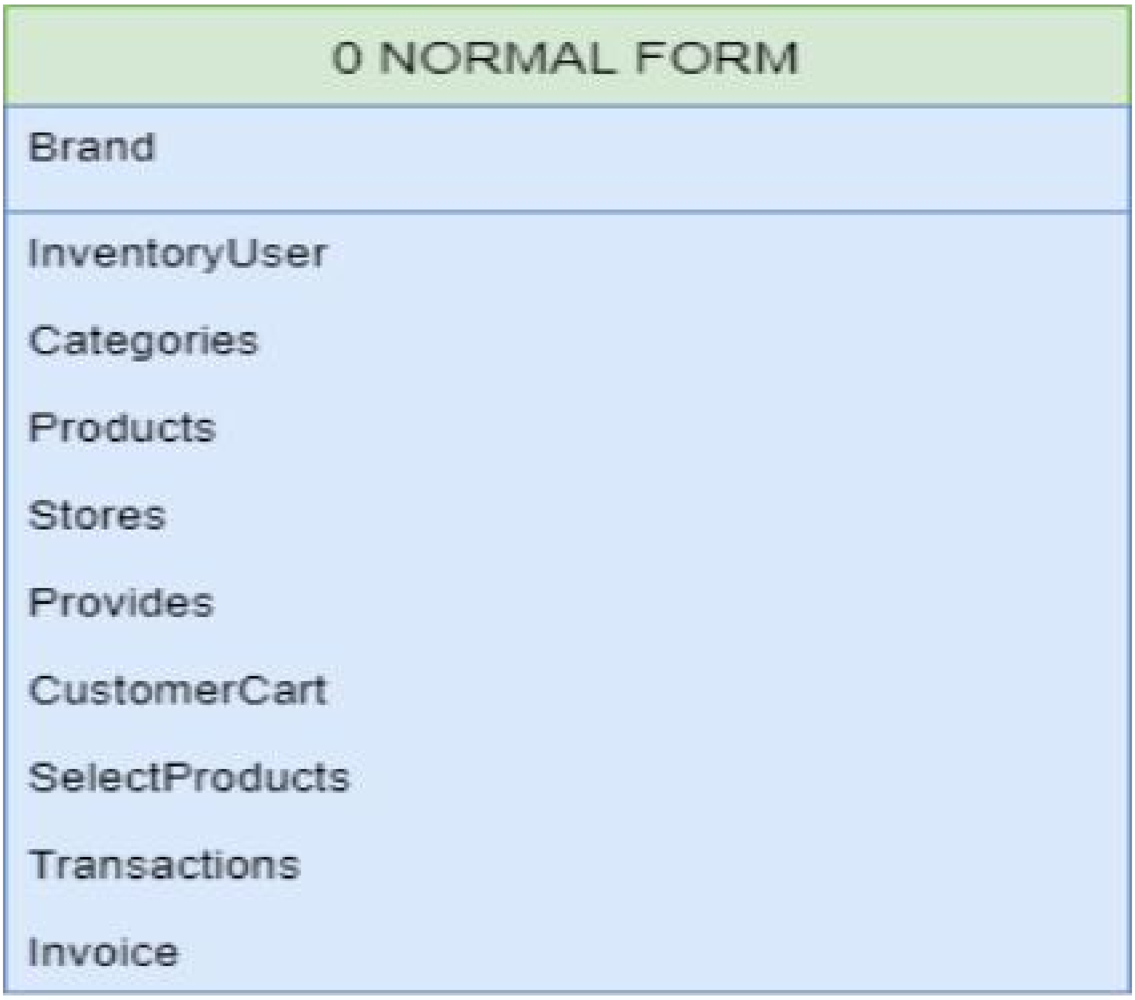
SUBJECT: DBD-281
ASSESSMENT:PROJECT
NAME: TRAVER MHERE

STUNDENT NUMBER 576809

a) Database Management Systems is the way to go in terms of keeping, monitoring, maintaining and retrieving of data efficiently. There is an organization called TravTon which runs several branches, and it sells different products. The company has been facing few problems in keeping track of their customers, and inventory, they have been using the old school of file system as their database which at times turns to be challenging.

