **brand\_name** because varchar is used for fields that hold alpha numeric data and it takes up as much space as the entered data.

Stock Table: This is the table containing the stock information of the products.

Contains the attributes **stockID**,**stock\_status, numberOf\_stock** .

We used **integer** type for **stockID** ve **numberOf\_stock** because it can hold integers and the data range is wide.

We used a **varchar** type for **stock\_status** because varchar is used for fields that hold alpha numeric data and it takes up as much space as the entered data.

Category Table: This is the table containing the names of the categories.

We used an **integer** type for **categoryID** because it can hold integers and the data range is wide.

We used a **varchar** type for **category\_name** because varchar is used for fields that hold alpha numeric data and it takes up as much space as the entered data.

**Since it is obvious why they took those names, we did not explain what the attributes are used for.**

**4:** Define your primary and foreign keys. Why they are keys? Explain.

**Primary keys:** **userID**(table of User), **staffID**(table of Staff), **addressID**(table of Address) , **customerID**(table of Customer), **orderID**(table of Order), **orderDetailedID**(table of Order Detail), **productID**(table of Product), **brandID**(table of Brand), **stockID**(table of Stock), , **categoryID**(table of Category)

They are primary keys because in order not to confuse the values ​​of the rows, the information of this field should not be repeated.

**Foreign keys:** **adressID**(table of Staff and Customer), **staff\_name** (table of User), **customerID**(table of Order), **orderID**(table of Order Detail), **productID**(table of Order Detail), **brandID**(table of Product), **stockID**(table of Product), **categoryID**

They are foreign keys because possible data duplication can be prevented and the relevant data can be updated everywhere in future updates. In summary, to associate values ​​that can be entered in one data table with fields in another data table.