DBMS Project

Team Members:

Saksham Suri(2015082)
Tushar Kataria(2015184)

Project Description:

1)Objective: We plan to develop a database management system for the pharmacies. Pharmacies have a large number of different medicines which makes it difficult for the chemist/seller to quickly retrieve the medicine. What we plan to do is implement a database management system which help the pharmacists by digitalising all the manual work such as creating invoice, storing records etc. It will also help the pharmacist store the history of purchases of a person as it may be needed in the future.

2)Feature: We will present different view to the user in order to answer the queries which we have mentioned below.

- View for purchasing medicines from different suppliers integrated with the product manager
- Product Manager view which lists different kinds of products
- Sale view which could be deployed on different POS(Point of Sale) terminals
- Monthly Ledger generator for generating accounting reports
- Breakage/Expiry view for updating stock in case of expiry and loss in transition

Queries:

- Creating purchase receipts**(Complex)**
    - Purchase Receipts will have the following effect on the database, the DRUG relation will be updated using update when the purchase is completed.The SUPPLIES relation will also be updated with entries linking distributer,Drug,Qty.
- Getting physical location about where medicine is present
    - This query will Select All Drugs to display list of all medicines and then use the select with ‘like’ to search the relation DRUG for the location of the required medicine
- Identifying medicines which are going to expire
    - This query will Select all name FROM DRUGS in which expiry_date.month()-current_date.month()<6 or expiry_date.month()<input.month and expiry_date.year()<input.year

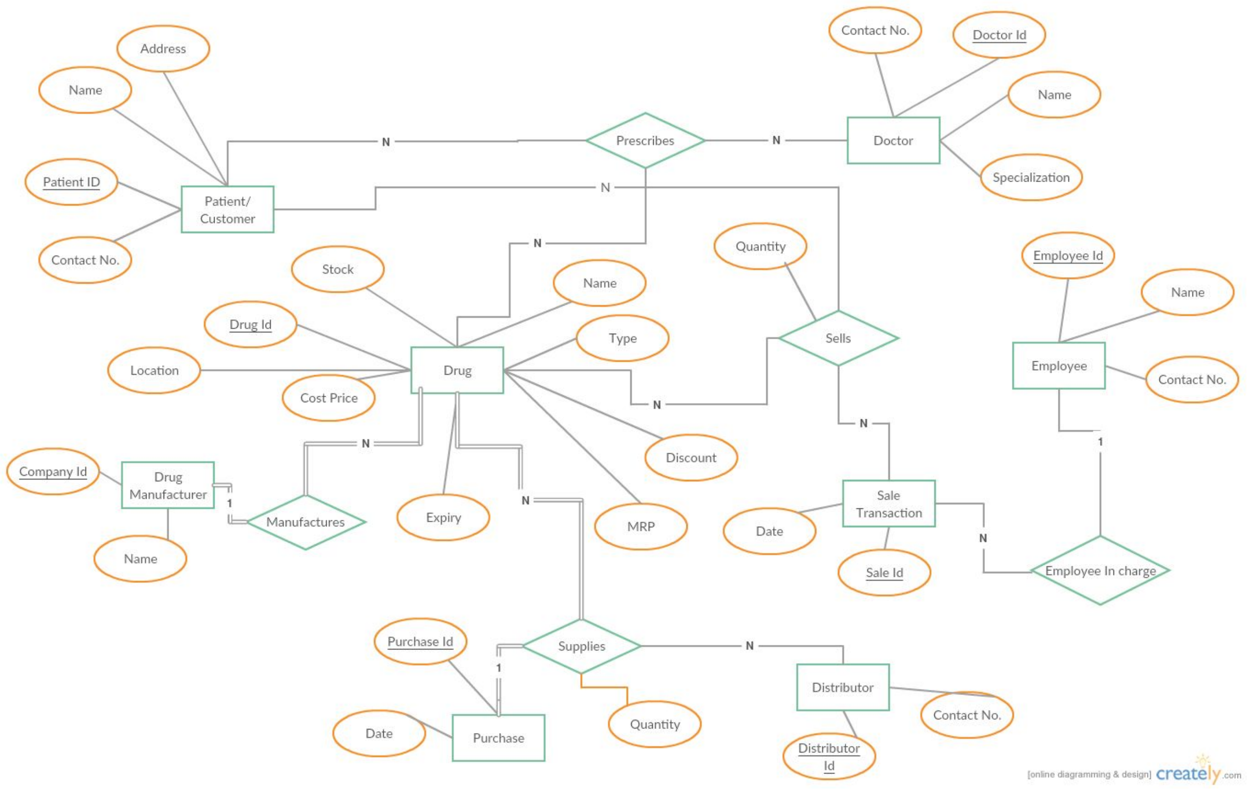
○ Create Purchase order for coming month**(Complex)**

