

6. Лекция

Category	Empty
Files	Empty
Created	May 28, 2023 11:22 AM
Reminder	Empty
Status	Open
URL	Empty
Updated	May 28, 2023 11:22 AM

Релационен модел

12 основни правила

- Цялата информация се представя чрез таблици, наречени **релации**
- Връзките между данните са също таблици

0. Правило

СУБД трябва да може да управлява БД изцяло през its relational capabilities.

1. Информационно правило

Информацията в БД се представя по единствен начин - чрез стойности подредени по колони и редове в таблици

Важно - имаме единен начин за представяне

2. Правило за гарантиран достъп

Гарантиран достъп - към данните се осъществява еднозначен достъп

- изискване за първичен ключ (PK) - всяка индивидуална стойност в БД да може логически да бъде адресирана чрез специфициране на името на таблицата, името на колоната и PK на реда, съдържащ стойността

Идентификацията става чрез посочване на първичния ключ, за който искаме да намерим стойностите по съответния атрибут.

Посочва се релацията, първичния ключ и за исканите стойности, посочваме името на колоната.

Името на релацията се посочва във From clause-ата

Условията се посочват в Where clause-ата

3. Систематична поддръжка на нулевите стойности

Всяка БД трябва да позволява някои атрибути да имат null стойности.

NULL - дадена стойност не е известна/налична

NOT NULL - отбелязваме дали даден атрибут може да приема null стойности.

СУБД трябва да позволява всяко поле да може да е нулево (или празно)

Queruing a DBMS

DBMS provides a Query language

- Data manipulation language (DML) as a part of SQL (Structured query language)

4. Активен онлайн каталог - описание на БД и нейното съдържание на логическо ниво - като таблица

Потребителите имат достъп до каталога със същият език, който използват за достъп до данните в БД.

5. Системата трябва да поддържа поне един език, който:

- се ползва и чрез приложни програми
- поддържа операции за **дефиниране** на данни, **манипулация** на данни, ограничения за сигурност и интегритет

Важно, защото по този начин работим/обработваме данните

6. Всички изгледи, които теоретично могат да бъдат обновявани, да се обновяват от ст-та

При изгледа ние не съхраняваме като отделен обект view-то с неговата дефиниция и данните, а съхраняваме само схемата на view-то (с неговите дефиниции).

View-то представлява само една дефиниция

7. High-level insert, update, delete

СУБД поддържа и вмъкване, обновяване, изтриване на ниво множество.

Тоест СУБД поддържа добавяне/обновяване/изтриване на множество редове

8. Physical data independence

Промени на физическото ниво не трябва да изискват промени в приложенията.

9. Logical data independence

Ако по някакви причини променим малко концептуалния модел (добавяне/премахване на атрибути/релации), то това не трябва да пречи на работата на старите приложения

(Слайд 7)

Физически модел - на физическите носители представяме тези данни

Концептуален модел -

Външни view-та - достигат до информацията

Това са правилата за логическа и физическа независимост

10. Integrity independence

Езикът на БД трябва да може да дефинира правила за цялост. Те трябва да бъдат записани в каталога и да не могат да бъдат пренебрегвани

- първичен и вътрешен ключ

11. Distribution Independence

Независимост на разпределението.

От гледна точка на потребители не би трябвало да има значение дали БД е разпределена или не (части от БД са на различни устройства)

12. The nonsubversion rule

Не трябва да има възможност да се заобиколят правилата за интегритет с помощта на езици от по-ниско ниво (от това на езика SQL - език от 4то ниво).

Примерно езици като C/C++, Java, Python, ...

Structured Query Language

Език за описание на схемата на БД и операциите върху таблици

- SELECT - само четем, не променяме данните

Data Definition Language - подезик за описание на схемата

Data Manipulation Language - подезик за работа с данните:

- Insert
- Update
- Delete

Data Control Language

Създаваме потребители, даваме и права за четене/промяна на данните ..