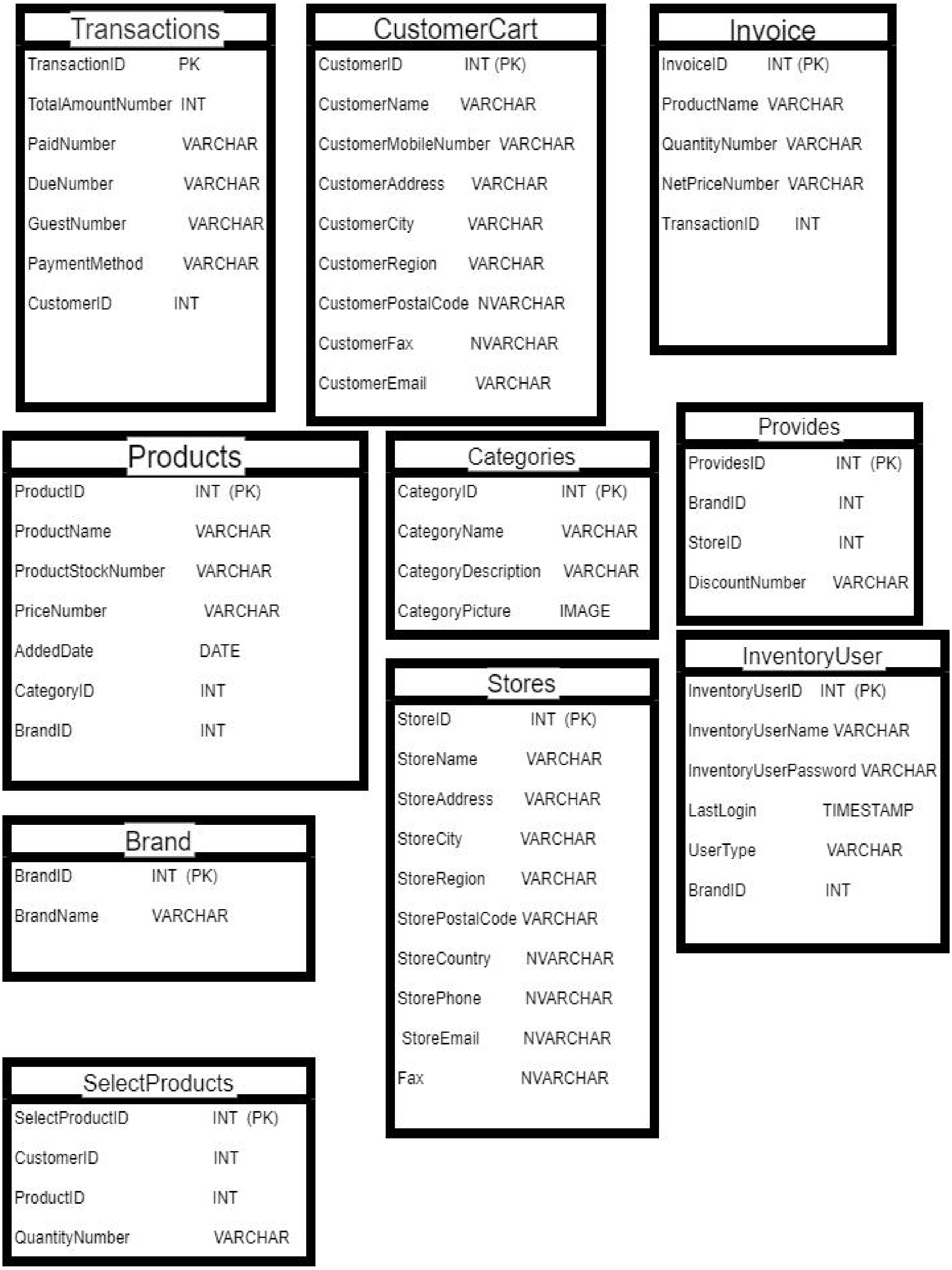
The database which l am creating is precisely based on few aspects such as inventory management, checking the quantity of the available stock in the inventory, creating invoices and creating a market with the customer information creation. All these can assist in maintaining proper levels of stock thus reducing overstocking, holding capital which could be used elsewhere.

