

## Тема 13

### Документация – База от данни за верига аптеки

Изготвил: Веселин Вълков, ф.н. 72082

#### 1. Обхват на модела. Дефиниране на задачата.

Базата от данни за верига аптеки ще съхранява информация за данните във веригата. Изготвената база от данни ще се управлява от вътрешна система на веригата. Веригата може да има аптеки в няколко града. Възможно е в един град да има няколко аптеки. Аптеките еднозначно се определят от тяхното име, което се състои от името на града и името на улицата на която се намира.

Веригата има служители, като се съхранява тяхното име, телефонен номер и служебен номер. Определят се от служебния си номер. Служителите представляват продавачи и мениджъри. Продавач може да работи само в една аптека. Всеки управител отговаря за определен град, като се записва името на града, за който отговаря.

За доставчиците се пази името на фирмата, което я определя еднозначно. Съхранява се информация за лекарствените продукти, които веригата продава. Лекарствените продукти имат име, фирма, който ги е произвела, лекарствена форма, дали са по лекарско предписание и препоръчителна цена. Името ги определя еднозначно.

#### 2. Множества от същности и техните атрибути

- Аптеки - име, адрес
- Продавач - име, телефонен номер, служебен номер
- Управител - име, телефонен номер, град, за който са отговорни
- Доставчик - име на фирмата, град
- Продукт - цена, име, фирма, който го е произвела
  - Лекарствен продукт - лекарствена форма, лекарско предписание
  - Нелекарствен продукт - тип

#### 3. Домейн на атрибутите

- Аптеки - име : низ, адрес : низ
- Служители - име : низ, телефонен номер : низ, служебен номер : низ
- Продавач - име : низ, телефонен номер : низ, служебен номер : низ
- Управител - име : низ, телефонен номер : низ, град, за който са отговорни : низ
- Доставчик - име на фирмата : низ, град : низ
- Продукт : име : низ, фирма : низ, цена : реално число или NULL

- Лекарствен продукт - лекарствена форма : низ или NULL, лекарско предписание :0 или 1
- Не лекарствен продукт - тип : низ

#### **4. Връзки**

- В една аптека работят много продавачи. Продавачите могат да работят само в една аптека.
- В един град може да работи един управител. Един управител работи точно в един град.
- Един доставчик може да доставя на няколко аптеки. Една аптека може да има няколко доставчика.
- Една аптека може да продава много лекарствени продукти. Един лекарствен продукт може да се продава в много аптеки.

#### **5. Ключове**

- Аптека - име: еднозначно определя
- Служител - служебен номер: еднозначно определя
- Доставчик - име: еднозначно определя
- Лекарствен продукт - име: еднозначно определя

