USE CASE STUDY REPORT

Group No: Group 23

Student Name: Sree Reshma Paramel

NUID: 002821717

Pharmacy Management System

Executive Summary:

The creation and application of a relational and logical model for pharmacies was the main goal of this project. This project offers insight into how a pharmacy management system is designed and put into practice. Making a database of the medications the store sells is how this is accomplished. The main goals of the pharmacy management system are to increase efficiency, safety, and accuracy in the drugstore.

The database was modelled with mainly the pharmacy and medicine tables, and the EER and UML diagrams were modelled with their following attributes, and the conceptual model is also mapped to the relational model with the prominent primary keys and foreign keys. This database was then implemented fully on the MySQL Workbench and a prototype with two tables were implemented on Neo4j using cypher queries.

The database was created by connecting it to Python. Different type of visualisations using the database were also implemented and different graphs were made. These queries can be very helpful in tracking anything about the pharmacies found in different locations and the medicines available in those pharmacies and when it is open or if it is out of stock.

Introduction:

The Pharmacy Management System program assists pharmacists in managing their pharmacies in a methodical manner. Any system used in a pharmacy to aid in the automation of the pharmacy process is known as pharmacy management software. When a medication's name is input, the Pharmacy Management System can provide details about it, which can simplify the process. A computer provides the medication's data, including its rate and expiration date. Large medical stores find it extremely challenging to manually handle the information of every medication, thus we can preserve the records of every medication by using this pharmacy management system. Every time a new medication is introduced, it receives updates with an expiration date and a search function. The details of the medication are provided when we read the complete name.

This include doing things like preparing prescriptions and analysing doctor orders, managing inventories and ordering drugs, managing insurance and billing, offering counselling, spotting inconsistencies, and more all while adhering to legal requirements and procedures.

The user of this pharmacy management system can also produce reports in a predetermined amount of time. For the purpose of opening stock and conducting sales transactions, the system enables the user to input the manufacturing and expiration dates for a certain drug or product. The technology in the pharmacy management system is strong and integrated. The pharmacist would want to create a report each month detailing the amount of pharmaceuticals coming into and going out of the pharmacy. This report would include details about the drugs themselves, such as their position within the pharmacy, expiration date, and date of purchase.