b) ERD for the Inventory Database for TravTon Holdings Ltd



0 NORMAL FORM
Brand
InventoryUser
Categories
Products
Stores
Provides
CustomerCart
SelectProducts
Transactions
Invoice

1 NORMAL FORM



Transactions

TransactionID	PK
TotalAmountNumber	INT
PaidNumber	VARCHAR
DueNumber	VARCHAR
GuestNumber	VARCHAR
PaymentMethod	VARCHAR
CustomerID	INT

Products

ProductID	INT (PK)
ProductName	VARCHAR
ProductStockNumber	VARCHAR
PriceNumber	VARCHAR
AddedDate	DATE
CategoryID	INT
BrandID	INT

Brand

BrandID	INT (PK)
BrandName	VARCHAR

SelectProducts

SelectProductID	INT (PK)
CustomerID	INT
ProductID	INT
QuantityNumber	VARCHAR

CustomerCart

CustomerID	INT (PK)
CustomerName	VARCHAR
CustomerMobileNumber	VARCHAR
CustomerAddress	VARCHAR
CustomerCity	VARCHAR
CustomerRegion	VARCHAR
CustomerPostalCode	NVARCHAR
CustomerFax	NVARCHAR
CustomerEmail	VARCHAR

Categories

CategoryID	INT (PK)
CategoryName	VARCHAR
CategoryDescription	VARCHAR
CategoryPicture	IMAGE

Stores

StoreID	INT (PK)
StoreName	VARCHAR
StoreAddress	VARCHAR
StoreCity	VARCHAR
StoreRegion	VARCHAR
StorePostalCode	VARCHAR
StoreCountry	NVARCHAR
StorePhone	NVARCHAR
StoreEmail	NVARCHAR
Fax	NVARCHAR

Invoice

InvoiceID	INT (PK)
ProductName	VARCHAR
QuantityNumber	VARCHAR
NetPriceNumber	VARCHAR
TransactionID	INT

