#### EXPLANATION OF THE SYSTEM

##### Brief Information About the System and Process

Our company was established to act as an intermediary on e-commerce sites. First, the user creates his/her account by registering at login and typing his/her name, surname and address. The customer then uses the application to specify in which category he wants to search for products, each of the categories of the products has its own id value, and when this id is entered, the products are ranked according to the rank grades scored out of 5. Then, the customer will see the price of these products, for example on the amazon canada site, as well as the prices and stock status of the amazon turkey site, accordingly, the product will be sold to amazon turkey, for example, the product bought from the amazon canada site, keeping the profit margin at the maximum.

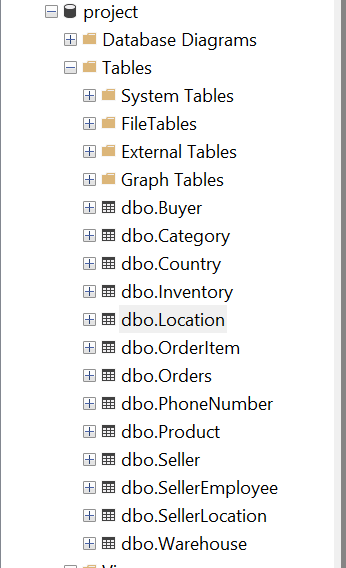
Expected Benefits of the Software

It is designed to give a general description of the wide range of products on e-commerce sites. The database will help the company learn more about the customer's wishes and the company sales rate. From the data collected using our company catalog support, it will be easier to identify the main best-selling products among our customers and to access the price difference between the countries.