1. Conceptual Data Modeling

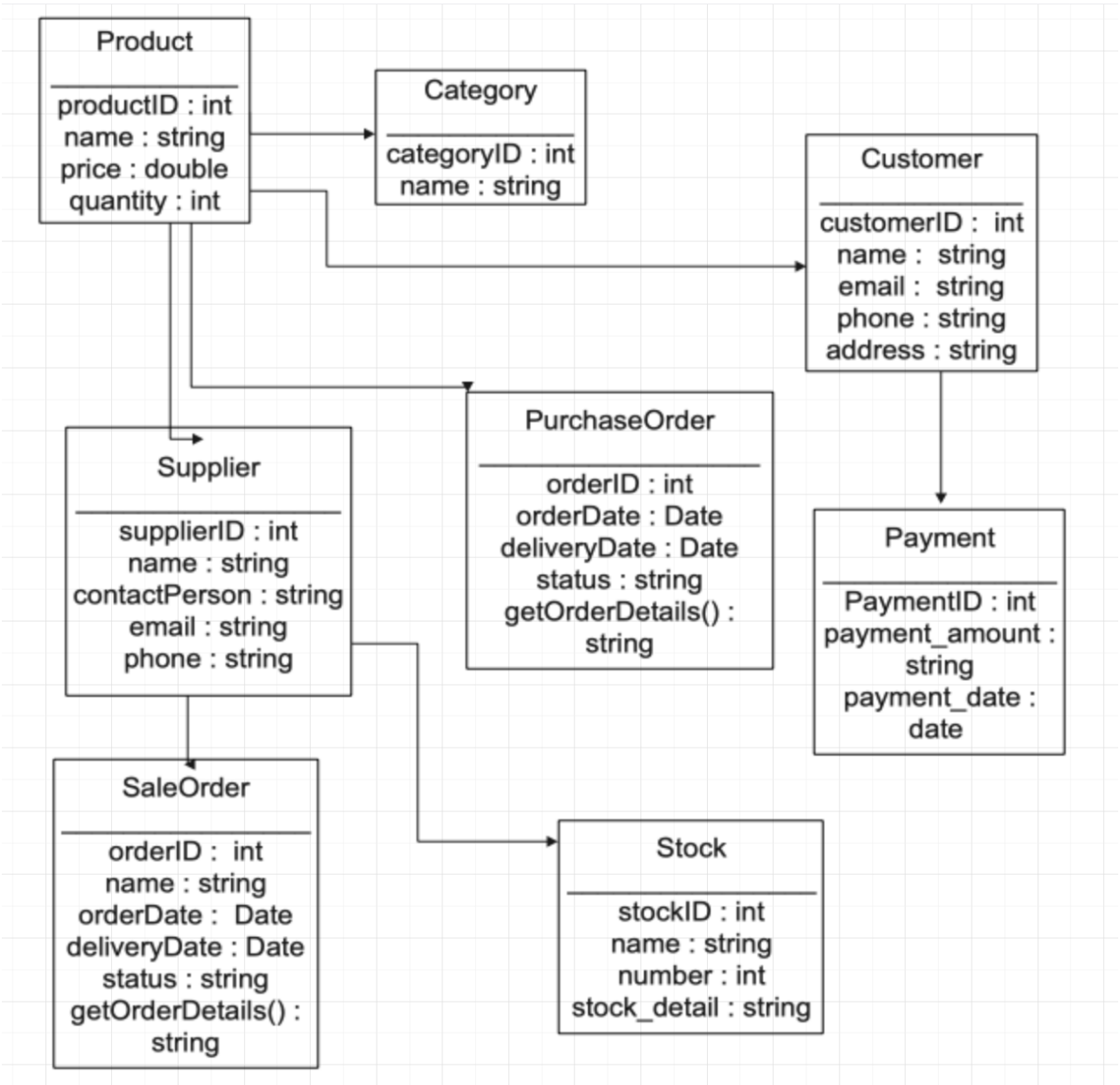
a. EER Model

b. UML Model

1 a) EER Model:


Name
Location
Patient_ID
Sex
Patient
Prescribes
Contact Information
Doctor_ID
Name
ContactNo
Doctor
Location
Speciality
Payment_ID
Prescription_ID
Drug_ID
Drug
Sell
Payment
Location
Cost Price
Name
Payment_method
Payment_date
Amount_paid
Pharmacy
Contact
Location
Pharmacy_ID
Name
Pharmacist_ID
Pharmacist
Name
Contact
Pharmacy_ID
Med_ID
Name
Pres_ID
Prescription
Medication
Description
Patient_ID
Date status
Pharm_ID
Price
Manufacturer

1 b) UML Diagram:



2. Mapping conceptual model to relational model

1. Patient: (PatientID, Name, DateOfBirth, Address, Phone, Sex)

2. Doctor: (DoctorID, Name, Specialization, Phone, Location)

3. Prescription: (PrescriptionID, *PatientID* (Foreign Key), Patient (PatientID), DoctorID FOREIGN KEY REFERENCES Doctor(DoctorID), Date, Status)

4. Medication: (MedicationID, Name, Manufacturer, Price, Quantity, SupplierID FOREIGN KEY REFERENCES Supplier (SupplierID))

5. Pharmacy: (PharmacyID, Name, Address, Phone) Pharmacist: (PharmacyID, Name, Contact, PharmacistID)

6. Payment: (PaymentID, PrescriptionID, Amount, Payment_date, Payment_method)

7. Supplier: (SupplierID, Name, Address, Phone)

8. Stock_alert: (Threshold_quantity, MedicationID)

9. Doctor: DoctorID (Primary Key), Name, Specialization, Phone, Location

10. Pharmacist: PharmacistID (Primary Key), PharmacyID (Foreign Key referencing Pharmacy table),Name, Contact

3. Implementation of Relation Model via MySQL

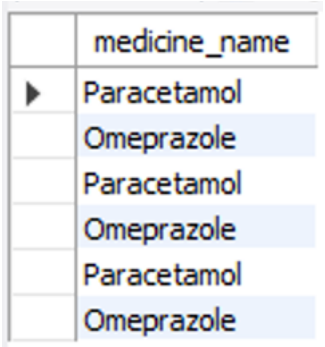
MySQL Implementation:
The database was created in MySQL Workbench and the following queries were performed:

Query 1: What are the names of medicines found in locations where the opening hours of pharmacy include '9:00 AM'?"

