PHARMACY DATABASE

BY:

Nikhil Saxena, 177242

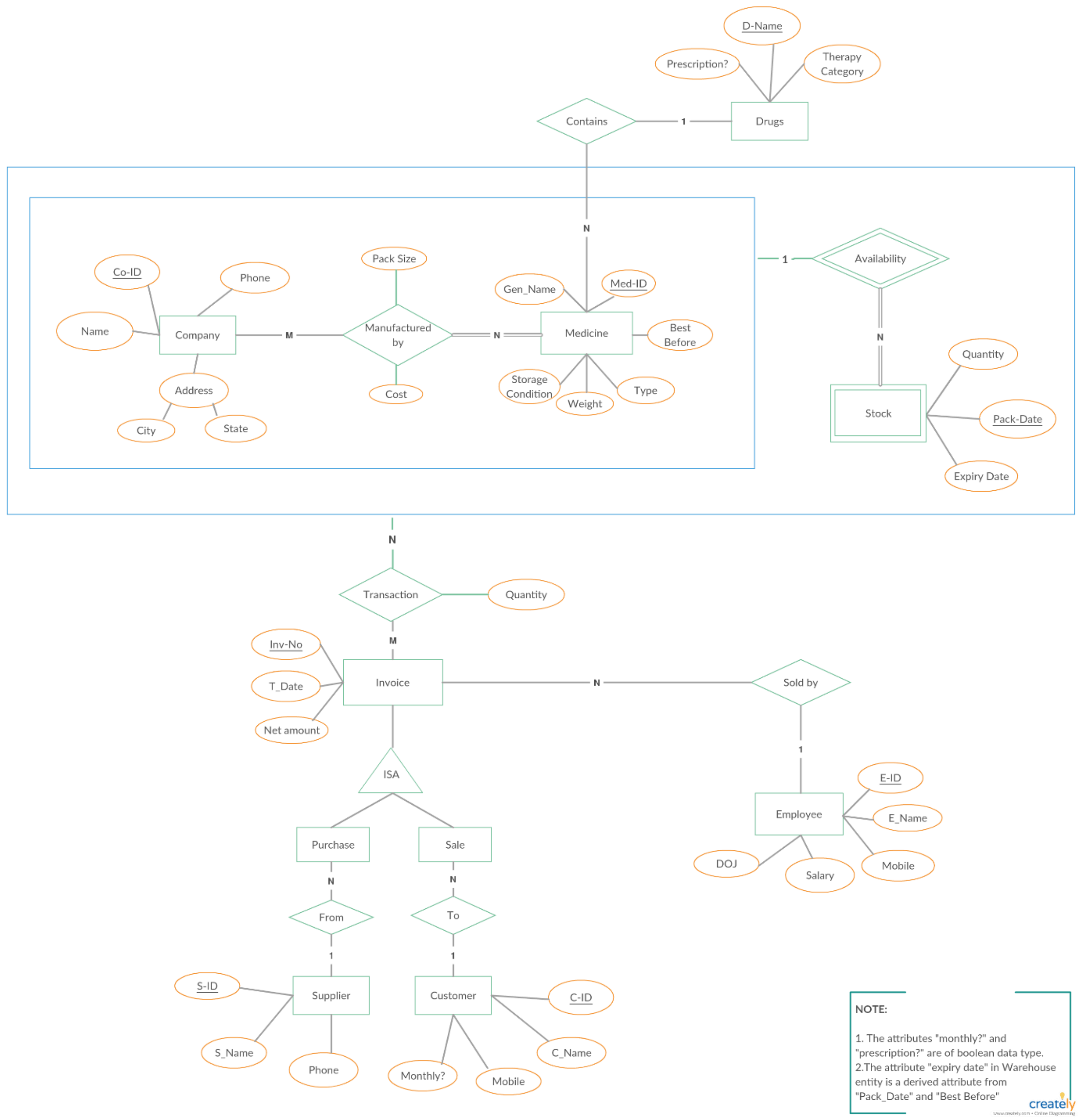
K.V.Preetam, 177226

PROBLEM STATEMENT -

To create database management system for pharmacy. It should store the details of medicines manufactured by different companies. Also maintain the details of employees working at the pharmacy, suppliers from whom the pharmacy buys the medicines, the customers.

- To manage the stock depending on all the day-to-day transactions happening from both the customer’s end as well as the supplier’s end.
- To maintain Invoices of all the transactions.
- To keep track of expiry dates of the medicines and alert the employee about medicines that will expire soon.
- To be able to display the medicines which are low in stock and need to be purchased again
- To display the most common medicines depending on sales figures.
- To notify monthly customers when to collect the medicines.