|  |  |  |  |
| --- | --- | --- | --- |
| ***DATABASE DESIGN DOCUMENT*** | | | |
| ***TABLE NAME*** | ***DATA FIELD*** | ***DATA TYPE*** | ***DEFINITION*** |
| ***PRODUCT*** | *Product ID (PK)* | *Auto number* | *Represents a unique ID number of each product.* |
| *List Price* | *Number* | *Shows the list price of the products.* |
| *Description* | *Long Text* | *Provides explanations about the product.* |
| *Rank* | *Number* | *Shows the ranking of the product in general sales after the products are scored by the users* |
| *Rating Count* | *Number* | *Shows the total rating by users* |
| *Category ID* | *Auto number* | *Each product has a category to which it belongs, and these categories have different ID numbers.* |
| ***CATEGORY*** | *Category ID (PK)* | *Auto number* | *Category ID is used to distinguish each category from each other.* |
| *Category Name* | *Short text* | *Category Name is used to search for categories* |
| *Category URL* | *Short text* | *Category Url is used to display the category web page on the site where the product is located.* |
| ***SELLER*** | *Company ID(PK)* | *Auto number* | *It is a different and distinctive number of each company selling the product.* |
| *Company Name* | *Short text* | *The name of the company selling the product that appears to the customers* |
| *Badge Number* | *Auto number* | *Represents the company's badge number* |
|  | *Inventory ID(PK)* | *Auto number* | *The special id number of the inventory with the products.* |
|  | *Production ID* | *Auto number* | *Different id number for each product.* |
| *INVENTORY* | *Warehouse ID* | *Auto number* | *Special ID numbers that identify the warehouse where the products in the inventory are stored.* |
|  | *Quantity Hand* | *Number* | *A value manually entered by the seller, showing the remaining amount of the item in the inventory.* |
| ***Order Item*** | *Order ID (PK)* | *Auto Number* | *It is the value that is different for each order used to distinguish orders from each other.* |
| *Product ID* | *Auto Number* | *Different id number for each product.* |
| *Unit Price* | *Number* | *It is the value that shows the unit price of the product.* |
| *Quantitiy* | *Number* | *Data showing the order quantity.* |
| ***ORDERS*** | *Order ID (PK)* | *Auto number* | *This value is used to distinguish each order from each other..* |
| *Buyer ID* | *Short text* | *Displays a unique value for the purchasing user* |
| *Order Date* | *Auto number* | *Indicates the order date* |
| *Order Code* | *Auto number* | *It is the order tracking number given so that the buyer can track the order.* |
| *Order Total* | *Number* | *Shows the total amount of this order.* |
| ***WAREHOUSE*** | *Warehouse ID(PK)* | *Auto Number* | *They are numbers used to distinguish warehouses from each other.* |
| *Warehouse Name* | *Short Text* | *Names of Warehouses so that users can find out their location on their website.* |
| *Location ID* | *Auto number* | *Values that specify the location of the warehouse and are used to distinguish between warehouses that are the primary key for another entity.* |
| *Warehouse Phone* | *Number* | *It is the part that shows the code of the phone number of the warehouses.* |
| ***COUNTRY*** | *Country ID (PK)* | *Auto Number* | *Randomly assigned value to distinguish between countries* |
| *Country Name* | *Number* | *The short text to be displayed by the users of the country id values that are different from each other.* |
| *Country Phone Code* | *Number* | *The part where the phone codes of each country, which are different from each other, are stored.* |
| ***PHONE NUMBER*** | *Phone Number ID* | *Auto number* | *Different id numbers of phone numbers.* |
| *Country Code* | *Number* | *Country codes used in phone numbers, with a different value for each country* |
| *Phone Number* | *Number* | *It is the part where phone numbers are stored directly and seen by users.* |
| *Buyer ID* | *Auto Number* | *It is the part where phone numbers are stored directly and seen by users.* |
| *Eeeee*  ***LOCATION*** | *Location ID (PK)* | *Auto Number* | *The id value of the data, where the location information is kept, which shows the address information in more detail, each one different from the other.* |
| *Country ID* | *Auto Number* | *Randomly assigned value to distinguish between countries* |
| *Adress Line* | *Short Text* | *Detailed address writing* |
| *City* | *Short Text* | *The name of the city where the location is located.* |
| *State* | *Short Text* | *Short texts indicating the state* |
| *Post Code* | *Number* | *The zip code that will be used by the companies when shipping the products and is different for each location* |
| ***SELLER EMPLOYEE***  ***BUYER*** | *Seller Employee ID(PK)* | *Auto number* | *Each employee has his own number to identify the employee.* |
| *Company ID* | *Auto Number* | *The id number of the company where the employees work.* |
| *Employee First Name* | *Short Text* | *Employee First Name.* |
| *Employee Second Name* | *Short Text* | *Employee Second Name.* |
| *Buyer ID (PK)* | *Auto Number* | *ID numbers used to distinguish purchasers from each other* |
| *First Name* | *Short Text* | *Buyer first name.* |
| *Last Name* | *Short Text* | *Buyer last name* |
| *User Name* | *Short Text* | *Usernames used as nicknames in the profile created by users.* |
| *Gender* | *Short Text* | *The part with information about the city to which the order will be delivered.* |
| *Phone Number* | *Short Text* | *The gender that users specify in their account* |

#### ENTITY RELATIONSHIP DIAGRAM USING SQL SERVER

**DATA MODEL**

****

Below is the database of our company.

Each Product must be having one or more Inventory.

Each Category must be having one or more Product.

Each Product must be placed by one only one Category.

Each Order Item must be having one or more Inventory.

Each Orders must be having one or more Order Item.

Each Order Item must be placed by one only one Orders.

Each Inventory must be having one or more Warehouse.

Each Warehouse must be placed by one only one Location.

Each Location must be placed by one only one Country.

Each Phone Number must be placed by one only one Location.

Each Buyer must be having one or more Phone Number.

Each Buyer must be having by one only one Location.

Each Seller must be having one or more Seller Employee.

Each Seller Employee must be placed by one only one Seller.