Code:

```
SELECT medicine_name
FROM medicine
WHERE location_found IN (
    SELECT location_found
    FROM medicine
    WHERE opening_hours LIKE '%9:00 AM%'
);
```

Output:



	medicine_name
▶	Paracetamol
	Omeprazole
	Paracetamol
	Omeprazole
	Paracetamol
	Omeprazole

Query 2: What is the total number of pharmacies overall, the distinct locations they are situated in, and the count of pharmacies in each city?

Code:

```
SELECT
    (SELECT COUNT(*) FROM pharmacy) AS total_pharmacies,
    location,
    (SELECT COUNT(*) FROM pharmacy p2 WHERE p1.location = p2.location) AS pharmacies_in_city
FROM
    (SELECT DISTINCT location FROM pharmacy) AS p1;
```

Output:

total_pharmacies	location	pharmacies_in_city
9	123 Main St, boston	1
9	456 Elm St, philadelphia	1
9	789 Oak St, MA	1
9	567 Sunset Blvd, Los Angeles	1
9	890 Maple Ave, Seattle	1
9	111 City Ave, New York	1
9	321 North St, Chicago	1
9	456 Central Ave, Houston	1

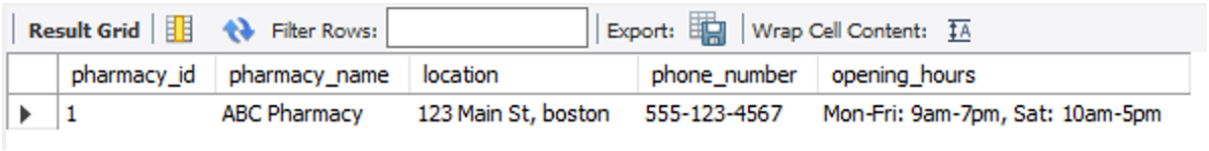
Query 3: Find a query which retrieve details of a pharmacy, including its ID, name, location, phone number, and opening hours, where the location matches that of 'ABC Pharmacy'?

Code:

```
SELECT pharmacy_id, pharmacy_name, location, phone_number, opening_hours
FROM pharmacy
```

```
WHERE location = (
    SELECT location
    FROM pharmacy
    WHERE pharmacy_name = 'ABC Pharmacy'
);
```

Output:



Result Grid | Filter Rows: | Export: | Wrap Cell Content:

pharmacy_id	pharmacy_name	location	phone_number	opening_hours
1	ABC Pharmacy	123 Main St, boston	555-123-4567	Mon-Fri: 9am-7pm, Sat: 10am-5pm

Query 4: Find the query which retrieves the names of medicines and the locations where they are found, filtering for medicines whose names contain the substring 'in' and sorting the results alphabetically by the location name?

Code:

```
SELECT medicine_name, location_found
FROM medicine
WHERE medicine_name LIKE '%in%'
ORDER BY location_found;
```

Output:

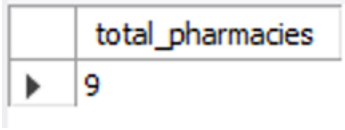
medicine_name	location_found
Aspirin	Pharmacy B
Aspirin	Pharmacy B
Aspirin	Pharmacy B
Amoxicillin	Pharmacy E
Amoxicillin	Pharmacy E
Amoxicillin	Pharmacy E
Lisinopril	Pharmacy F
Lisinopril	Pharmacy F

Query 5: Find the query to count the total number of pharmacies listed.

