3. Релационен модел на данните

3.1. Понятия в релационния модел

Релационният модел е предложен от Е. Код през 1970, когато излиза историческата му статия в CACM, която поставя нов период в областта на базите данни. Този модел има две важни характеристики, които го издигат до положението на основен подход в съвременната технология на БД, а именно:

- прост и естествен начин за представяне на данните – таблици.
- строга математическа основа на модела – теория на множествата и предикатната логика.

Нека разгледаме пример на две таблици, съдържащи съответно данни за отделите и служителите в предприятие.

dep

dno	**dname**	**budget**
1	Административен	2000
2	ФСО	1000
3	Продажби	5000

emp

eno	**ename**	**salary**	**dno**
100	Иванов	500	1
120	Петрова	150	1
200	Антонов	300	2
210	Георгиева	200	2

Всяка таблица се състои от редове и колони, като значенията във всяка колона са сравними, т.е. принадлежат на едно множество от допустими значения. Обикновено всяка колона има наименование, което показва какъв е смисъла на значенията в колоната. Сега да разгледаме формалните понятия в релационния модел, които приблизително съответстват на неформалните понятия.

Неформални понятия	Формални понятия
table / таблица	relation / релация
row / ред	tuple / n-орка / ред
column / колона	attribute / атрибут
множество от допустими значения	domain / домен

Домен е множество от атомарни (неделими) значения от един и същи тип. Атомарни означава, че те са най-малката семантична единица данни за модела, т.е. ако разложим значението, то ще загуби смисъла си. Напр., множеството от всички имена на хора, всички имена на отдели и т.н.

Релация R над домените D1, D2, … Dn се състои от:

- Заглавие – фиксирано множество от имена на атрибути A1, A2, … An, такива че Ai съответства на домена Di, за i=1,2,…n.

- Тяло – изменящо се във времето множество от редове (tuples, n-торки).

Ред е множество от двойки от вида <име-на-атрибут: значение-на-атрибут>, т.е. { <A1:v_{i1}>, <A2: v_{i2}>,… <An: v_{in}>} където v_{ij} принадлежи на домена Dj за j=1,…n. Броят n на атрибутите на релацията се нарича **степен на релацията**. Броят на редовете на релацията се нарича **кардинално число**.

Свойства на релациите, които са следствие от тази дефиниция:

- В релациите няма еднакви редове.
- Редовете в една релация са ненаредени.
- Атрибутите в релацията са ненаредени.
- Стойността на всеки атрибут е атомарна (неделима).

Релация, която притежава последното свойство, се нарича нормализирана. Релационият модел използва само нормализирани релации, за разлика от математическите релации, които не е задължително да са нормализирани.

Много често таблицата се използва като изображение на абстрактното понятие релация. При това представяне се внасят някои свойства, които не са верни според формалната дефиниция. В таблицата редовете са наредени отгоре надолу, също както и колоните – отляво надясно. Но когато говорим неформално понятията релация и таблица са синоними, както и понятията атрибут и колона. Това е едно от неоспоримите преимущества на релационния модел – едно формално понятие има просто неформално представяне. Още повече, че в езика SQL се използват неформалните понятия – таблица, колона, ред.

Възможен и първичен ключ

Нека R{A1, A2, … An} е релация, а K⊆{A1, A2,…An}, т.е. K е подмножество от атрибути на R. Ще казваме, че K е възможен ключ (candidate key) на R, ако притежава свойствата:

- уникалност – не съществуват два реда на R, които имат еднакви значения за всички атрибути от K;
- без излишество – никое подмножество на K не притежава свойството уникалност.

Всяка релация има поне един възможен ключ. Ако K се състои от един атрибут го наричаме прост, в противен случай - съставен. Един от възможните ключове се избира за първичен ключ (primary key), а останалите ще наричаме алтернативни ключове (alternate keys) или възможни ключове.

Първичният ключ осигурява инструмент за адресиране на редове в релационния модел.

Външен ключ

Нека R1{PK, A1, … } е релация с първичен ключ PK и R2{B1, … FK, …} е релация. FK ще наричаме външен ключ (Foreign key) на R2 ако съответства на първичния ключ на R1, което означава следното:

- FK и PK са сравними атрибути – включват еднакъв брой атрибути и съответните атрибути са определени са над един и същи домен;
- Всяко значение на FK в ред от R2 съвпада с някое значение на PK в ред от R1.

Чрез външен ключ се създават връзки между редове в различни релации, т.е. външният ключ е нещо като указател, който сочи към ред от друга релация. Всъщност R1 и R2 не е задължително да са различни релации.

Пример. Нека релациите са:

```
dep{dno, dname, budget}
emp{eno, ename, job, sal, addr, dno}
```

Атрибутът dno в релацията emp е външен ключ, който съответства на първичния ключ на релацията dep. Релацията emp, съдържаща външния ключ, се нарича referencing relation, а релацията dep, съдържаща съответния първичен ключ, се нарича referenced relation.

Релационна БД (РБД) е съвкупност от изменящи се във времето релации от различна степен.

3.2. Общи правила за цялостност

Във всеки момент БД съдържа данни, които моделират ПО. Следователно описанието на БД трябва да включва правила за цялостност, които да информират СУБД за различни ограничения в ПО и по този начин да се предотвратява въвеждането в БД на неверни данни. Има правила за цялостност, които са специфични за конкретната ПО и се описват чрез ограничения за цялостност, но има две общи правила, които се прилагат към всяка БД.