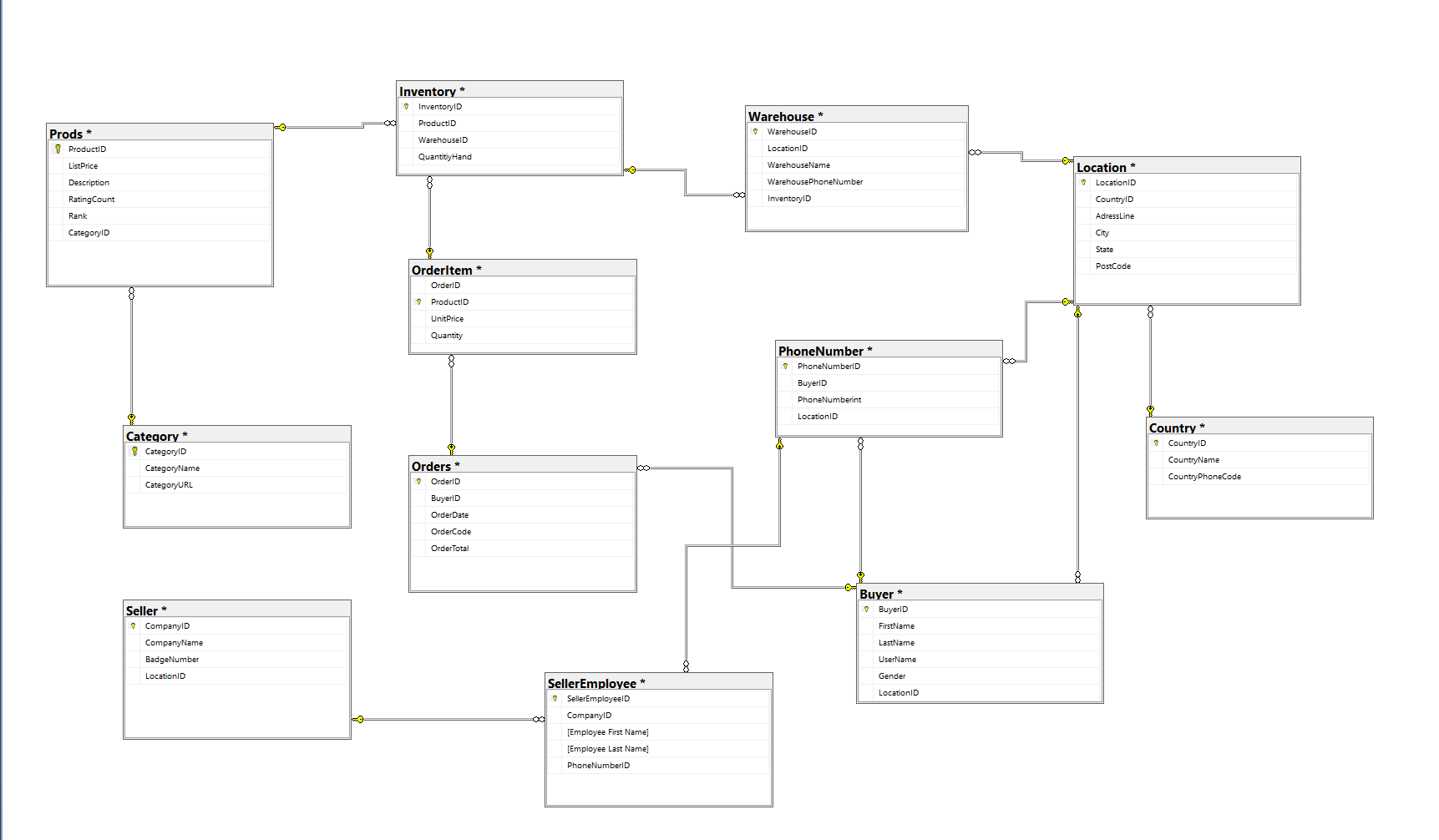
Each Seller Employee must be placed by one only one Phone Number.

Each Buyer must be having one or more Orders.

Each Orders must be having by one only one Buyer.