■ This query will use the past transactions using the data from DRUG and SELLS by joining them

■ Now we divide this data month wise and using data for previous months we predict the quantity of medicines to be ordered for next month keeping in mind current stock

- Creating Sale Transactions**(Complex)**
    - Sale Transaction will have following effect on the database the DRUG relation will be updated and new entry in customer(if a new customer) and in doctor(if a new doctor) and a new entry in SELLS with a new entry in SALE_TRANSACTION be added to corresponding relations.
- Monthly transactions summary**(Complex)**
    - Retrieve data from SELLS and DRUG relations corresponding to particular month
    - Then group the sales day wise
    - For each day report the total money earned by summing up the sales for that day
- Gross profit sale Wise
    - For each sale We will query the SELLS relation for the various drug id and using them query the DRUGS relation to calculate the profit and then SUM up each of them using sale id.
- Performance Report**(Complex)**
    - We will use data from the EMPLOYEE , SALE_TRANSACTION , SELLS and DRUG relations
    - From these tables we'll separate out the sales of each employee separately
    - For all the sales of a particular employee calculate his/her total sales
    - Do this for all employees
- Purchase history
    - We will Select using patient ID from SELLS relation and then using the drug id calculate the profit made in the sale and display by grouping for a single patient on the sales made.
- Identify slow moving stock**(Complex)**
    - Select data from past sales from SELLS and DRUG
    - Using this data create a list of medicine which take a long time to sell and also lead to very low profits

ER Diagram



Constraints and the Relational Schema:

1. CUSTOMER(Patient_ID,Name,Address,Contact_No,Name)
    a. Patient_ID is primary key
2. DRUG_MANUFACTURER(Company_ID,Name)
    a. Company_ID is primary key
3. DOCTOR(Doctor_ID,Contact_No,Name,Specialization)
    a. Doctor_ID is primary key
4. SALE_TRANSCATION(Sale_ID,Date,Employee_ID)
    a. Sale_ID and Employee_ID form the primary key
    b. Sale_ID and Employee_ID are also foreign keys
5. PURCHASE(Date,Purchase_ID)
    a. Purchase_ID is primary key
6. SELLS(Sale_ID,Drug_ID,Patient_ID,Quantity)
    a. Sale_ID,Drug_ID and Patient_ID form the primary key
    b. Sale_ID,Drug_ID and Patient_ID are also foreign keys
7. PRESCRIBES(Patient_ID,Doctor_ID,Drug_ID)
    a. Drug_ID,Doctor_ID and Patient_ID form the primary key
    b. Drug_ID,Doctor_ID and Patient_ID are also foreign keys

8. DRUG(Location,Cost_Price,Stock,Drug_ID,Name,Discount,MRP,Company_ID,Expiry)
    a. Drug_ID is the primary key
    b. Company_ID is foreign key
9. SUPPLIES(Quantity,Purchase_ID,Distributor_ID,Drug_ID)
    a. Distributor_ID,Drug_ID and Purchase_ID form the primary key
    b. Distributor_ID,Drug_ID and Purchase_ID are also foreign keys
10. DISTRIBUTOR(Distributor_ID,Contact_No)
    a. Distributor_ID is the primary key
11. EMPLOYEE(Employee_ID,Name,Contact_No)
    a. Employee_ID is the primary key
12. Quantity can never be 0
13. No attributes can be NULL in general