

7. Концепция за ключове

Лекционен курс "Бази от данни"

Въведение

Това е компонентът на реляционния модел, който през последните години е най-много променян.

Забележки:

- Във всеки момент всяка БД съдържа определена конфигурация от стойности на данните - тази конфигурация отразява (е модел на) една реална ситуация;
- Определени конфигурации нямат реален смисъл - те не представляват някакво възможно състояние на реалния свят;

- Дефиницията на БД трябва да бъде разширена - тя трябва да включва правила за цялостност;
- Тяхната цел е да информират СУБД за някои ограничения в реалния свят - напр., теглото не може да бъде отрицателно число;
- Така могат да се избегнат някои конфигурации, които нямат смисъл в реалния свят.

БД са субект на многобройни правила за цялостност – обикновено те са зависими от приложната област на БД.

Пример за таблица с книги от библиотека:

- Не е реално цената да е отрицателно число - т.е. стойността ѝ трябва да е по-голяма от нула;
- Годишната на публикуване не може да е по-голяма от текущата година;
- ISBN номерът не се формира по произволен начин – генерира се по определени международни правила и е уникален за всяка книга.

Някои правила за цялостност са специфични за точно определена БД.

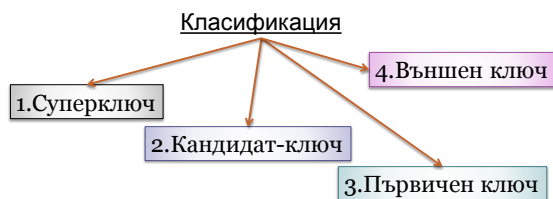
Освен специфични правила за цялостност релационният модел включва два типа **общовалидни** правила за цялостност, валидни за **всяка** релационна БД:

- Концепцията за първични ключове
- Концепцията за външни ключове

Концепция за ключове

Ключ е атрибут или група от атрибути, който се използва за идентификация на n-торка (ред) в релация.

Ключовете могат да бъдат общо класифицирани на (1)Суперключ, (2)Кандидат-ключ, (3)Първичен ключ и (4)Външен ключ:



Неформални определения

1. Суперключ

Подмножество от атрибути на релация, което уникално идентифицира всяка п-торка от тялото ѝ.

Суперключовете представят ограничение, което предпазва от това две п-торки да имат съвпадащи стойности за тези атрибути.

Неформални определения

2. Кандидат-ключ

Кандидат-ключът е минимален суперключ.

Кандидат-ключ за дадена релация е минималното множество от атрибути, чиито стойности уникално идентифицират п-торка в тази релация.

Неформални определения

3. Първичен ключ

Първичният ключ е конкретно избран кандидат-ключ за идентификация на редовете в една релация.

Атрибутите, съставлящи първичен ключ задължително трябва да имат стойности.

Неформални определения

4. Външен ключ

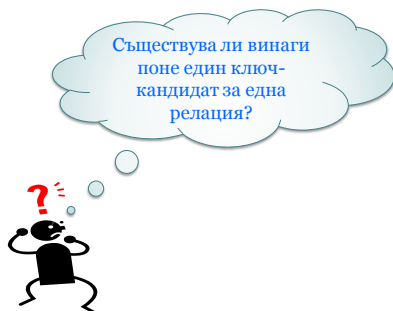
Подмножество от атрибути на дадена релация, използвано да "отнася" (референцира) п-торка от релацията към п-торка от друга релация.

Ключ-кандидат - дефиниция

Нека R е релация с атрибути $\{A_1, A_2, \dots, A_n\}$. Множеството на атрибутите $K = (A_i, A_j, \dots, A_k)$ от R е ключ-кандидат (candidate key) на R , ако то удовлетворява следните две, независещи от времето условия:

- **уникалност (uniqueness)** - в един и същ момент не съществуват два различни записа на R с еднаква стойност за A_i , за A_j, \dots , за A_k ;
- **достатъчност (minimality)** - никой от A_i, A_j, \dots, A_k не може да бъде премахнат, без това да наруши уникалността.

Основен въпрос



Коментар

- Всяка релация има най-малко един ключ-кандидат - понеже релациите не съдържат дублирани п-торки (редове);
- Понеже п-торките са уникални, следва че най-малко комбинацията от всички атрибути на релацията притежава свойството уникалност;
- В практиката много релации имат повече от един ключ-кандидат. За дадена релация разработчиците избират един от ключовете-кандидати за първичен и всички останали стават (ако съществуват) алтернативни ключове.