#### ENTITY RELATIONSHIP DIAGRAM USING

#### SQL SERVER



**SQL CODE GENERATION**

SQL SERVER CODE

**CREATE TABLE PRODUCT**

(Product ID INTEGER NOT NULL,

List Price INTEGER NOT NULL,

Description VARCHAR (150) NULL,

Rating Count INTEGER NULL,

Rank INTEGER NOT NULL,

Category ID INTEGER NOT NULL,

PRIMARY KEY (ProductID)).

**CREATE TABLE CATEGORY**

(Category ID INTEGER NOT NULL,

Category Name VARCHAR(50) NOT NULL,

Category URL VARCHAR (150) NOT NULL,

PRIMARY KEY (CategoryID)).

**CREATE TABLE SELLER**

(Company ID INTEGER NOT NULL,

Company Name VARCHAR(50) NOT NULL,

Badge Number INTEGER NULL,

Location ID INTEGER NOT NULL,

PRIMARY KEY (CompanyID)).

**CREATE TABLE INVENTORY**

(Inventory ID INTEGER NOT NULL,

Product ID INTEGER NOT NULL,

Warehouse ID INTEGER NOT NULL,

Quantity on Hand INTEGER NOT NULL,

PRIMARY KEY (CompanyID)).

**CREATE TABLE ORDER ITEM**

(Product ID INTEGER NOT NULL,

Order ID INTEGER NOT NULL,

Unit Price INTEGER NULL,

Quantity INTEGER NULL,

PRIMARY KEY (ProductID)).

**CREATE TABLE ORDERS**

(Order ID INTEGER NOT NULL,

Buyer ID INTEGER NOT NULL,

Order Date DATE NOT NULL,

Order Code INTEGER NOT NULL,

Order Total INTEGER NULL,

PRIMARY KEY (OrderID)).

**CREATE TABLE SELLER EMPLOYEE**

(Seller Employee ID INTEGER NOT NULL,

Company ID INTEGER NOT NULL,

Employee First Name VARCHAR(50) NOT NULL,

Employee Last Name VARCHAR(50) NOT NULL,

Phone Number ID INTEGER NOT NULL,

PRIMARY KEY (SellerEmployeeID)).

**CREATE TABLE WAREHOUSE**

(Warehouse ID INTEGER NOT NULL,

Location ID INTEGER NOT NULL,

Warehouse Name VARCHAR(50) NOT NULL,

Warehouse Phone Number INTEGER NOT NULL,

InventoryID INTEGER NOT NULL,

PRIMARY KEY (WarehouseID)).

**CREATE TABLE LOCATION**

(Location ID INTEGER NOT NULL,

Country ID INTEGER NOT NULL,

Adress Line VARCHAR(250) NOT NULL,

City VARCHAR(50) NOT NULL,

State VARCHAR(50)NOT NULL,

Post Code INTEGER NOT NULL,

PRIMARY KEY (Location ID )).

**CREATE TABLE COUNTRY**

(Country ID INTEGER NOT NULL,

Country Name VARCHAR(50) NOT NULL,

Country Phone Code INTEGER NOT NULL,

PRIMARY KEY (Country ID )).

**CREATE TABLE PHONE NUMBER**

(Phone Number ID INTEGER NOT NULL,

Buyer ID INTEGER NOT NULL,

Phone Number INTEGER NOT NULL,

Location ID INTEGER NOT NULL,

PRIMARY KEY (PhoneNumberID )).

