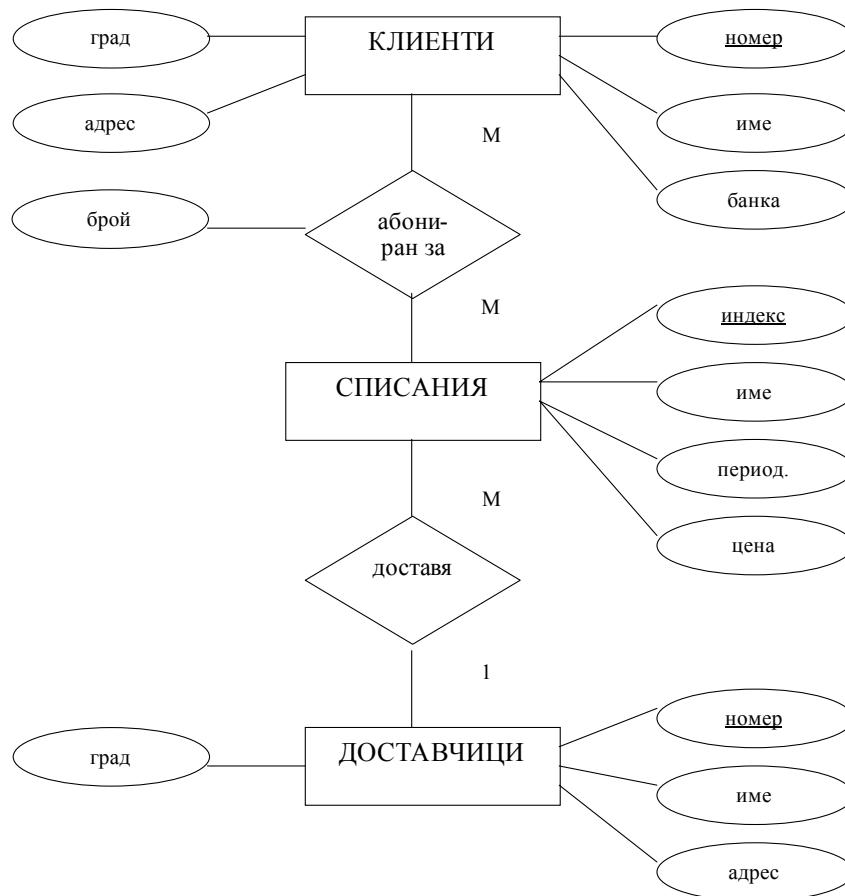


## 7. Проектиране на релационна база данни

При проектиране на БД трябва да изберем релациите (таблиците) и техните атрибути, които най-точно и безпроблемно да представят предметната област. За една определена предметна област може да има различни релационни модели. Някои от вариантите са по-добри от други. Защо проекта на БД може да е лош?

Ще разгледаме един лош вариант за релационен модел на БД и проблемите, които той създава. Проектираме БД-абонаменти, която съхранява информация за абонаментите за периодични издания. ER диаграмата е следната:



Съответният релационен модел, който е построен е:

CLIENT (сно, cname, bank, ccity, caddr)

AB (сно, jno, jname, jprice, jper, dno, dname, dcity, daddr, cnt)

Първичен ключ в CLIENT е сно, а за AB първичният ключ е съставен - {сно, jno}. В таблицата AB колоната сно е външен ключ, съответстващ на CLIENT.сно. Какви проблеми възникват при такава структура на БД.

- Излишество на данните. Наименованието, цената, периодичността и доставчика на списанието се повтарят за всеки абонамент, т.е. съхраняват се многократно в БД.
- Проблеми при изменение. Вследствие на излишеството съществува опасност за противоречивост на данните. Може да се измени наименованието, цената или периодичността на списание в едни редове, а в други не.
- Проблеми при добавяне. Не може да се добави информация за списание, ако за него няма нито един абонамент.

- Проблеми при изтриване. Ако в някакъв момент се изтрият всички редове за абонаментите на определено списание (всички клиенти са се отказали), то губим и информацията за списанието.

Казано накратко, не можем да съхраняваме информация за списание, ако за него няма поне един абонамент.

