DATABASE DESIGN FOR PHARMACY

CS/SE 6360.003
FINAL PROJECT

ARTH SHAH-AXS175430

RAJAT KOTI-RXK174330

DEVESH RATHO-DXR172530

Foreword

This project was a combined effort of ours and we are glad to present it to everyone.

The Pharmacy Database is our interpretation of how a Pharmacy Database might look like. The Pharmacy Database consists of everything one would expect in a real-world Pharmacy such as patient, drug manufacturer, doctor, drug description.

We would like to thank everyone who supported us and our special thanks to Dr. Nurcan Yuruk, for guiding us for this project. Her helpful tips and suggestions have enabled us to develop an effective project.

Thank you, and we hope you like it.

Table of Contents

Requirements

Services offered by Pharmacy

The pharmacy provides a range of drugs to the people that come to buy drugs. The drugs can be classified as ones that require a prescription or as one that can be bought over the counter, they require no prescriptions to be bought. The pharmacy may also offer preventive healthcare services such as vaccinations. Pharmacists at a pharmacy can also assist patients with potential negative effects caused by certain drugs.

Patient

Each patient has an insurance company which provides an insurance number for the patient that is used to refer to their insurance details. The patient has a phone number by which they can be contacted. A patient can visit multiple doctors on different days and obtain prescriptions for drugs that may be required for their illnesses. The patient can buy different quantities of drugs from the pharmacy.

Doctor

A doctor is identified by a Registration Number and works in a city. A doctor can be contacted through their respective phone number. Each doctor can write multiple prescriptions for a patient.

Drug Manufacturer:

Drug manufacturers are the companies that create the drugs that are provided to the pharmacy. Each company has a unique registration and is based in a specific location. The manufacturers supply the pharmacy in batches. They also have a quantity that they supply to the pharmacy.

Prescription:

The prescription is provided by a doctor to a patient and lists the medicines that the patient can buy from the pharmacy. The prescription consists of a date when it was issued and has a unique identification number. It also contains an expiry date after which the pharmacy may no longer allow the usage of the prescription to buy medicines. The prescription also lists if the patient is allowed refills for the medicines listed in the prescription.

Medicine Batch:

Medicines are available in the pharmacy in batches. Each batch has a unique batch number to identify them. The batches may come in different quantity based on the manufacturer that provides them. Each batch also has the cost of an individual medicine in it. All the medicines in a batch have a common date of expiration. Each drug that is part of a batch has a unique identification and a name by which it is known.

Modeling of Requirements as ER-Diagram:

The summarized/derived from the ER Diagram are –

1. A patient can have can have many prescriptions (1: N)

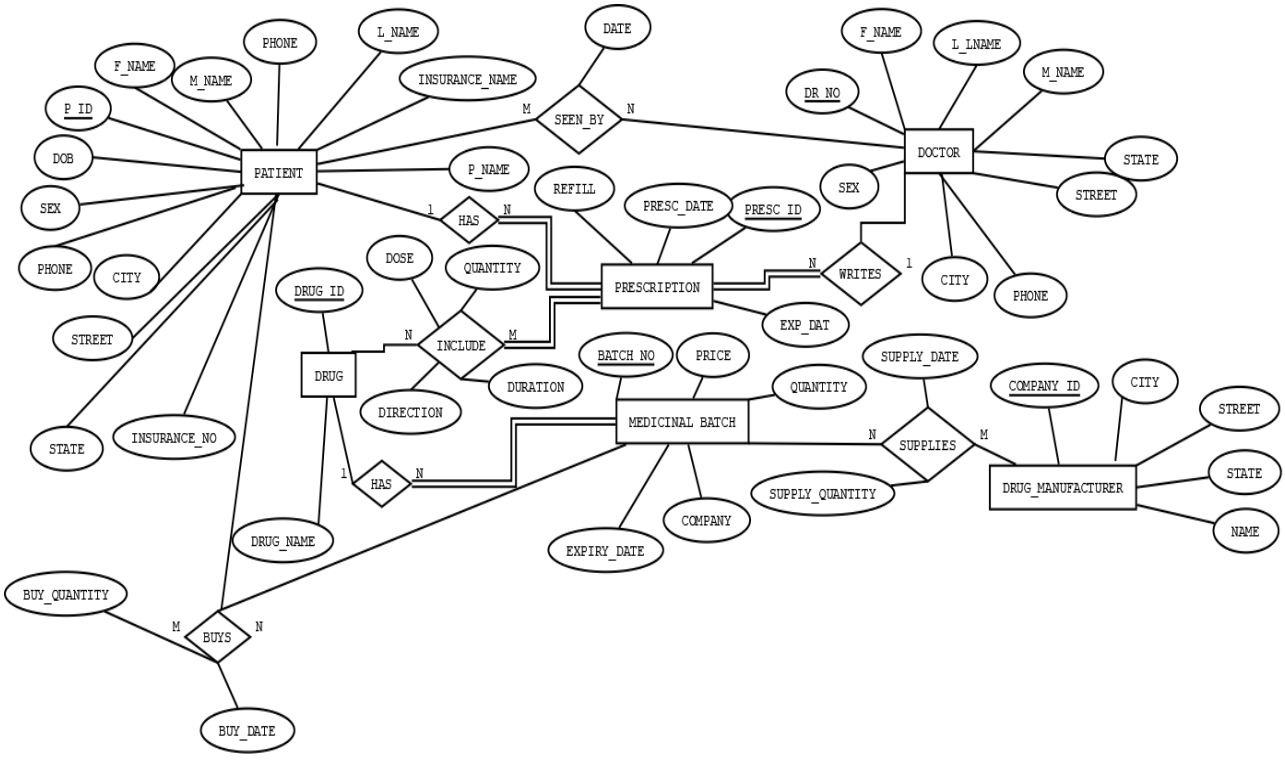
2. A patient can be seen by (examined by) different doctors at the same time and doctors can examine many different patients at the same time. (M: N)