**CREATE TABLE SELLER BUYER**

(Buyer ID INTEGER NOT NULL,

Gender VARCHAR(25) NULL,

First Name VARCHAR(50) NOT NULL,

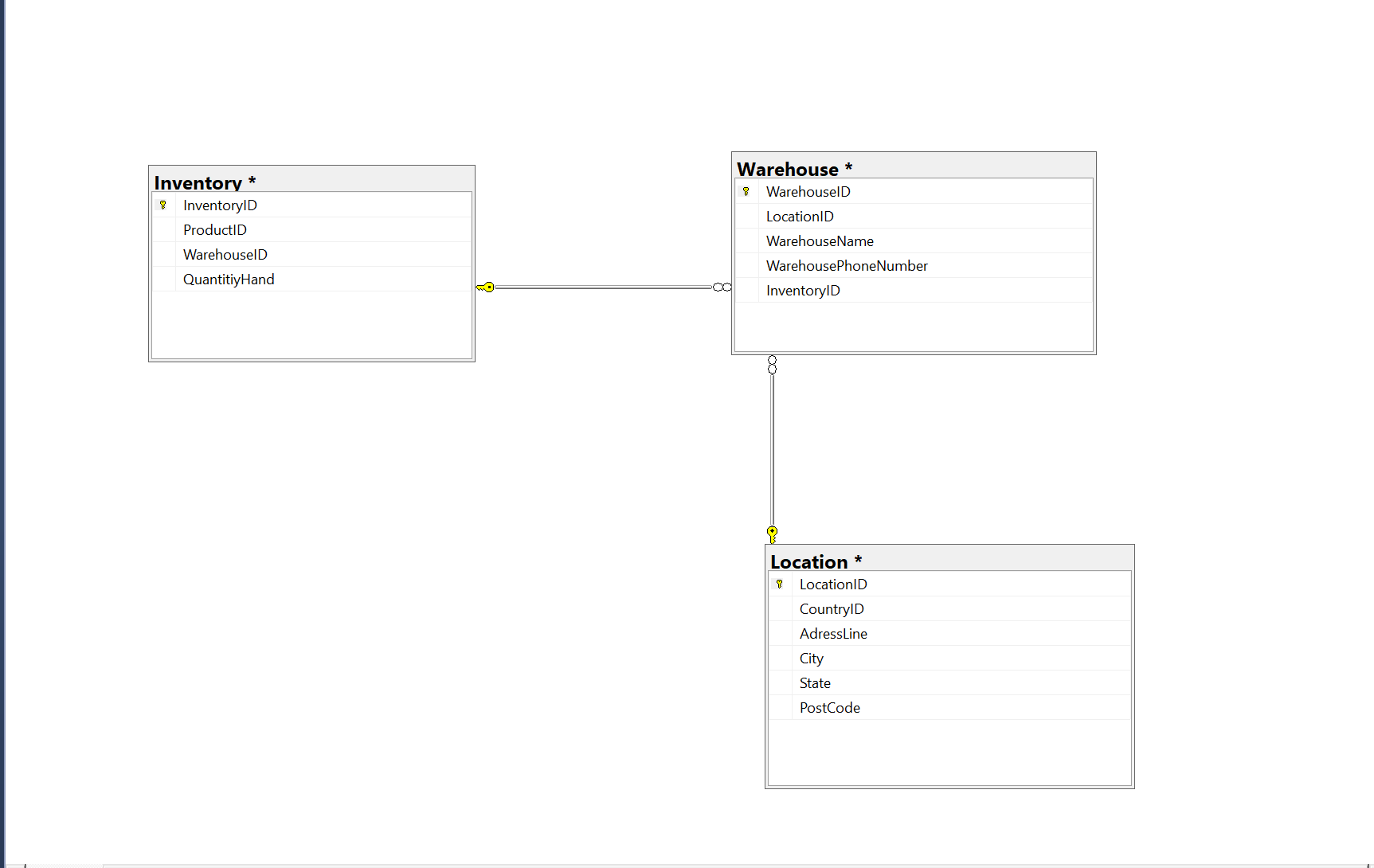
Last Name VARCHAR(50) NOT NULL,

Location ID INTEGER NOT NULL,

PRIMARY KEY (BuyerID)).

**SELECTION of THREE TABLES**

**and RELATIONS**



**CREATE TABLE WAREHOUSE**

(Warehouse ID INTEGER NOT NULL,

Location ID INTEGER NOT NULL,

Warehouse Name VARCHAR(50) NOT NULL,

Warehouse Phone Number INTEGER NOT NULL,

InventoryID INTEGER NOT NULL,

PRIMARY KEY (WarehouseID)).