ENTITY-RELATIONSHIP MODEL:



NOTE:

1. The attributes "monthly?" and "prescription?" are of boolean data type.
2.The attribute "expiry date" in Warehouse entity is a derived attribute from "Pack_Date" and "Best Before"

ENTITIES:

- DRUGS (D-Name , Therapy Category , Prescription?)
- MEDICINE (Med-ID , D-Name , Gen_Name , Type , Weight , Storage condition , Best Before)
- COMPANY (Co-ID , Name , Phone , City , State)
- STOCK (Pack-date , QUANTITY , Expiry Date) -- *Weak entity*
- INVOICE (Inv-No , Amount , T_Date , T_Type)
- EMPLOYEE (E-ID , E_Name , Mobile , Salary , DOJ)
- SUPPLIER (S-ID , S_Name , Phone)
- CUSTOMER (C-ID , C_Name , Mobile , Monthly?)

RELATIONSHIPS:

- CONTAINS:
  1. DRUGS --- MEDICINE
  2. ‘DRUGS’ and ‘MEDICINE’ have a one to many relationship.
  3. Many medicines can be made with the same drug but one medicine is made of one drug only.

- MANUFACTURED BY:
  1. COMPANY --- MEDICINE
  2. ‘COMPANY’ and ‘MEDICINE’ have a many to many relationship.
  3. Many companies make the same medicine with small changes and one company makes multiple medicines.
  4. Each company will have independent ‘COST’ and ‘PACK SIZE’.

- SOLD BY:
  1. EMPLOYEE --- INVOICE
  2. ‘EMPLOYEE’ and ‘INVOICE’ have a one to many relationship.
  3. One employee can handle multiple invoices but one invoice is prepared by one employee only.

- TRANSACTION:
  1. AGGREGATION --- INVOICE
  2. ‘INVOICE’ and ‘COMPANY+MEDICINE+STOCK’ have a many to many relationship.
  3. One invoice has many medicines in it and each medicine can be made by any of the available companies. And similarly, any combination of medicine and company can be present in multiple invoices.
  4. The invoice will have ‘QUANTITY’ of each medicine.

- AVAILABITLITY:
  1. AGGREGATION --- WAREHOUSE
  2. ‘MEDICINE+COMPANY’ and ‘STOCK’ will have a one to many relationship.
  3. One medicine prepared by a certain company can have different packaging dates

- FROM:
    1. PURCHASE --- SUPPLIER
    2. ‘PURCHASE’ and ‘SUPPLIER’ have a one to many relationship.
    3. We can have multiple purchases from a supplier but one purchase is made from one supplier only.

- TO:
    1. SALE --- CUSTOMER
    2. ‘SALE’ and ‘CUSTOMER’ have a one to many relationship.
    3. We can have multiple sales to a customer but one sale is made to one customer only.

AGGREGATIONS:

1. ‘COMPANY’ and ‘MEDICINE’: While checking the stock, we need to keep track of the company that a medicine has been manufactured by hence there is an aggregation of company and medicine.
2. ‘COMPANY’ and ‘MEDICINE’ and ‘STOCK’: Each transaction will vary according to the type of company of a medicine. And to keep the stock updated, it is added in the aggregation.

DESCRIPTION:

- In the entity ‘Invoice’, the attributes Co-ID and Med-ID are linked and come from the aggregation formed from the entities ‘Company’ and ‘Medicine’.
- In the entity ‘Stock’, the attribute Expiry Date is a derived attribute coming from Pack-Date and Best before attributes.
- There exist two types of transactions. One is purchase and the other one is sale. So, we use T_Type two differentiate between the two types and hence two different entities with an IS A relationship.
- ‘Medicine’ is in total participation with the entity ‘Company’.
- ‘Stock’ is in total participation with the aggregation of ‘Company’ and ‘Medicine’.

NORMAL FORM:

- DRUGS:

    D-Name → Therapy category,
    D-Name → Prescription?

    D-Name is the primary key. Since the non-key attributes are non-transitively dependent on the primary key, the table is in 3NF.

- MEDICINE:

  Med-ID $\rightarrow$ D-Name,
  Med-ID $\rightarrow$ Gen_Name,
  Med-ID $\rightarrow$ Type,
  Med-ID $\rightarrow$ Weight,
  Med-ID $\rightarrow$ Storage Condition

  Med-ID is the primary key. Since the non-key attributes are non-transitively dependent on the primary key, the table is in 3NF.

- COMPANY:

  Co-ID $\rightarrow$ Name,
  Co-ID $\rightarrow$ Phone,
  Co-ID $\rightarrow$ City
  City $\rightarrow$ State

  Co-ID is the primary key. It is in 2NF. State is transitively dependent on Co-ID. So we decompose the table into two new tables.

  COMPANY (Co-ID, Name, Phone, City) and CS (City, State).

  The common attribute of the two tables is City and $[City]^+$ gives us CS. Hence it is a lossless join decomposition. And the two new tables are in 3NF.

- CS:

  City $\rightarrow$ State

  City is the primary key and the table is in 3NF.