3. A drug includes 1 or more services and a prescription can include by 1 or more drugs simultaneously. (M: N)

4. A drug can be a part of more than 1 Medicinal batch. (1: N).

5. A drug can be supplied by more than 1 drug manufacturer; a drug manufacturer can supply more than one type of drug (M: N).

6. A patient can buy more than 1 medicinal batch; a medicinal batch can be a bought by more than 1 patient (M: N).



Mapping of ERD in Relational Schema:

1. PATIENT

P_ID	F_NAME	M_NAME	L_NAME	PHONE	SEX	DOB	INSURANCE_NAME	INSURANCE_NUMBER	CITY	STREET	STATE

Primary Key: P_ID

2. DOCTOR

DR_NO	F_NAME	M_NAME	L_NAME	PHONE	SEX	SPECIALITY	STREET	STATE	CITY

Primary Key: DR_NO

3. MEDICINAL BATCH

BATCH_NO	QUANTITY	EXP_DATE	PRICE	DRUG_ID	COMPANY

Primary Key: BATCH_NO

Foreign Key: DRUG_ID References DRUG(DRUG_ID)

4. SEEN_BY

P_ID	DR_NO	CURRENT_DATE

Primary Key: P_ID, DR_NO

Foreign Key: P_ID References PATIENT(P_ID), DR_NO References DOCTOR(DR_NO)

5. BUYS

BATCH_NO	P_ID	QUANTITY	BATCH_NO

Primary Key: BATCH_NO, P_ID

Foreign Key: P_ID References Patient(P_ID), BATCH_NO References MEDICINAL BATCH(BATCH_NO)

6. PRESCRIPION

PRESC_ID	P_ID	DR_NO	PRESC_DATE	EXP_DATE	REFILL

Primary Key: PRESC_ID

Foreign Key: DR_NO References DOCTOR (DR_NO), P_ID References PATIENT(P_ID)

7. DRUG

DRUG_ID	DRUG_NAME

Primary Key: DRUG_ID

8. DRUG MANUFACTURER

COMPANY_ID	CITY	C_NAME	STATE	STREET

Primary Key: COMPANY_ID

9. SUPPLIES

BATCH_NO	COMPANY_ID	SUPPLY_DATE	SUPPLY_QUANTITY

Primary Key: BATCH_NO, COMPANY_ID

Foreign Key: BATCH_NO References BATCH(BATCH_ID), COMPANY_ID References DRUG MANUFACTURER(COMPANY_ID)

10. INCLUDES

PRESC_ID	DRUG_ID	DOSE	QUANTITY	DIRECTION	DURATION

Primary Key: PRESC_ID, DRUG_ID

Foreign Key: PRESCRIPTION References BATCH(PRESC_ID), DRUG_ID References DRUG (DRUG_ID)