**CREATE TABLE LOCATION**

(Location ID INTEGER NOT NULL,

Country ID INTEGER NOT NULL,

Adress Line VARCHAR(250) NOT NULL,

City VARCHAR(50) NOT NULL,

State VARCHAR(50)NOT NULL,

Post Code INTEGER NOT NULL,

PRIMARY KEY (Location ID )).

**CREATE TABLE INVENTORY**

(Inventory ID INTEGER NOT NULL,

Product ID INTEGER NOT NULL,

Warehouse ID INTEGER NOT NULL,

Quantity on Hand INTEGER NOT NULL,

PRIMARY KEY (CompanyID)).

**Relationships and Definitions**

**Product & Inventory**

Product and Inventory tables provide users with information about the products in the inventory. The Product ID is the foreign key in the inventory table.There is one to many (1:M) relation.

**Product & Category**

Product and category tables have a relation that describes to customer and executive which product is in which category. Accordingly, Category ID is a foreign key in Product Table. There is one to many (1:M) relation.

**Inventory & Warehouse**

It informs users about inventories and the warehouse where these inventories are kept.Accordingly, Inventory ID is a foreign key in WarehouseTable. There is one to many (1:M) relation.

**Warehouse & Location**

It gives information to managers and customers about their warehouses and their locations. The location ID is found in the Warehouse table as a foreign key. This relation is mandatory relation.

**Location & Country**

Shows the link between your location and the countries where these locations are located. The Country ID is found in the Location table as a foreign key. This relation is mandatory relation.

**Location & Phone Number**

It shows the connection between the phone number of the users and the location. It was created by considering the difference in the code of the phone numbers according to the location. Accordingly, Location ID is a foreign key in Phone Number table. There is one to many (1:M) relation.

**Phone Number & Buyer**

Shows the link between purchasers and their phone numbers.The Buyer ID is found in the Phone Number table as a foreign key. This relation is mandatory relation.

**Location & Buyer**

It shows the link of the buyers and the locations of these users to the administrators. Accordingly, Location ID is a foreign key in Buyer table. There is one to many (1:M) relation.

**Seller & Seller Employee**

Shows the link with employee within Seller companies.. Accordingly, Company ID is a foreign key in Seller Employee table. There is one to many (1:M) relation.

**Seller Employee & Phone Number**

Shows the link between sellers and their phone numbers.The Phone Number ID is found in the Seller Employee table as a foreign key. This relation is mandatory relation.

**Buyer & Orders**

It is the table that shows the manager and users the link between the orders and the buyers of those orders. Accordingly, Buyer ID is a foreign key in Orders Table. There is one to many (1:M) relation.