Тези проблеми ще изчезнат ако заменим таблицата АВ с няколко подходящи таблици. Процесът, при който една таблица се разделя на няколко, при което се премахват недостатъците в проекта, се нарича **нормализация** чрез декомпозиция. Код определи три нива на нормализация, които нарече първа, втора и трета нормална форма (1НФ, 2НФ, 3НФ). Целта на теорията на нормализация е да се създават "чисти" проекти на БД, т.е. проекти, в които всеки факт се съхранява веднаж. Такива проекти са добри, защото са:

- лесни за разбиране (това означава простота на модела);
- лесни за разширяване (т.е. точност на представяне на ПО);
- не създават проблеми при използване.

Правилата за проектиране на релационна БД на основата на ER модел, формулирани при разглеждането на релационния модел, имаха същата цел - да се получи "чист" модел.

### 7.1. Функционални зависимости

Нека  $R\{A_1, A_2, \dots, A_n\}$  е релация, а  $X$  и  $Y$  са подмножества от атрибути на  $R$ . Казваме, че  $X$  функционално определя  $Y$  (или  $Y$  функционално зависи от  $X$ ) и го означаваме  $X \rightarrow Y$ , ако в  $R$  не може да има два или повече реда, съдържащи еднакви значения за  $X$  и различни за  $Y$ , и това е вярно във всеки момент от съществуването на  $R$ .

Ако  $R_K$  е първичен ключ на релацията  $R$ , то са в сила функционалните зависимости  $R_K \rightarrow X$ , където  $X$  е произволно подмножество от атрибути на  $R$ . Това твърдение следва от дефиницията на първичен ключ. Същото твърдение естествено се отнася и за всеки възможен ключ на  $R$ .

Функционална зависимост (ФЗ) е понятие, което е свързано със семантиката на данните. ФЗ в определена релация се определя чрез анализ на предметната област, тъй като тя е предположение за смисъла на нещата в реалния свят.

Например в БД-абонамент в релациите АВ и CLIENT са в сила ФЗ:

```
jno->jname, jno->jper, jno->jprice, jno->dno,
dno->dname, dno->dcity, dno->daddr, {jno,cno}->cnt
cno->cname, cno->bank, cno->ccity, cno->caddr
```

Тези функционални зависимости означават, че:

- Индексът на списанието го идентифицира.
- Всяко списание има само един доставчик.
- Всеки доставчик се идентифицира с номер.
- Всеки клиент има уникален номер.

Нека  $F$  е множество ФЗ за релацията  $R$ . Една ФЗ  $X \rightarrow Y$  се нарича логическо следствие от  $F$ , ако всяко значение на  $R$ , което удовлетворява функционалните зависимости в  $F$  удовлетворява и  $X \rightarrow Y$ . У. Армстронг формулира правилата, чрез които могат да се изведат всички ФЗ, които са логическо следствие от определено  $F$ . Това множество се нарича обвивка на  $F$  и се обозначава с  $F^+$ . Правилата са известни като аксиоми за извод на Армстронг.

#### Аксиоми на Армстронг

Нека  $x, y, z$  и  $w$  са атрибути на релацията  $R$  (може и съставни).

- A1 (рефлексивност). Ако  $y \subseteq x$ , то  $x \rightarrow y$ . (тривиална ФЗ)
- A2 (попълнение). Ако  $x \rightarrow y$ , то  $\{x, z\} \rightarrow \{y, z\}$ .
- A3 (транзитивност). Ако  $x \rightarrow y$  и  $y \rightarrow z$ , то  $x \rightarrow z$ .
- A4 (обединение). Ако  $x \rightarrow y$  и  $x \rightarrow z$ , то  $x \rightarrow \{y, z\}$ .
- A5 (псевдотранзитивност). Ако  $x \rightarrow y$  и  $\{w, y\} \rightarrow z$ , то  $\{w, x\} \rightarrow z$ .
- A6 (декомпозиция). Ако  $x \rightarrow y$  и  $z \subseteq y$ , то  $x \rightarrow z$ .

## 7.2. Нормални форми

Ако е в сила ФЗ  $x \rightarrow y$  и  $y$  не зависи от никое подмножество на  $x$ , то казваме, че  $y$  е в **пълна ФЗ** от  $x$ .