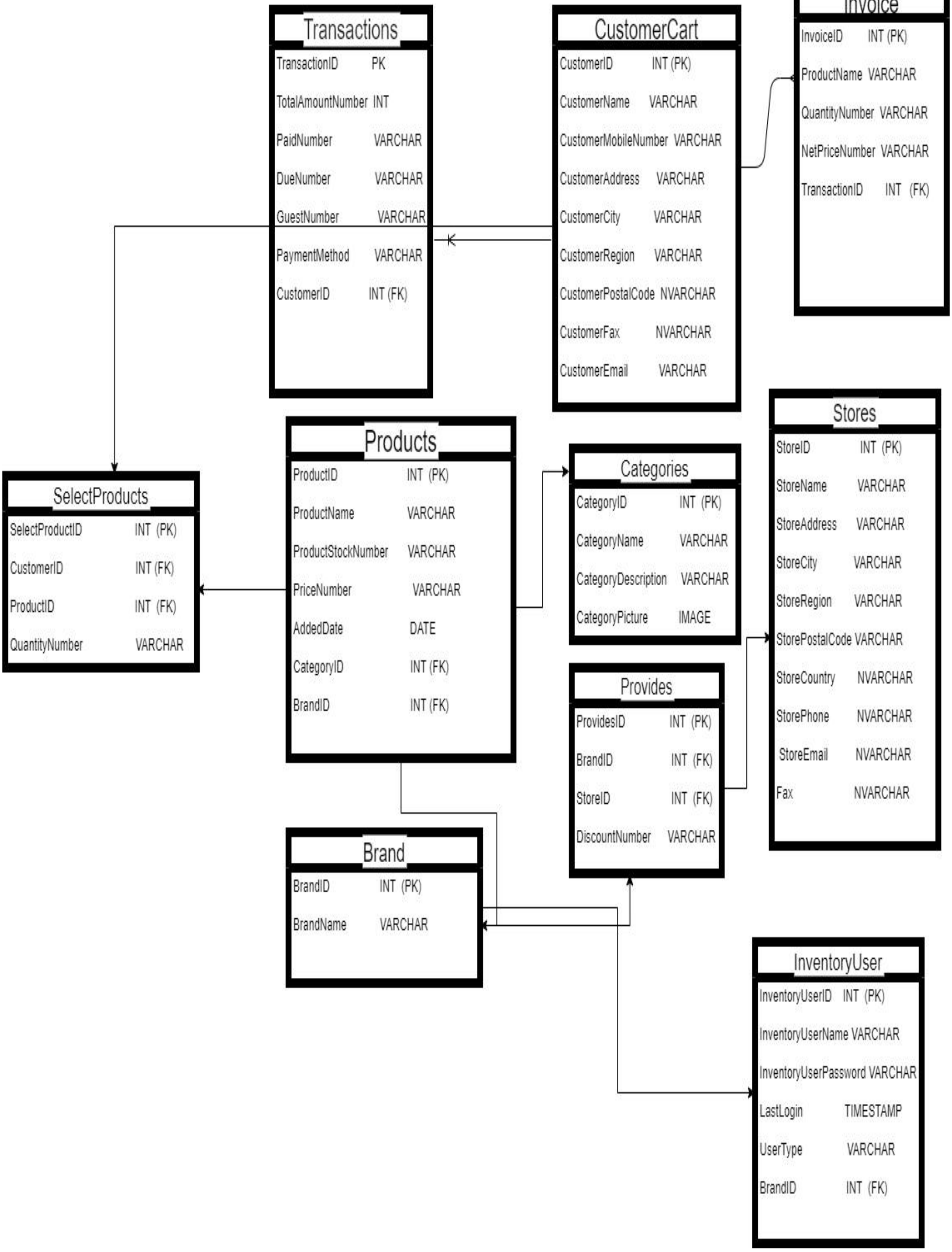
Provides

ProvidesID	INT (PK)
BrandID	INT
StoreID	INT
DiscountNumber	VARCHAR

InventoryUser

InventoryUserID	INT (PK)
InventoryUserName	VARCHAR
InventoryUserPassword	VARCHAR
LastLogin	TIMESTAMP
UserType	VARCHAR
BrandID	INT

2 NORMAL FORM



Transactions

TransactionID	PK
TotalAmountNumber	INT
PaidNumber	VARCHAR
DueNumber	VARCHAR
GuestNumber	VARCHAR
PaymentMethod	VARCHAR
CustomerID	INT (FK)

CustomerCart

CustomerID	INT (PK)
CustomerName	VARCHAR
CustomerMobileNumber	VARCHAR
CustomerAddress	VARCHAR
CustomerCity	VARCHAR
CustomerRegion	VARCHAR
CustomerPostalCode	NVARCHAR
CustomerFax	NVARCHAR
CustomerEmail	VARCHAR

Invoice

InvoiceID	INT (PK)
ProductName	VARCHAR
QuantityNumber	VARCHAR
NetPriceNumber	VARCHAR
TransactionID	INT (FK)

SelectProducts

SelectProductID	INT (PK)
CustomerID	INT (FK)
ProductID	INT (FK)
QuantityNumber	VARCHAR

Products

ProductID	INT (PK)
ProductName	VARCHAR
ProductStockNumber	VARCHAR
PriceNumber	VARCHAR
AddedDate	DATE
CategoryID	INT (FK)
BrandID	INT (FK)

Categories

CategoryID	INT (PK)
CategoryName	VARCHAR
CategoryDescription	VARCHAR
CategoryPicture	IMAGE

Stores

StoreID	INT (PK)
StoreName	VARCHAR
StoreAddress	VARCHAR
StoreCity	VARCHAR
StoreRegion	VARCHAR
StorePostalCode	VARCHAR
StoreCountry	NVARCHAR
StorePhone	NVARCHAR
StoreEmail	NVARCHAR
Fax	NVARCHAR