Code:

```
SELECT COUNT(*) AS total_pharmacies FROM pharmacy;
```

Output:



	total_pharmacies
▶	9

4. NoSQL Implementation

Queries were in Neo4j playground:

1. Write a query to retrieve all pharmacies in Seattle.

Code:

```
MATCH (p:pharmacy {location: 'Seattle'})
RETURN p
```

3. Write a query to find pharmacies which sell Paracetamol.

Code:

```
MATCH (p:pharmacy)-[:SOLD_AT]->(m:medicine {medicine_name: 'Paracetamol'})
RETURN p
```

4. Write a query to find all medicines available at pharmacies opening on Saturday.

Code:

```
MATCH (m:medicine)-[:SOLD_AT]->(p:pharmacy)
WHERE p.opening_hours CONTAINS 'Sat'
RETURN m
```

5. Database Access via Python

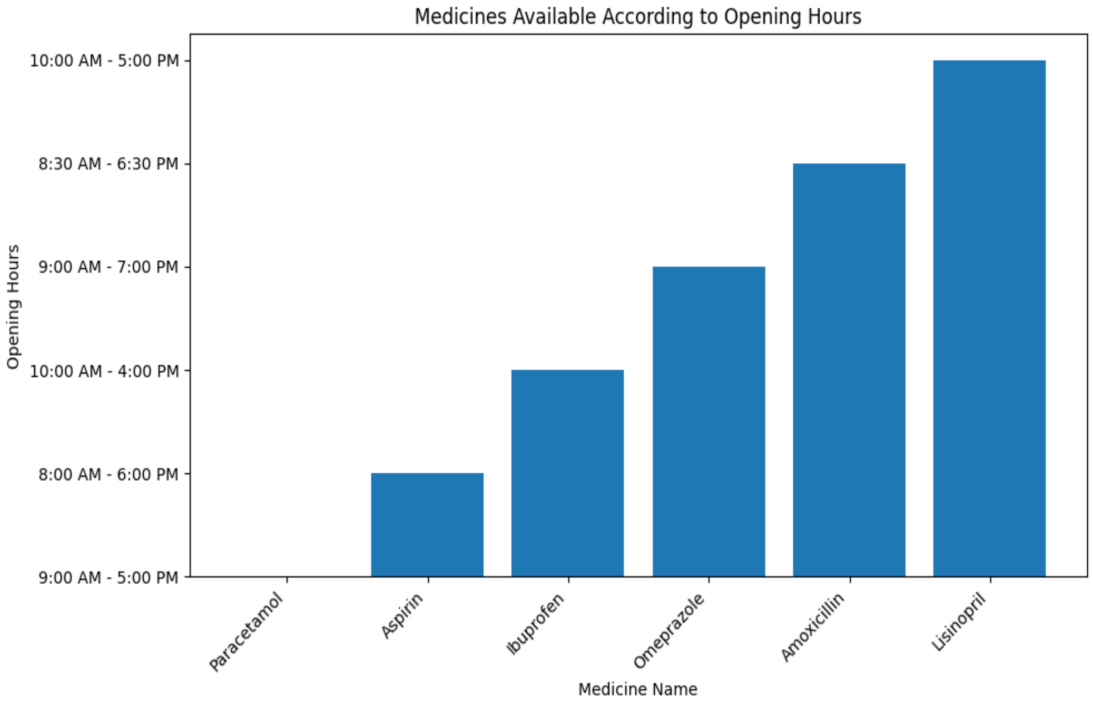
The database is accessed using Python and visualization of analyzed data is shown below. The connection of MySQL to Python is done using the below code:

```
!pip install mysql-connector-python
import mysql.connector
```