SQL Statements to create Relations in DB and Add Constraints

```
CREATE TABLE BUYS (
BUY_DATE DATE NOT NULL,
QUANTITY INT NOT NULL,
P_ID VARCHAR2(20) DEFAULT NULL,
BATCH_NO VARCHAR2(20) DEFAULT NULL
)

CREATE TABLE DOCTOR(
DR_NO VARCHAR2(20) NOT NULL,
F_NAME VARCHAR2(50) NOT NULL,
M_NAME VARCHAR2(50) DEFAULT NULL,
L_NAME VARCHAR2(50) DEFAULT NULL,
PHONE VARCHAR2(10) NOT NULL,
SEX VARCHAR2(10) NOT NULL,
SPECIALITY VARCHAR2(20) DEFAULT NULL,
CITY VARCHAR2(20) DEFAULT NULL,
STREET VARCHAR2(30) DEFAULT NULL,
STATE_NAME VARCHAR2(50) DEFAULT NULL,
PRIMARY KEY (DR_NO)
)

CREATE TABLE DRUG (
DRUG_ID VARCHAR2(20) NOT NULL,
DRUG_NAME VARCHAR2(50) NOT NULL,
PRIMARY KEY (DRUG_ID)
)
```

```
CREATE TABLE DRUG_MANUFACTURER (
COMPANY_ID VARCHAR2(20) NOT NULL,
CITY VARCHAR2(20) NOT NULL,
C_NAME VARCHAR2(20) NOT NULL,
STATE_NAME VARCHAR2(20) NOT NULL,
STREET VARCHAR2(20) NOT NULL,
PRIMARY KEY (COMPANY_ID)
)

CREATE TABLE INCLUDES (
DOSE VARCHAR2(20) NOT NULL,
QUANTITY INT NOT NULL,
DIRECTION VARCHAR2(20) DEFAULT NULL,
PRESC_DATE DATE NOT NULL,
DRUG_ID VARCHAR2(20) DEFAULT NULL,
DURATION VARCHAR2(20) NOT NULL
)

CREATE TABLE MEDICINAL_BATCH (
BATCH_NO VARCHAR2(20) NOT NULL,
QUANTITY INT NOT NULL,
EXP_DATE DATE NOT NULL,
PRICE VARCHAR2(20) NOT NULL,
DRUG_ID VARCHAR2(20) DEFAULT NULL,
COMPANY VARCHAR2(20) NOT NULL,
PRIMARY KEY (BATCH_NO),
CONSTRAINT FK_DRUG_ID FOREIGN KEY (DRUG_ID) REFERENCES DRUG(DRUG_ID)
```

```
)

CREATE TABLE PATIENT (
P_ID VARCHAR2(20) NOT NULL,
F_NAME VARCHAR2(50) NOT NULL,
M_NAME VARCHAR2(50) DEFAULT NULL,
L_NAME VARCHAR2(50) DEFAULT NULL,
PHONE VARCHAR2(10) NOT NULL,
SEX VARCHAR2(10) NOT NULL,
DOB DATE NOT NULL,
INSURANCE_NAME VARCHAR2(20) NOT NULL,
INSURANCE_NO VARCHAR2(20) NOT NULL,
CITY VARCHAR2(20) DEFAULT NULL,
STREET VARCHAR2(30) DEFAULT NULL,
STATE_NAME VARCHAR2(50) DEFAULT NULL,
PRIMARY KEY (P_ID)
)

CREATE TABLE PRESCRIPTION (
PRESC_ID VARCHAR2(20) NOT NULL,
P_ID VARCHAR2(20) DEFAULT NULL,
DR_NO VARCHAR2(20) DEFAULT NULL,
PRESC_DATE DATE NOT NULL,
EXP_DATE DATE NOT NULL,
PRIMARY KEY (PRESC_ID),
CONSTRAINT FK_DR_NO FOREIGN KEY (DR_NO) REFERENCES DOCTOR(DR_NO),
CONSTRAINT FK_P_ID FOREIGN KEY (P_ID) REFERENCES PATIENT(P_ID)
)
```

```
CREATE TABLE SEEN_BY (
P_ID VARCHAR2(20) DEFAULT NULL,
DR_NO VARCHAR2(20) DEFAULT NULL,
CURR_DATE DATE NOT NULL,
CONSTRAINT FK_SEEN_PATIENT FOREIGN KEY (P_ID) REFERENCES PATIENT(P_ID),
CONSTRAINT FK_SEEN_DR FOREIGN KEY (DR_NO) REFERENCES DOCTOR (DR_NO)
)

CREATE TABLE SUPPLIES (
SUPPLY_DATE DATE NOT NULL,
BATCH_NO VARCHAR2(20) DEFAULT NULL,
COMPANY_ID VARCHAR2(20) DEFAULT NULL,
SUPPLY_QUANTITY INT NOT NULL,
CONSTRAINT FK_SUPPLY_BATCH FOREIGN KEY (BATCH_NO) REFERENCES MEDICINAL_BATCH
(BATCH_NO),
CONSTRAINT FK_SUPPLY_COMPANY FOREIGN KEY (COMPANY_ID) REFERENCES DRUG_MANUFACTURER
(COMPANY_ID)
)
```