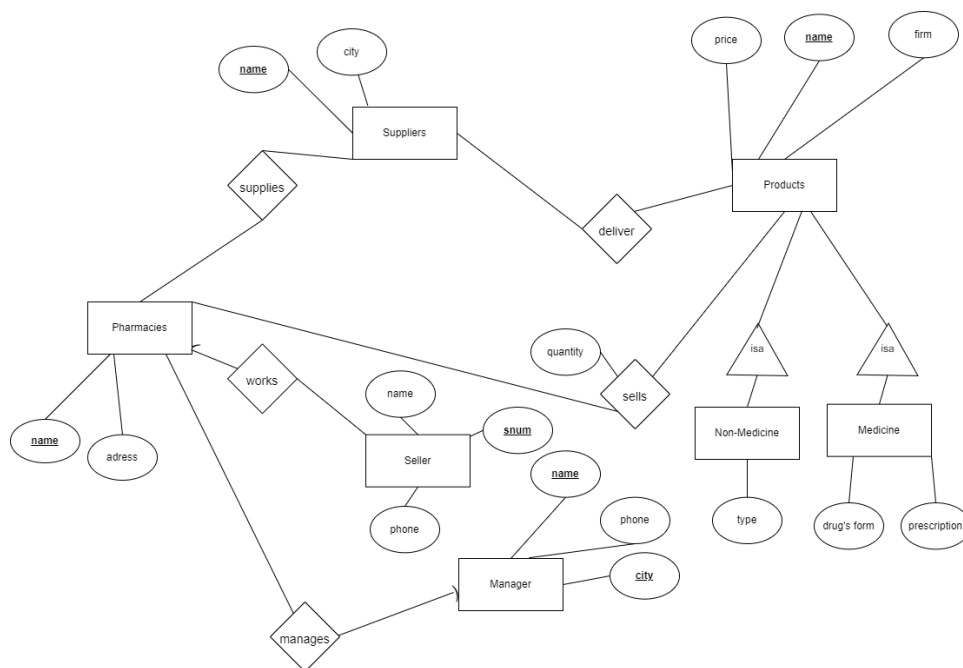
#### **6. Правила и проверки**

Лекарствен продукт - Да се прави проверка за лекарско предписание

Продавач - Да се прави проверка за валиден телефонен номер

Мениджър - Да се прави проверка за валиден телефонен номер

#### **7. E/R модел на данни**



## 8. Релационен модел на данни

Първо преобразуваме isa-йерархиите, после множествата от същности, после връзките много-много и накрая връзките много-един. Връзките много-един оптимизираме.

За преобразуването на i-sa йерархия може да използваме три подхода: ER, ООП и NULL. За is-a йерархията Products-Non-Medicine-Medicine най-подходящи са ООП и E/R.

От ООП подхода получаваме:

Products(price, name, firm) X (не създаваме, защото няма такава)

Products\_Non-Medicine(price, name, firm, type)

Products\_Medicine(price, name, firm, drug's form, prescription)

Products\_Non-Medicine\_Medicine(price, name, firm, type, drug's form, prescription) X (не създаваме, защото няма такава)

От E/R подхода получаваме:

Products(price, name, firm)

Products\_Non-Medicine(name, type)

Products\_Medicine(name, drug's form, prescription)

От начина на търсене на информацията ще се определи най-добрият подход. Ако търсенето става главно само в лекарствените продукти и/или в нелекарствените продукти тогава ООП ще е най-подходящия, защото информацията ще е най-точна. Ако се предимно се търсят продукти независимо от това какви са, тогава E/R ще е препоръчително, защото информацията ще е само в една таблица.

Ще използваме E/R подхода.

Следващите релационни схеми са както следва:

Pharmacies(name, address, manager\_snum)

Suppliers(name, city)

Manager(name, phone, city)

Seller(name, phone, snum)

Manages(pharmacy\_name, snum) X (оптимизираме)

Works(pharmacy\_name, snum) X (оптимизираме)

Supplies(supplier\_name, pharmacy\_name)

Deliver(supplier\_name, product\_name)

Sells(quantity, pharmacy\_name, product\_name)

Окончателно схемата на базата от данни се състои от следните релационни схеми:

Products(price, name, firm)

Non-Medicine(name, type)

Medicine(name, drug's form, prescription)

Manager(name, phone, city)

Seller(name, phone, snum, pharmacy\_name)

Pharmacies(name, address, manager\_name, city)

Suppliers(name, city)

Supplies(supplier\_name, pharmacy\_name)

Deliver(supplier\_name, product\_name)

Sells(quantity, pharmacy\_name, product\_name)

## 9. ФЗ и Нормализация

Когато се използва първо ER модел за да се направи дизайн на БД и след това се преобразува до релационни схеми всички ФЗ са в 3NF. Счита се, че ако схемата на БД е в НФБК, то дизайна на БД е добър. Така че, при използването на подхода първо абстрактен модел – ER диаграма, която след това се преобразува до релационни схеми, единствено трябва да проверим за ФЗ в релационните схеми, които нарушават НФБК.

В сила са следните ФЗ:

ФЗ-1: name -> price, firm, type (Non-Medicine)

ФЗ-2: name -> price, firm, drug's form, prescription (Medicine)

ФЗ-3: snum -> name, phone, pharmacy\_name (Seller)