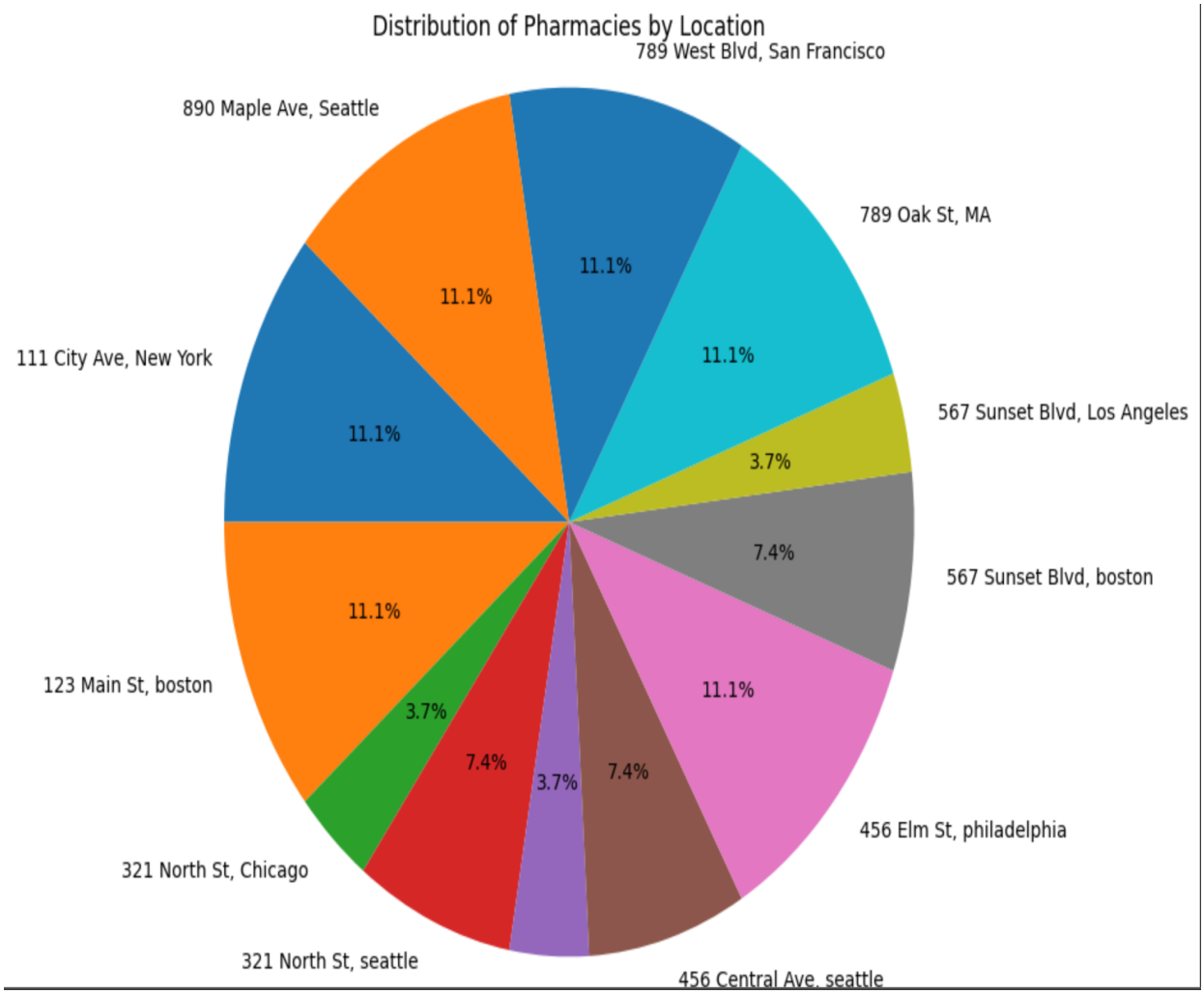
and a "**cursor.execute**" code to run and a **mycursor.fetchall()** query.