- MEDICINE—COMPANY AGGREGATION:

  Co-ID, Med-ID $\rightarrow$ Pack Size,
  Co-ID, Med-ID $\rightarrow$ Cost

  Co-ID, Med-ID combined make the primary key. Since the non-key attributes are non-transitively dependent on the primary key, the table is in 3NF.

- STOCK:

  Co-ID, Med-ID, Pack-Date $\rightarrow$ Quantity

  Co-ID, Med-ID, Pack-Date combined make the primary key. The table is in 3NF.

- INVOICE:

  Inv-ID → Amount,
  Inv-ID →T_Date,
  Inv-ID → T_Type

  Inv-ID is the primary key. Since the non-key attributes are non-transitively dependent on the primary key, the table is in 3NF.

- TRANSACTION:

  Inv-ID, Co-ID, Med-ID → Quantity

  Co-ID, Med-ID, Inv-ID combined make the primary key. The table is in 3NF.

- EMPLOYEE:

  E-ID → E-Name,
  E-ID → Mobile,
  E-ID → Salary,
  E-ID → DOJ

  E-ID is the primary key. Since the non-key attributes are non-transitively dependent on the primary key, the table is in 3NF.

- SUPPLIER:

  S-ID → S-Name,
  S-ID → Phone

  S-ID is the primary key. Since the non-key attributes are non-transitively dependent on the primary key, the table is in 3NF.

- CUSTOMER:

  C-ID → C-Name,
  C-ID → Mobile,
  C-ID → Monthly?

  C-ID is the primary key. Since the non-key attributes are non-transitively dependent on the primary key, the table is in 3NF.

TABLE CREATION:

```
CREATE TABLE DRUGS(

   D_NAME VARCHAR(20) PRIMARY KEY,

   PRESCRIPTION VARCHAR(5),

   THERAPY_CATEGORY VARCHAR(10)

);


CREATE TABLE MEDICINE(

   M_ID INT PRIMARY KEY,

   D_NAME VARCHAR(20),

   GENERAL_NAME VARCHAR(20),

   BBF_MONTHS INT,

   TYPEE VARCHAR(10),

   WEIGHT_MG INT,

   STORAGE_CONDITION VARCHAR(10),

   FOREIGN KEY (D_NAME) REFERENCES DRUGS(D_NAME)

);


CREATE TABLE CS(

   CITY VARCHAR(20) PRIMARY KEY,

   STATE VARCHAR(20)

);


CREATE TABLE COMPANY(

   CO_ID INT PRIMARY KEY,

   PHONE INT,

   C_NAME VARCHAR(20),

   CITY VARCHAR(20),

   FOREIGN KEY (CITY) REFERENCES CS(CITY)

);
```

```
CREATE TABLE MED_COM(
   CO_ID INT,
   M_ID INT,
   PACK_SIZE INT,
   PRICE INT,
   PRIMARY KEY (CO_ID,M_ID),
   FOREIGN KEY (CO_ID) REFERENCES COMPANY(CO_ID),
   FOREIGN KEY (M_ID) REFERENCES MEDICINE(M_ID)
);

CREATE TABLE STOCK(
   CO_ID INT,
   M_ID INT,
   PACK_DATE DATE,
   AVAILABLE INT,
   PRIMARY KEY (CO_ID,M_ID,PACK_DATE),
   FOREIGN KEY (CO_ID) REFERENCES COMPANY(CO_ID),
   FOREIGN KEY (M_ID) REFERENCES MEDICINE(M_ID)
);

CREATE TABLE EMPLOYEE(
   E_ID INT PRIMARY KEY,
   E_NAME VARCHAR(20),
   PHONE INT,
   SALARY INT,
   DOJ DATE
);
```

```
CREATE TABLE SUPPLIER(
   S_ID VARCHAR(10) PRIMARY KEY,
   S_NAME VARCHAR(20),
   PHONE INT
);

CREATE TABLE CUSTOMER(
   C_ID VARCHAR(10) PRIMARY KEY,
   C_NAME VARCHAR(20),
   PHONE INT,
   MONTHLY VARCHAR(5)
);

CREATE TABLE INVOICE(
   I_ID INT PRIMARY KEY,
   T_DATE DATE,
   AMOUNT INT,
   E_ID INT,
   T_TYPE_ID VARCHAR(10)
);

CREATE TABLE TRANSCTION(
   CO_ID INT,
   M_ID INT,
   I_ID INT,
   QUANTITY INT,
   PRIMARY KEY (CO_ID,M_ID,I_ID),
   FOREIGN KEY (CO_ID) REFERENCES COMPANY(CO_ID),
   FOREIGN KEY (M_ID) REFERENCES MEDICINE(M_ID),
   FOREIGN KEY (I_ID) REFERENCES INVOICE(I_ID)
);
```