Normalization of Relational Schema

The following Functional Dependencies exists in the relational schema –

1. PATIENT {P_ID -> PHONE, SEX, INSURANCE_NAME, INSURANCE_NO, DOB, F_NAME, M_NAME, L_NAME, STREET, CITY, STATE}
2. DOCTOR {DR_NO->PHONE, SEX, SPECIALITY, F_NAME, M_NAME, L-NAME, STREET, CITY, STATE}
3. PRESCRIPTION{PRESC_ID->PRESC_DATE, REFILL, EXP_DATE}
4. DRUG {DRUG_ID->DRUG_NAME}
5. MEDICINAL_BATCH {BATCH_NO->COMPANY, EXP_DATE, PRICE, QUANTITY}
6. DRUG_MANUFACTURER{COMPANY_ID->NAME, STREET, CITY, STATE}
7. BUYS {BATCH_NO, P_ID->BUY_DATE, BUY_QUANTITY}
8. INCLUDE {PRESC_ID, DRUG_ID -> DOSE, QUANTITY, DIRECTION, DURATION}
9. SUPPLIES {BATCH_NO, COMPANY_ID -> SUPPLY_DATE, SUPPLY_QUANTITY}

The functional dependencies defined above cause the schema to be in the Third Normal Form.

PL/SQL – Triggers

The following triggers are used to implement various requirements –

Trigger-I EXPIRED_MEDICINE

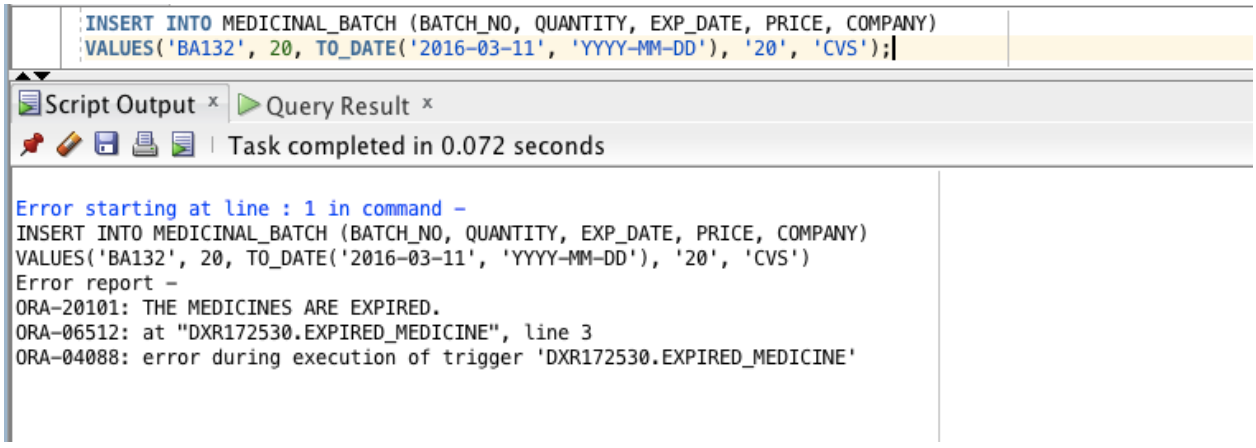
On entering a batch of medicines, if the Expiry Date is later than the current date, the operation is rejected.