6. Visualizations from the data of 2 tables (Pharmacy and Medicine)

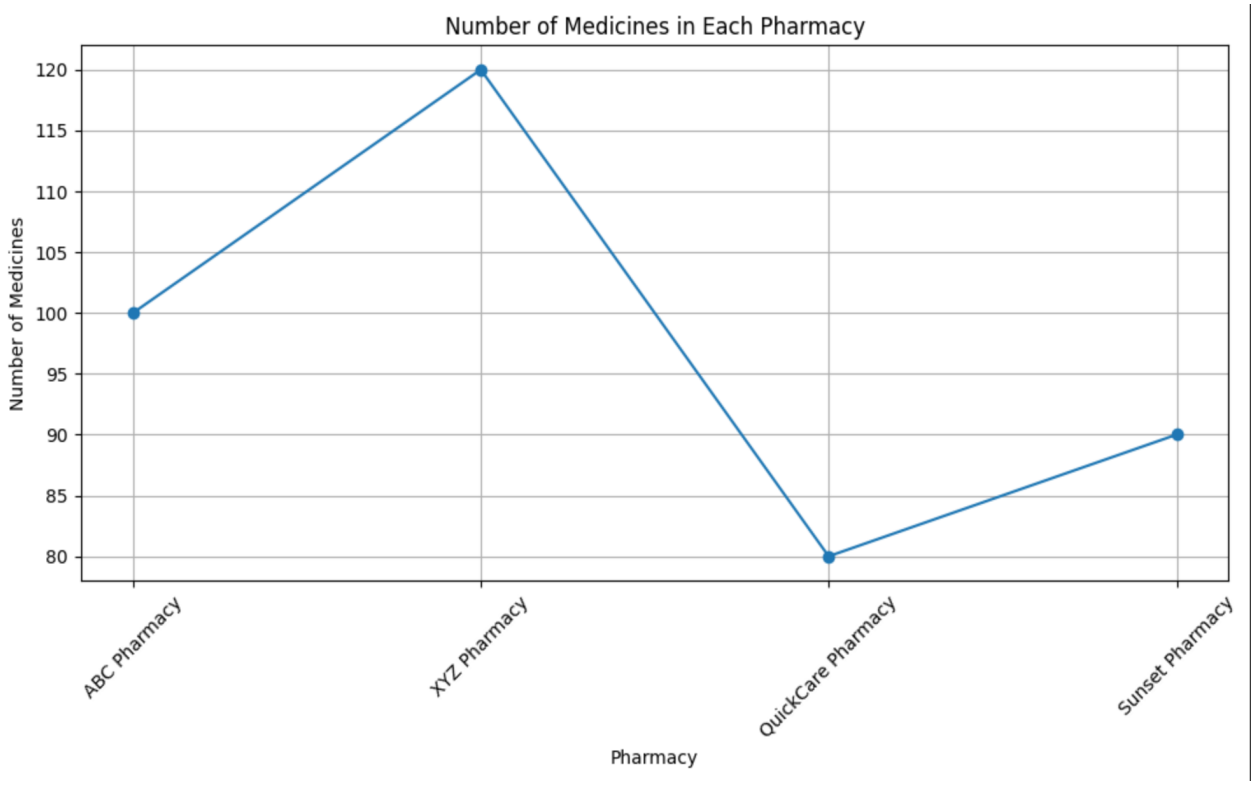
1. Medicines available according to opening hours (Bar chart)



2. Distribution of pharmacies by location (Pie Chart)



3. Number of medicines in each pharmacy (Line Chart)



7. Summary and Recommendations:

An important development in pharmacy operations is demonstrated by the deployment of a comprehensive Pharmacy Management System (PMS) that makes use of relational databases, NoSQL databases, and query languages including SQL, MongoDB, and Neo4j. The system effectively gets data from the relational database using SQL queries, making it easy for pharmacists to obtain specifics on prescription drugs, pharmacies, and their individual details. Furthermore, flexible data retrieval is made possible by MongoDB queries, which makes it easier to perform activities like name-based pharmaceutical searches, location-specific pharmacy inquiries, and pharmacy searches that provide the greatest number of available medications.

In conclusion, through the use of strong query languages in conjunction with relational and NoSQL databases, the system enables pharmacists to improve patient care, make well-informed judgments, and save a substantial amount of money for the industry. The PMS, with its extensive feature set and analytical powers, is a big step toward updating pharmacy management procedures and satisfying the changing demands of the healthcare industry.