- Дефиницията на ключ-кандидат се прилага към стойността на релацията - не към релационната променлива.



Дефиницията на кандидат-ключ може да бъде прецизирана:

Нека R е релационна променлива - тогава един ключ-кандидат за $R \in K = \{A_1^R, A_2^R, \dots, A_n^R\}$, така че за **всеки момент** той притежава свойствата:

- уникалност - няма две п-торки в текущата стойност на R , които да имат една и съща стойност за K ;
- достатъчност - всички S подмножества на K не притежават свойството уникалност.

Спецификация на ключ-кандидат

```
candidate-key-definition ::=
  CANDIDATE KEY (attribute-list) |
  PRIMARY KEY (attribute-list)
```

Пояснения

Въпреки че на практика повечето релации имат само един ключ-кандидат, то е възможно да имат и повече от един.

Например релацията ELEMENTS, представляваща таблицата на химическите елементи. Всеки химичен елемент има уникални:

- име
- символ
- атомарен номер

```
CREATE BASE RELATION ELEMENTS
  ( NAME .... ,
    SYMBOL .... ,
    NUMBER .... ,
    ... )
  CANDIDATE KEY (NAME)
  CANDIDATE KEY (SYMBOL)
  CANDIDATE KEY (NUMBER)
```

Пояснения

Дефинирахме ключовете-кандидати като множества от атрибути.

Един ключ-кандидат, който се състои от повече от един атрибут, се нарича съставен.

С един атрибут – прост.

Пояснения

За понятието **достатъчност** - напр., дефинираме комбинацията {ISBN, YEAR_PUB} вместо само ISBN за ключ-кандидат:

- Тогава системата не може да приложи ограничението, че номерът на книгата е уникален в "глобален" смисъл, тя го интерпретира като уникален в "локален" смисъл, т.е. за конкретната година на публикуване;
- Следователно верните ключове-кандидати не трябва да включват атрибути, които не са съществени за целите на уникалната идентификация.

Пояснения

Има основателна причина за кандидат-ключовете да изискваме **несъкратимост**: ако един външен ключ референцира съкратим кандидат-ключ, това ще означава, че и самият той е съкратим, а релацията, към която принадлежи, ще е в конфликт с принципите за нормализацията, защото ще притежава излишество.

Дефиницията на суперключ може да се изведе от тази на КК чрез премахване на изискването за несъкратимост.

Пояснения

Не трябва да се смесва понятието за ключ-кандидат с това за физически уникален индекс - понеже последният често се използва за реализация на първия.

Защо са важни ключовете-кандидати?

- Те поддържат единствения адресен механизъм на ниво записи в релационния модел – единственият гарантиран от системата начин за локализиране на точно един конкретен запис е посредством комбинацията (R, k), където R е името на релацията, съдържаща записа, а k е стойността на ключа-кандидат;
- Стойностите на ключа-кандидат могат да се използват и в други релации от БД, където да служат като референции към записите, които се идентифицират чрез тези стойности.

Пример

REGION_ID уникално идентифицира географски регион

| REGION_ID | REGION_NAME | POPULATION |
|-----------|------------------------|---------------|
| 1 | Eastern Europe | 500 000 000 |
| 2 | Americas | 1 000 000 000 |
| 3 | Asia | 2 500 000 000 |
| 4 | Middle East and Africa | 1 000 000 000 |
| 5 | Western Europe | 500 000 000 |

Пример

Ако в релацията REGIONS атрибутът REGION_ID е ключ-кандидат, то изразът:

REGIONS where REGION_ID = 1

гарантирано ще върне най-много един запис.

Ако REGION_NAME е също ключ-кандидат за REGIONS, изразът

REGIONS where REGION_NAME = 'Asia'

също ще върне най-много един запис.

От друга страна, изразът

REGIONS where POPULATION = 1000000000

би върнал непредсказуем брой записи в общия случай.

Следователно, **ключовете-кандидати** са абсолютно основополагащи за функционирането на цялостния релационен модел така, както **адресите** на оперативната памет са важни за успешното изпълнение на операциите в компютъра.

Първични и алтернативни ключове

- Както видяхме, възможно е една базова релация да има повече от един ключ-кандидат;
- В такъв случай релационният модел изисква един от тях да бъде обявен (избран) за първичен ключ (ПК);
- Останалите се наричат алтернативни ключове (АК);
- Кой ключ ще бъде избран зависи от прагматични причини.



Пример

Ако в релацията ELEMENTS изберем “символ” за първичен ключ, то “име” и “атомарен номер” ще бъдат алтернативни ключове.