#### DDL STATEMENTS:

metin içeren bir resim

Açıklama otomatik olarak oluşturuldu

#### DML STATEMENTS:

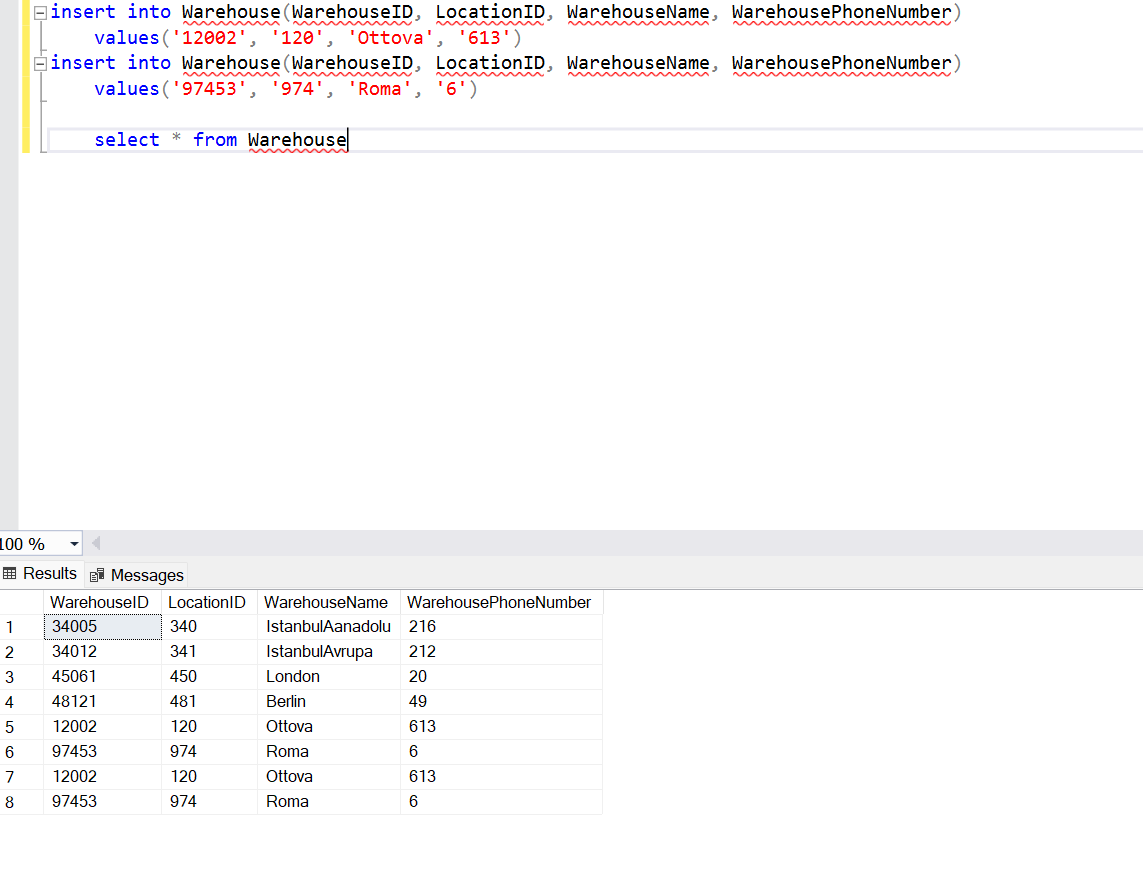
#### SELECT STATEMENT:

tablo içeren bir resim

Açıklama otomatik olarak oluşturuldu

#### 

#### INSERT STATEMENT:



#### UPDATE STATEMENT:

#### DELETE STATEMENT:

#### DQL STATEMENTS:

#### JOIN STATEMENT:

metin içeren bir resim

Açıklama otomatik olarak oluşturuldu

#### INNER JOIN STATEMENT: metin içeren bir resim Açıklama otomatik olarak oluşturuldu

#### LEFT JOIN STATEMENT:

#### RIGHT JOIN STATEMENT:

metin içeren bir resim

Açıklama otomatik olarak oluşturuldu

#### Group by and Order by statements:

#### DCL STATEMENTS:

**1-)COMMIT STATEMENT**

A COMMIT statement in SQL ends a transaction within a relational database management system (RDBMS) and makes all changes visible to other users. The general format is to issue a BEGIN WORK statement, one or more SQL statements, and then the COMMIT statement.

metin içeren bir resim

Açıklama otomatik olarak oluşturuldu

**2-) ROLLBACK STATEMENT:**

The ROLLBACK statement is the inverse of the COMMIT statement. It undoes some or all database changes made during the current transaction. For more information, see "Overview of Transaction Processing in PL/SQL

metin içeren bir resim

Açıklama otomatik olarak oluşturuldu

metin içeren bir resim

Açıklama otomatik olarak oluşturuldu

#### 3-)GRANT STATEMENT

Use the GRANT statement to give privileges to a specific user or role, or to all users , to perform actions on database objects.

#### 4-) REVOKE STATEMENT

The revoke statement revokes privileges. It removes database privileges or role access granted to the specified users, groups, roles, or PUBLIC.

#### metin içeren bir resim Açıklama otomatik olarak oluşturuldu