Trigger:

```
CREATE OR REPLACE TRIGGER EXPIRED_MEDICINE
   BEFORE INSERT ON MEDICINAL_BATCH
   FOR EACH ROW
BEGIN
   IF :NEW.EXP_DATE<SYSDATE THEN
      RAISE_APPLICATION_ERROR(-20101, 'THE MEDICINES ARE EXPIRED.');
   END IF;
END;
```

Negative Test Case:

```
INSERT INTO MEDICINAL_BATCH (BATCH_NO, QUANTITY, EXP_DATE, PRICE, COMPANY)
VALUES('BA132', 20, TO_DATE('2016-03-11', 'YYYY-MM-DD'), '20', 'CVS');
```



Trigger-II EXPIRY_PRESCRIPTION:

A new prescription entry to the database is checked to ensure that expiry date for a prescription is less than 3 months from the current date.

Trigger:

```
CREATE or REPLACE TRIGGER EXPIRY_PRESCRIPTION
   BEFORE INSERT ON PRESCRIPTION
   FOR EACH ROW
BEGIN
   IF :NEW.EXP_DATE<ADD_MONTHS(SYSDATE, 3) THEN
      RAISE_APPLICATION_ERROR(-20101, 'Prescription will expire in 3 months');
   END IF;
END;
```

Negative Test Case:

```
INSERT INTO PRESCRIPTION (PRESC_ID, PRESC_DATE, EXP_DATE)
VALUES('P31132', TO_DATE('2017-12-04', 'YYYY-MM-DD'),TO_DATE('2018-3-04', 'YYYY-MM-DD'));
```

```
Error starting at line : 1 in command -
INSERT INTO PRESCRIPTION (PRESC_ID, PRESC_DATE, EXP_DATE)
VALUES('P31132', TO_DATE('2017-12-04', 'YYYY-MM-DD'),TO_DATE('2018-3-04', 'YYYY-MM-DD'))
Error report -
ORA-20101: Prescription will expire in 3 months
ORA-06512: at "DXR172530.EXPIRY_PRESCRIPTION", line 3
ORA-04088: error during execution of trigger 'DXR172530.EXPIRY_PRESCRIPTION'
```

PL/SQL- Procedures

Procedure-I Expiring_Batch

The procedure is used to check if the Medicine Batch will expire within a Month from the current date.

Procedure:

```
CREATE OR REPLACE PROCEDURE EXPIRING_BATCH(BATCHN IN
MEDICINAL_BATCH.BATCH_NO%TYPE) AS
   THIS_DATE MEDICINAL_BATCH.EXP_DATE%TYPE;
   THIS_BATCH MEDICINAL_BATCH.BATCH_NO%TYPE;
BEGIN
   SELECT BATCH_NO, EXP_DATE INTO THIS_BATCH, THIS_DATE FROM MEDICINAL_BATCH
WHERE BATCH_NO = BATCHN AND EXP_DATE < ADD_MONTHS(SYSDATE, 1);
   DBMS_OUTPUT.PUT_LINE('BATCH '||THIS_BATCH||' IS EXPIRING WITHIN A MONTH ON
'||THIS_DATE);
END;
```

Procedure-II Report Buys

The procedure generates a report of all the medicines that have been bought by patients over the last month.

Procedure:

```
CREATE OR REPLACE PROCEDURE REPORT_BUYS AS
   CURSOR INFO IS
   SELECT BUYS.BUY_DATE, BUYS.QUANTITY, PATIENT.F_NAME, PATIENT.L_NAME,
DRUG.DRUG_NAME
```

```
   FROM BUYS, PATIENT, MEDICINAL_BATCH, DRUG
   WHERE BUYS.P_ID = PATIENT.P_ID
   AND MEDICINAL_BATCH.BATCH_NO = BUYS.BATCH_NO
   AND DRUG.DRUG_ID = MEDICINAL_BATCH.DRUG_ID
   AND BUYS.BUY_DATE > ADD_MONTHS(SYSDATE, -1);
   THIS_ROW INFO%ROWTYPE;

BEGIN
   OPEN INFO;
   LOOP
      FETCH INFO INTO THIS_ROW;
      EXIT WHEN (INFO%NOTFOUND);
      DBMS_OUTPUT.PUT_LINE(THIS_ROW.F_NAME||' '||THIS_ROW.L_NAME||' bought
'||THIS_ROW.QUANTITY||' of '||THIS_ROW.DRUG_NAME||' on '||THIS_ROW.BUY_DATE);
   END LOOP;
   CLOSE INFO;
END;
```