Title

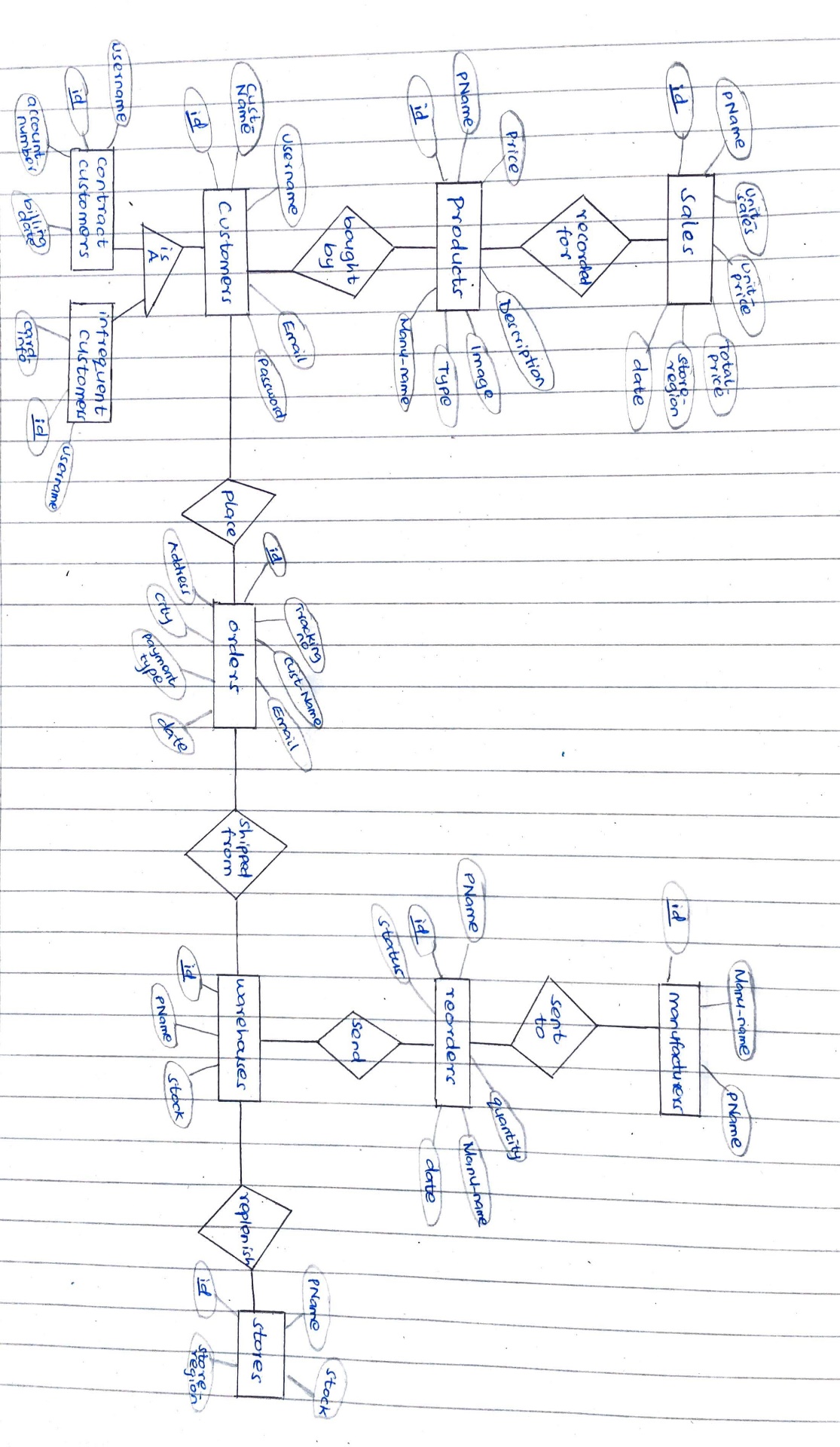
# ABSTRACT

# INRODUCTION

# RELATED WORK

# METHODS

## ER Diagram



## Relational Models

### Sales

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **PK**  id | pName | Unit\_sales | Unit\_price | Total\_Price | Store\_region | date |
|  |  |  |  |  |  |  |

### Products

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **PK**  id | pName | Price | Description | Image | Type | Manu\_name |
|  |  |  |  |  |  |  |

### Customers

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **PK**  id | Cust\_name | Username | Email | Password |
|  |  |  |  |  |

### Contract Customers

|  |  |  |  |
| --- | --- | --- | --- |
| **PK**  id | Username | Account\_number | Billing\_date |
|  |  |  |  |

### Infrequent Customers

|  |  |  |
| --- | --- | --- |
| **PK**  id | Username | Card\_info |
|  |  |  |

### Orders

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **PK**  id | Tracking\_no | Cust\_name | Email | Address | City | Payment\_type | date |
|  |  |  |  |  |  |  |  |

### Warehouses

|  |  |  |
| --- | --- | --- |
| **PK**  id | pName | Stock |
|  |  |  |

### Reorders

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **PK**  id | pName | quantity | Manu\_name | date | status |
|  |  |  |  |  |  |