Provides

ProvidesID	INT (PK)
BrandID	INT (FK)
StoreID	INT (FK)
DiscountNumber	VARCHAR

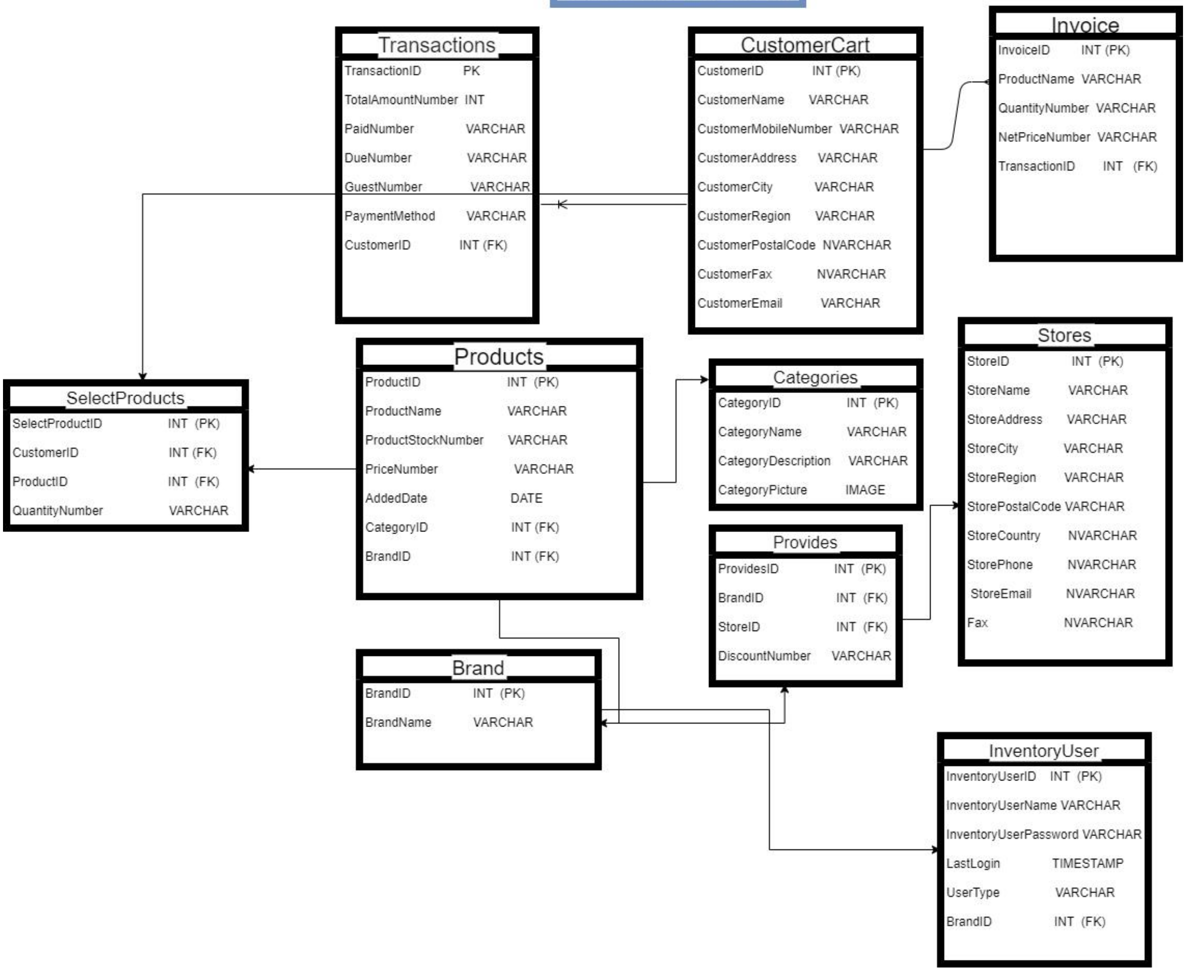
Brand

BrandID	INT (PK)
BrandName	VARCHAR

InventoryUser

InventoryUserID	INT (PK)
InventoryUserName	VARCHAR
InventoryUserPassword	VARCHAR
LastLogin	TIMESTAMP
UserType	VARCHAR
BrandID	INT (FK)

