Medical Store Management System
Phase-3

Shikhar Sharma
10682
shikhars@iitk.ac.in

CS315 Course Project
IIT Kanpur

March 19, 2013

Contents

1 EER/ER diagram of Phase-2 converted into tables

medicine (name, buy_timestamp, expiry_date, chem_amount, qty, cp, sp, pharmaco)

medicine_compound (name, buy_timestamp, expiry_date, chem_amount, cp, compound)

transaction (id, txn_timestamp, buy_sell, notes)

person (pid, name, address)

person_email (pid, email)

person_tel_no (pid, tel_no)

supplier_pharmaco (pid, pharmaco)

employee (pid, salary, duty_timings)

txn_on (name, buy_timestamp, chem_amount, expiry_date, cp, id, qty_buy_sell)

txn_person (id, pid_person, pid_employee)

2 Different kinds of integrity constraints satisfied by the DB

2.1 Entity Integrity

- None of the primary keys has a null value.
- The primary keys have been underlined in the section above this one.

2.2 Referential Integrity

- 'name', 'chem_amount', 'cp' in the relation **medicine_compound** are foreign keys referencing the same fields in the relation **medicine**.
- 'pid' in the relation **person_email** is a foreign key referencing the same field in the relation **person**.
- 'pid' in the relation **person_tel_no** is a foreign key referencing the same field in the relation **person**.
- 'pid' in the relation **supplier_pharmacol** is a foreign key referencing the same field in the relation **person**.

2.3 Domain Integrity

- They will be mentioned in the CREATE statements in Section 4.

2.4 User Defined Integrity

- medicine ≥ 0
- cp ≥ 0
- sp ≥ 0
- qty_buy_sell > 0
- salary ≥ 0
- buy_sell IN ('b','s')

3 Tables after normalization

medicine (name, buy_timestamp, expiry_date, chem_amount, qty, cp, sp)

name_pharma (name, pharmaco)

name_compound (name, compound)

transaction (id, txn_timestamp, buy_sell, notes)

person (pid, name, address)

person_email (pid, email)

person_tel_no (pid, tel_no)

supplier_pharmaco (pid, pharmaco)

employee (pid, salary, duty_timings)

txn_on (name, buy_timestamp, chem_amount, expiry_date, cp, id, qty_buy_sell)

txn_person (id, pid_person, pid_employee)

3.1 Justification

The functional dependencies are :

1. name → pharmaco
2. name → compound
3. name buy_timestamp chem_amount cp expiry_date → sp Qty pharmaco
4. id → txn_timestamp buy_sell notes
5. pid → name address
6. email → pid
7. tel_no → pid
8. pid → salary duty_timings
9. id name buy_timestamp chem_amount cp expiry_date → qty_buy_sell

The tables **transaction**, **person**, **person_email**, **person_tel_no**, **supplier_pharmaco**, **employee**, **txn_on**, and **txn_person** will stay unchanged.

Due to FD#2, the table **medicine_compound** will be split into **name_compound**(name, compound) and **name_details**(name, buy_timestamp, expiry_date, chem_amount, cp) but the table **name_details** is already contained in the table **medicine** and is therefore not needed.

Due to FD#1, the table **medicine** will be split into **medicine**(name, buy_timestamp, expiry_date, chem_amount, quantity, cp, sp) and **name_pharma**(name, pharma)

4 Create statements for the DB design (including the necessary integrity constraints)

-
  ```
  CREATE TABLE medicine (
      name varchar(60) NOT NULL,
      buy_timestamp timestamp NOT NULL,
      expiry_date date NOT NULL,
      chem_amount varchar(10) NOT NULL,
      qty int NOT NULL,
      cp int NOT NULL,
      sp int NOT NULL,
      PRIMARY KEY(name, buy_timestamp, expiry_date, chem_amount, cp)
  );
  ```

-
  ```
  CREATE TABLE name_pharma (
      name varchar(60) references medicine(name),
      pharmaco varchar(50) NOT NULL,
      PRIMARY KEY(name, pharmaco)
  );
  ```

-
  ```
  CREATE TABLE name_compound (
      name varchar(60) references medicine(name),
      compound varchar(50) NOT NULL,
      PRIMARY KEY(name, compound)
  );
  ```

-
  ```
  CREATE TABLE transaction (
      id int PRIMARY KEY,
      txn_timestamp timestamp NOT NULL,
      buy_sell char(1) NOT NULL,
      notes text
  );
  ```

-
  ```
  CREATE TABLE person (
      pid int PRIMARY KEY NOT NULL AUTO_INCREMENT,
      name varchar(60) NOT NULL,
      address text NOT NULL
  );
  ```

-
  ```
  CREATE TABLE person_email (
      pid int references person(pid),
      email varchar(45) PRIMARY KEY NOT NULL
  );
  ```

-
  ```
  CREATE TABLE person_tel_no (
      pid int references person(pid),
      tel_no int PRIMARY KEY NOT NULL
  );
  ```

-
  ```
  CREATE TABLE supplier_pharmaco (
      pid int references person(pid),
      pharmaco varchar(50) NOT NULL,
      PRIMARY KEY(pid, pharmaco)
  );
  ```

-
  ```
  CREATE TABLE employee (
      pid int references person(pid),
      salary int NOT NULL,
      duty_timings varchar(20) NOT NULL
  );
  ```

-
  ```
  CREATE TABLE txn_on (
      name varchar(60) NOT NULL,
  ```

```
   buy_timestamp timestamp NOT NULL,
   chem_amount varchar(10) NOT NULL,
   expiry_date date NOT NULL,
   cp int NOT NULL,
   id int NOT NULL,
   qty_buy_sell int NOT NULL,
   PRIMARY KEY(name, buy_timestamp, chem_amount, expiry_date, cp,
id),
   FOREIGN KEY(id) REFERENCES transaction(id)
);
```

- CREATE TABLE txn_person (

  ```
     id int NOT NULL,
     pid_person int NOT NULL,
     pid_employee int NOT NULL,
     PRIMARY KEY(id, pid_person, pid_employee)
  );
  ```