Ако са в сила функционалните зависимости  $x \rightarrow y$  и  $y \rightarrow z$ , но не е в сила  $x \rightarrow z$ , то казваме че  $x \rightarrow z$  е **транзитивна ФЗ**.

Атрибут на релация, участващ в първичен или възможен ключ, ще наричаме първичен атрибут, а такъв, който не участва - непървичен атрибут.

### Първа нормална форма (1НФ)

Всички релации в релационния модел са в 1НФ. Изискването за 1НФ е значенията на всички атрибути да са атомарни (неделими), т.е. релацията да е нормализирана.

### Втора нормална форма (2НФ)

Релацията  $R$  се намира в 2НФ, ако е в 1НФ и всеки непървичен атрибут (атрибут, който не участва в първичен или възможен ключ) е в пълна ФЗ от първичния и от всеки възможен ключ на  $R$ .

Казано по друг начин в  $R$  няма ФЗ на непървичен атрибут от подмножество на първичен или възможен ключ. Следователно ако първичният и всички възможни ключове се състоят от един атрибут, то релацията е и в 2НФ.

В релация АВ са в сила ФЗ:

```
{jno, cno}->jname, {jno, cno}->jper, {jno, cno}->jprice, {jno, cno}->dno,
{jno, cno}->dname, {jno, cno}->dcity, {jno, cno}->daddr, {jno, cno}->cnt
jno->jname, jno->jper, jno->jprice, jno->dno,
jno->dname, jno->daddr, jno->dcity
```

което означава непълна ФЗ на  $jname, jper, jprice, dno, dname, dcity, daddr$  от първичния ключ  $\{jno, cno\}$ . Това е релация, която е в 1НФ, но не е в 2НФ. Ще я приведем в 2НФ, като я декомпозираме до две релации:

J (**jno**, jname, jprice, jper, dno, dname, dcity, daddr)

CJ (**cno, jno**, cnt)

Релацията CLIENT е в 2НФ защото първичният ключ е от един атрибут. Така получихме нов модел, включващ само релации в 2НФ - CLIENT, J, CJ.

### Трета нормална форма (3НФ)

Релацията  $R$  се намира в 3НФ, ако е в 2НФ и всеки непървичен атрибут зависи нетранзитивно от първичния и от всеки възможен ключ на  $R$ .

Релацията J е в 2НФ, но не е в 3НФ защото са в сила ФЗ:

```
jno->dno, dno->dname, dno->dcity, dno->daddr
```

и не е в сила  $dno \rightarrow jno$ . Следователно, следните ФЗ са транзитивни.

```
jno->dname, jno->dcity, jno->daddr
```

Декомпозираме J до JOURNAL и DOST така, че да премахнем транзитивните ФЗ.

JOURNAL (jno, jname, jprice, jper, dno)

DOST (dno, dname, dcity, daddr)

Така получихме нов модел, включващ само релации в 3НФ - CLIENT, JOURNAL, DOST, CJ.

Не се ли губи информация при това преобразуване на релационния модел? Ще можем ли от CLIENT, JOURNAL, DOST, CJ да получим същата информация както и от началния модел CLIENT, АВ.

**Дефиниция.** Нека R е релация с множество от ФЗ F. Казваме, че декомпозицията на R до R1 и R2 притежава свойството **съединение без загуба** относно F, ако е изпълнено твърдението:

$$R \equiv R[R1] \text{ join } R[R2]$$

Ако проектираме R по множествата атрибути на R1 и R2 и след това съединим двете проекции, отново ще получим началната релация R. Това е вярно във всеки момент от съществуването на R, т.е. за всяко нейно тяло. Това означава, че при декомпозицията не се губи информация.

Дори при новата структура, в нашите примери на декомпозиция, може да се съхранява повече информация. В общия случай е вярно:

$$(R1 \text{ join } R2) [R1] \subseteq R1$$

$$(R1 \text{ join } R2) [R2] \subseteq R2$$

Ако съединим R1 и R2 и след това проектираме съединението по R1 може да получим множество от редове, което е подмножество на тялото на R1.