3 NORMAL FORM



c) Normalization is essentially a formal approach to applying a set of rules used to associate attributes with entities with the aim of identifying dependencies between data and minimizing data redundancy

i. The first step l did is creating an ERD with steps beginning with the **ZERO NORMAL FROM (0NF).**
Before starting with the normalization steps the first thing is to gather all attributes that are going to be necessary within the database, therefore I created a ZERO NORMAL FORM above which has all the attributes.

ii. **THE FIRST NORMAL FORM (1NF**) – this is the second step l took which is essentially creating tables that are related and the identification of each row with a PRIMARY KEY and eliminating duplicates, all entities are checked, and their duplicates are removed, thus the correct 1NF.

iii. **THE SECOND NORMAL FORM (2NF)** – the aim for the second normal form is to further remove subsets of data that apply to multiple rows of a table and place them in separate tables and importantly creating relationships between tables using FOREIGN KEYS.

iv. **THE THIRD NORMAL FORM (3NF)** – there are just two basic requirements for a database to be in third normal form, first it must meet the requirements of both the 1NF and 2NF and lastly all the columns that are not fully dependent upon the primary key must be removed.

d) A list of Tables, stored procedures, views, cursors, triggers and constraints.

I. **TABLES**

- **Brands** – This is a table for all the brands that are sold by the company.
- **InventoryUser** – This includes the users of the inventory.
- **Categories** – All types of Categories sold and kept in the database.
- **Products** – All the products in the inventory .
- **Stores** – All the stores that the database keep track of.
- **Provides** – This simply entails the store and the brand available for selling .
- **CustomerCart** – this includes the customer and the basket information on what they buy and their records .
- **SelectProduct** – This essentially provide the information for the customer and the products that they purchase.
- **Transactions** – includes all the transactions for the company.
- **Invoice** – the information needed by both the company and the customers.

II. **STORED PROCEDURES** – there are Procedures that gather information from different tables and check for any insert in order to update the data set and an alteration of the names of the procedure changing it.

III. **VIEWS** – there was a creation of two critical views which only displays clear and ready-made sequential graphical user interface information for the database users such as checking the Inventory levels and checking the customers who have their phone numbers for marketing purposes.

IV. **CURSORS** - there are two cursors configured the first for checking Categories and the other for checking Inventory Users and the login history.

V. **INDEXES** – there are basically 3 Non-Clustered Indexes that are configured, and a single Composite Index meant for data retrieval efficiency for most viewed objects.

VI. **TRIGGERS** – there are essentially 4 Triggers configured, for Updating, Inserting, Deleting and each giving information out after every Update by printing it to the user, and there is Trigger Sequence ordering.

VII. **CONSTRAINTS**

- Primary Keys – there are basically 9 Primary keys that are configured for every Table within the database.
- Foreign Keys – There are 11 Foreign Keys that I have set up creating the referential integrity

e) A list of all questions that must be answered by the database

i. **Inventory Balance** – there is need for inventory management which assist in maintain good stock levels and not creeping in on the working capital

ii. Inventory Turnover – thus by making sure the products are not becoming obsolete, spoiling and making calculations of selling products in order to assess where resources could be utilised.

iii. **Customer maintenance** – customers need to have a personal experience when they shop, therefore creating a database that gather their information and improve the connection can lead to better profits, at the end of the day and having history about everything they favour to buy is very vital.

iv. **Accurate planning** – knowing the stock in the inventory and stock rotation helps in staying in front of the curve such as keeping the right products for each season, thus keeping the clients satisfied.

v. **Employee efficiency** – by staying informed on what is selling a lot and what is left within the inventory through the database helps employees to utilise time and resources which makes them efficient and effective by simply starting by checking queries that give straight information about the inventory.