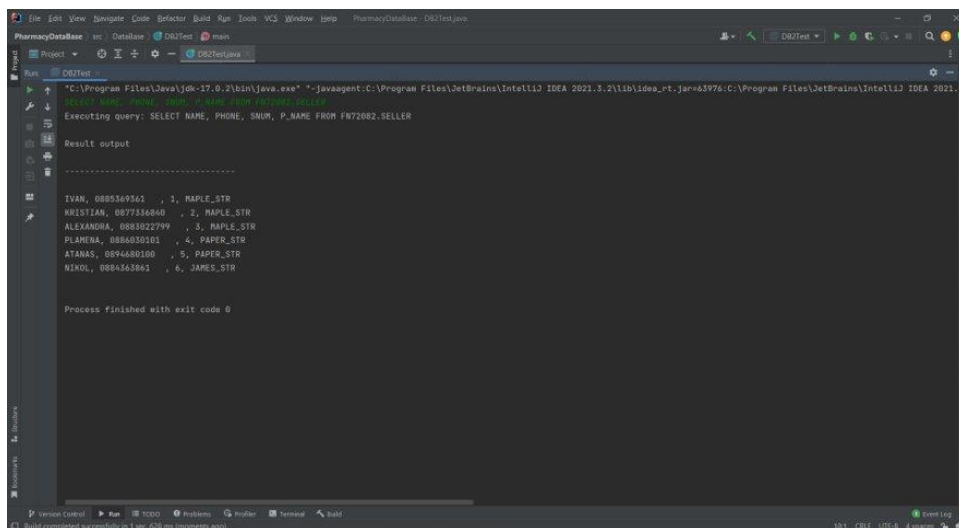
ФЗ-4: name, city -> phone (Manager)

ФЗ-5: name -> address, manager\_name, city (Pharmacies)

ФЗ-6: pharmacy\_name, product\_name -> quantity (Sells)

За всички ФЗ на релациите е в сила, че в лявата част се намира суперключ за релацията, следователно всички релации са в НФБК. Не се налага да правим нищо допълнително.

## 10. Приложение за достъп до базата



```
File Edit View Navigate Code Refactor Build Run Tools VCS Window Help PharmacyDatabase DBTest.java
Project Database DBTest main
Run DBTest
"C:\Program Files\Java\jdk-17.0.2\bin\java.exe" "-javaagent:C:\Program Files\JetBrains\IntelliJ IDEA 2021.3.2\lib\idea_rt.jar=43976:C:\Program Files\JetBrains\IntelliJ IDEA 2021.3.2\bin" -Dfile.encoding=UTF-8
Executing query: SELECT NAME, PHONE, SNUM, P.NAME FROM FN72082.SELLER
Result output
-----
IVAN, 0885369361 , 1, MAPLE_STR
KRISTIAN, 0877556840 , 2, MAPLE_STR
ALEXANDRA, 0883022799 , 3, MAPLE_STR
PLAMENA, 0884830381 , 4, PAPER_STR
ATANAS, 0894480100 , 5, PAPER_STR
NIKOL, 0884361862 , 6, JAMES_STR
Process finished with exit code 0
Build completed successfully in 1 sec, 620 ms (previous: 490 ms)
```

```
INSERT INTO FN72082.PRODUCTS(PRICE, NAME, FIRM) VALUES (10, 'KLACID', 'MYLAN')
Successfully inserted!

Process finished with exit code 0
```

	PRICE	NAME	FIRM
1	5.99	PARACETAMOL	SOPHARMA
2	6.99	GENTAMICIN	ACTAVIS
3	10.99	AZAX	NOBEL
4	7.91	VITAMIX	FORTEX
5	20	ZINAT	SOPHARMA
6	10	KLACID	MYLAN

```
DELETE FROM FN72082.PRODUCTS WHERE NAME = 'KLACID'
Successfully deleted!

Process finished with exit code 0
```

	PRICE	NAME	FIRM
1	5.99	PARACETAMOL	SOPHARMA
2	6.99	GENTAMICIN	ACTAVIS
3	10.99	AZAX	NOBEL
4	7.91	VITAMIX	FORTEX
5	20	ZINAT	SOPHARMA

## 11. Изгледи

```
CREATE VIEW MEDICINE_NO_PRESC
AS SELECT NAME, FORM FROM FN72082.MEDICINE WHERE PRESCRIPTION LIKE 'FALSE';

CREATE VIEW SELLER_PHONE
AS SELECT PHONE FROM FN72082.SELLER;

CREATE VIEW PRODUCT_UNDER_10_LEVA
AS SELECT NAME, FIRM FROM FN72082.PRODUCTS
WHERE PRICE < 10;
```

## 12. Източници

<https://sopharmacy.bg>