Entity integrity

Това правило е свързано с понятието първичен ключ и с едно специално значение, наречено `NULL`, което се поддържа от почти всички релационни СУБД. Това значение може да бъде приемано от всеки тип данни и означава неизвестно или неприложимо значение на атрибута в съответния ред. С използване на значение `NULL` в БД се решава проблема с отсъстващата информация, нещо което се среща често в реалния живот. Правилото за цялостност гласи: първичният ключ е уникален и не може никога да е `NULL`. Това е така, защото първичният ключ е уникален идентификатор на реда и в БД не може да се въвежда информация, която не може да бъде идентифицирана.

Referential integrity

Всяко значение на външен ключ трябва или да е напълно `NULL` или да е равно на някое значение на съответния му първичен ключ. Понякога е необходимо да се разреши външният ключ да приема значение `NULL`. Напр., ако определен служител не е към никой отдел, то атрибутът dno на релацията emp в съответния ред трябва да е `NULL`. В други случаи външният ключ не може да приема значение `NULL`.

Всяко състояние на БД, което не удоволетворява тези правила е некоректно и следователно не бива да се допуска. Но как да се поддържа БД в коректно състояние? Това не се казва в правилата за цялостност.

Подържането на ограничението за първичния ключ е сравнително ясно и лесно. А именно всяка операция за добавяне или изменение на ред, при която първичният ключ ще стане `NULL`, трябва да бъде отхвърлена, т.е. да не се изпълни.

Поддържането на ограничението за външен ключ е по-сложно, тъй като то засяга две релации. За всеки външен ключ трябва да се отговори на следните въпроси:

1. Може ли външният ключ да приема значение `NULL` или не?

2. Какво да се прави при опит да се изтрие ред, към който сочи външен ключ? Възможни отговори на втория въпрос са:

Restricted – операцията изтриване се разрешава само, ако с изтривания ред няма свързани редове със същото значение на външния ключ.

Cascades – каскадно изпълнение на операцията изтриване и за всички редове, съдържащи съответното значение на външния ключ.

Set null / nullifies – операцията изтриване се изпълнява, като в съответните редове на външния ключ се дава значение `NULL`.

3. Какво да се прави при опит да се измени значението на първичен ключ, към който сочи външен ключ?

И на този въпрос възможните отговори са същите: Restricted, Cascades, Set null.

3.3. Проектиране на РБД на основата на ER модел

След като е проектиран ER модела на ПО трябва да се проектира релационния модел, т.е. трябва да се изберат релациите и техните атрибути, които ще представят класовете обекти, връзки и техните свойства. Ще формулираме няколко полезни правила.

1. Независим клас обекти се представя чрез релация, атрибутите на която представят свойствата на обектите. Избира се най-подходящият първичен ключ.

2. Слаб клас обекти се представя чрез релация, в която освен атрибутите за свойствата на класа се включва и атрибут(и), идентифициращ обекта от когото зависи слабия обект. Този атрибут е външен ключ и може да е част от първичния ключ на релацията.

3. Клас връзки M:M между класовете обекти E1 и E2 се представя чрез отделна релация, в която има два външни ключа, съответстващи на първичните ключове на релациите за E1 и E2. Най-често първичният ключ в тази релация е комбинацията от двата външни ключа. Ако връзката има свойства за тях се добавят допълнителни атрибути в същата релация.

4. Клас връзки 1:M между класовете обекти E1 и E2 може да се представи както връзка M:M, но има и по-икономичен начин - чрез външен ключ в релацията за E2, който съответства на първичния ключ на релацията за E1. Ако връзката има свойства за тях се добавят допълнителни атрибути в същата релация. Връзка 1:1 може да се разглежда като частен случай на връзка 1:M, т.е. външният ключ може да е във всяка от двете релации.

На Рис.4 е представен релационния модел на БД-служители. Атрибутите, които са първични ключове на релациите са подчертани, а връзката външен първичен ключ е представена чрез стрелка.

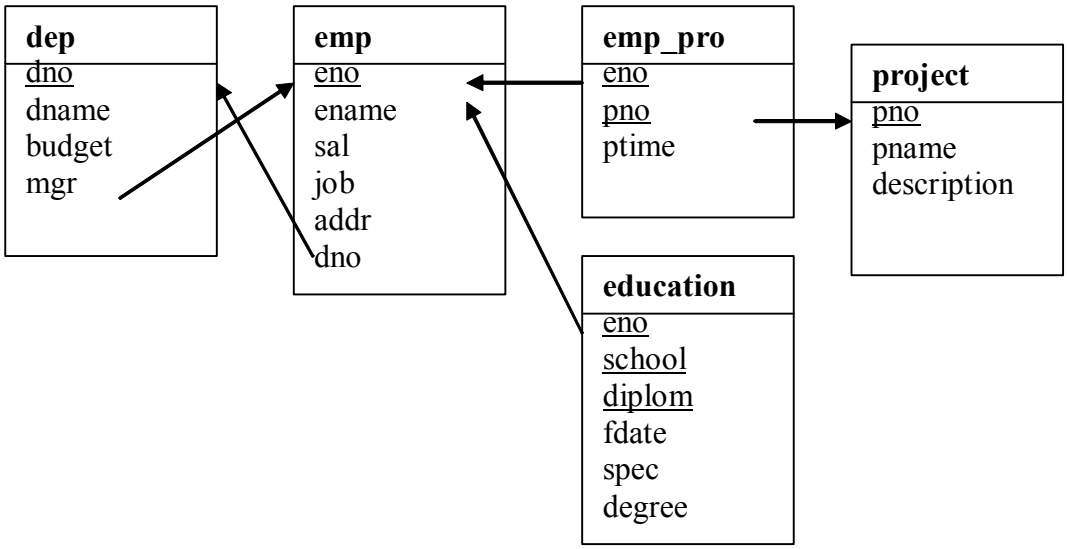

Рис.4. Релационен модел на БД-служители