### Stores

|  |  |  |  |
| --- | --- | --- | --- |
| **PK**  id | Store\_region | pName | Stock |
|  |  |  |  |

### Manufacturers

|  |  |  |
| --- | --- | --- |
| **PK**  id | Manu\_name | pName |
|  |  |  |

## SQL Statements

Below are the SQL statements used to create the tables:

CREATE TABLE IF NOT EXISTS "order" (

    "id"    INTEGER NOT NULL,

    "reference" VARCHAR(5),

    "first\_name"    VARCHAR(20),

    "last\_name" VARCHAR(20),

    "phone\_number"  INTEGER,

    "email" VARCHAR(50),

    "address"   VARCHAR(100),

    "city"  VARCHAR(100),

    "status"    VARCHAR(10),

    "payment\_type"  VARCHAR(10),

    "payment\_method"    VARCHAR(10),

    PRIMARY KEY("id")

);

CREATE TABLE IF NOT EXISTS "product" (

    "id"    INTEGER NOT NULL,

    "name"  VARCHAR(50),

    "price" INTEGER,

    "stock" INTEGER,

    "description"   VARCHAR(500),

    "image" VARCHAR(100),

    "manu\_name" TEXT,

    "type"  TEXT,

    PRIMARY KEY("id"),

    UNIQUE("name")

);

CREATE TABLE IF NOT EXISTS "sales" (

    "id"    INTEGER,

    "pName" TEXT,

    "unit\_sales"    INTEGER,

    "unit\_price"    INTEGER,

    "total\_price"   INTEGER,

    "store\_region"  TEXT,

    "date"  TEXT,

    PRIMARY KEY("id")

);

CREATE TABLE IF NOT EXISTS "customers" (

    "id"    INTEGER,

    "cust\_name" TEXT,

    "username"  TEXT,

    "email" TEXT,

    "password"  TEXT,

    PRIMARY KEY("id" AUTOINCREMENT)

);

CREATE TABLE IF NOT EXISTS "contract\_customers" (

    "id"    INTEGER,

    "username"  TEXT,

    "account\_number"    TEXT,

    "billing\_date"  TEXT,

    PRIMARY KEY("id" AUTOINCREMENT)

);

CREATE TABLE IF NOT EXISTS "infrequent\_customers" (

    "id"    INTEGER,

    "username"  TEXT,

    "card\_info" TEXT,

    PRIMARY KEY("id" AUTOINCREMENT)

);

CREATE TABLE IF NOT EXISTS "warehouses" (

    "id"    INTEGER,

    "pName" TEXT,

    "stock" INTEGER,

    PRIMARY KEY("id" AUTOINCREMENT)

);

CREATE TABLE IF NOT EXISTS "stores" (

    "id"    INTEGER,

    "store\_region"  TEXT,

    "pName" TEXT,

    "stock" INTEGER,

    PRIMARY KEY("id" AUTOINCREMENT)

);

CREATE TABLE IF NOT EXISTS "reorders" (

    "id"    INTEGER,

    "pName" TEXT,

    "quantity"  INTEGER,

    "manu\_name" TEXT,

    "date"  TEXT,

    "status"    TEXT,

    PRIMARY KEY("id" AUTOINCREMENT)

);

CREATE TABLE IF NOT EXISTS "manufacturers" (

    "id"    INTEGER,

    "manu\_name" TEXT,

    "pName" TEXT,

    PRIMARY KEY("id" AUTOINCREMENT)

);

## Test Queries

Below are some of the queries that the client wanted turned in, used to test the database functionality. The others can be tested practically on the website, in the admin section.

* Find the top 2 products by price sold in the past year.

    def toptwo\_byprice(self):

        myConnection = sqlite3.connect('project.db')

        myCursor = myConnection.cursor()

        myCursor.execute('SELECT total\_price FROM Sales ORDER BY total\_price DESC LIMIT 2')

        myRS = myCursor.fetchall()

        myConnection.commit()

        myConnection.close()

        print(myRS)

* Find the top 2 products by unit sales in the past year.

    def toptwo\_byunitsales(self):

        myConnection = sqlite3.connect('project.db')

        myCursor = myConnection.cursor()

        myCursor.execute('SELECT unit\_sales FROM Sales ORDER BY unit\_sales DESC LIMIT 2')

        myRS = myCursor.fetchall()

        myConnection.commit()

        myConnection.close()

        print(myRS)

* Find those products that are out-of-stock at every store in Nairobi.

    def out\_of\_stock(self, store):

        myConnection = sqlite3.connect('project.db')

        myCursor = myConnection.cursor()

        myCursor.execute("SELECT pName FROM stores WHERE region = 'Nairobi' AND stock = 0")

        myRS = myCursor.fetchall()

        myConnection.commit()

        myConnection.close()

        print(myRS)

## Tools and Platform

## Directory Structure

# CHALLENGES FACED

# FUTURE WORK

# CONCLUSION

# Appendix