**Теорема.** Декомпозицията на R до R1 и R2 притежава свойството съединение без загуба относно F, тогава и само тогава когато една от следните две ФЗ е следствие от F.

$$R1 \cap R2 \rightarrow R1 - R2$$

$$R1 \cap R2 \rightarrow R2 - R1$$

Друго важно свойство на една декомпозиция е да **запазва ФЗ**. Не се ли губят ФЗ при преобразуването на модела?

**Дефиниция.** Нека R е релация с множество от ФЗ F, която е декомпозирана до R1 и R2. Нека F1 е множеството ФЗ, които са в сила в R1, а F2 – в R2. Казваме, че декомпозицията на R **запазва ФЗ**, ако всяка ФЗ от F е логическо следствие на ФЗ от F1 и F2, т.е. в сила е твърдението:

$$F^+ \equiv (F1 \cup F2)^+$$

## Правила за преобразуване в 2НФ и 3НФ

### 1. От 1НФ в 2НФ

Нека в R (K1, K2, X, Y) {K1, K2} е първичен ключ и са в сила ФЗ

$$\{K1, K2\} \rightarrow X, K1 \rightarrow X, \{K1, K2\} \rightarrow Y$$

Декомпозираме R до R1 (K1, K2, Y) и R2 (K1, X).

Според теоремата тази декомпозиция на R до R1 и R2 притежава свойството съединение без загуба, защото  $K1 \rightarrow X$ , т.е.  $R1 \cap R2 \rightarrow R2 - R1$ .

### 2. От 2НФ в 3НФ

Нека в R (K, X, Y, Z) K е първичен ключ и са в сила ФЗ

$$K \rightarrow X, X \rightarrow Y, K \rightarrow Y, K \rightarrow Z, \text{ т.е. } K \rightarrow Y \text{ е транзитивна.}$$

Декомпозираме R до R1 (K, X, Z) и R2 (X, Y)

И тази декомпозиция притежава свойството съединение без загуба, защото  $X \rightarrow Y$ , т.е.  $R1 \cap R2 \rightarrow R2 - R1$ .

### Нормална форма на Бойс-Код (НФБК)

Релацията  $R$  се намира в НФБК, ако е изпълнено условието: винаги когато е в сила пълната ФЗ  $X \rightarrow A$  и  $A \notin X$ , то  $X$  е първичен или възможен ключ.

Ако  $R$  се намира в НФБК, то тя е и в ЗНФ, но обратното не е вярно. Ако в  $R$  има два пресичащи се възможни ключа, то  $R$  може да е в ЗНФ и да не е в НФБК.

Примери:

$R(\underline{\text{town}}, \underline{\text{addr}}, \text{pcode})$

В  $R$  са в сила функционалните зависимости:

$\{\text{town}, \text{addr}\} \rightarrow \text{pcode}$ ,  $\{\text{addr}, \text{pcode}\} \rightarrow \text{town}$ ,  $\text{pcode} \rightarrow \text{town}$

Първичен ключ е  $\{\text{town}, \text{addr}\}$ , а  $\{\text{addr}, \text{pcode}\}$  е възможен ключ. Всички атрибути са първични, следователно  $R$  е в ЗНФ. Но  $R$  не е в НФБК, защото  $\text{pcode}$  не е възможен ключ.

$\text{emp\_project}(\underline{\text{pno}}, \underline{\text{eno}}, \text{pname}, \text{ptime})$

Това е друг вариант на релацията от БД-служители, вместо релациите  $\text{project}$  и  $\text{emp\_pro}$ , ако всеки проект има уникален номер и наименование, но няма описание.

Първичен ключ в релацията е  $\{\text{pno}, \text{eno}\}$ , а  $\{\text{pname}, \text{eno}\}$  е възможен ключ. В сила са функционалните зависимости:

$\{\text{pno}, \text{eno}\} \rightarrow \text{ptime}$ ,  $\{\text{pname}, \text{eno}\} \rightarrow \text{ptime}$ ,  $\text{pno} \rightarrow \text{pname}$ ,  $\text{pname} \rightarrow \text{pno}$

Единственият непървичен атрибут  $\text{ptime}$  е в пълна ФЗ първичния и възможния ключ, следователно  $\text{emp\_project}$  е в ЗНФ, но не е в НФБК заради ФЗ  $\text{pno} \rightarrow \text{pname}$  и  $\text{pname} \rightarrow \text{pno}$ .