Функционални зависимости

- Значение на FD's
- Ключове и суперключове
- Аксиоми на Армстронг

Няколко атрибута определят еднозначно стойностите на друг(и) атрибут(и)

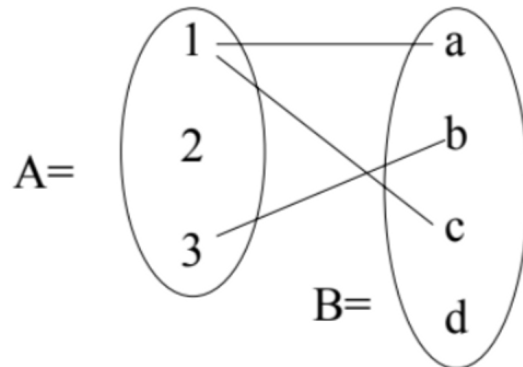
Зависимости в рамките на една релационна схема. Не са зависимости между отделните релации

Основни понятия в Релационен модел

Релация

Ако A , B са m -ва, релацията R е подмножество на $A \times B$

$A = \{1, 2, 3\}$, $B = \{a, b, c, d\}$,
 $R = \{(1, a), (1, c), (3, b)\}$



makes е подмножество на **Product** x **Company**

Нотация

$R(A_1, A_2, \dots, A_n)$

R - име на релация

A_1, A_2 - имена на атрибути

$r = \{t_1, t_2, \dots, t_m\}$ - екземпляр на релацията - всички кортежи в текущата релация (текущото състояние на релацията)

$t = \langle v_1, v_2, \dots, v_n \rangle$ - кортеж, който съдържа конкретни стойности

$t[A_1]$ or $t.A_1$ - A_1 е атрибут на релацията t

Имена на релации: Q, R, S

Екземпляри на релации: q, r, s

Кортежи (tuples): t, u, v

Всяка дума/индекс трябва да можем да обясним какво означава - на изпита

Функционални зависимости

Зависимости между атрибутите в една релация.

Не засягат зависимости между релации в цялата БД, а единствено зависимости между атрибутите в рамките на една релация

Група правила как от +1 пряко посочени функционални зависимости можем да получим нови такива, които са верни.

Няколко правила са т.ч. от тях могат да се изведат всички други. Те се наричат **Аксиоми на Армстронг**

💡 lightshot

Дефиниция на функционалната зависимост

$A_1, A_2, \dots, A_n \rightarrow B$

- Чете се: " A_1, A_2, \dots, A_n функционално определя B "

Ако два кортежа от $r(R)$ съвпадат по атрибутите A_1, A_2, \dots, A_n of R , те трябва да съвпадат и по атрибута B



Пример

Movies					
<i>title</i>	<i>year</i>	<i>length</i>	<i>filmType</i>	<i>studioName</i>	<i>starName</i>
Star Wars	1977	124	color	Fox	Carrie Fisher
Star Wars	1977	124	color	Fox	Mark Hamill
Star Wars	1977	124	color	Fox	Harrison Ford
Mighty Ducks	1991	104	color	Disney	Emilio Estevez
Wayne's World	1992	95	color	Paramount	Dana Carvey
Wayne's World	1992	95	color	Paramount	Mike Myers

title year → length

title year → filmType

title year → studioName

но не:

title year → starName

Функционална зависимост във всяка релация - primary key и ключовете на релацията

Функционалните зависимости изразяват семантични връзки в данните A, B, C, ... м-ва от атрибути

- Понякога за по-голяма яснота се използва AA или BB
- A1, A2, A3, ..., An означават индивидуални атрибути

F се използва за означение на м-то на функционалните зависимости

- С малки букви f означаваме **единични функционални зависимости**

FD е твърдение за схемата на релацията, не за конкретен екземпляр

- FD's не могат да се определят чрез просто преглеждане на данните
- FDs са свойства на семантиката на атрибутите
- Всички данни ги удовлетворяват

Най-основният пример за функционална зависимост е тази между ключа на релацията и всички атрибути в релацията.

Ключове на релации

$K = \{A_1, A_2, \dots, A_n\}$ е ключ за релацията R ако:

1. М-то K функционално определя всички атрибути на R.

2. За нито едно подмножество на K (1) не е вярно

Ако K удовлетворява (1), но не удовлетворява (2), то K е суперключ.

- За ключовете в E/R модела няма изискване за минималност

Алтернативна дефиниция

Нова дефиниция в термините на FD's

$K = \{A_1, A_2, \dots, A_n\}$ е ключ на релацията R ако:

1. K определя функционално ВСИЧКИ други атрибути на R
 - $K \rightarrow R$
 - Не е възможно 2 различни кортежа t и u да съвпадат по A_1, A_2, \dots, A_n
2. Нито едно подмножество на K не може да определи функционално всички останали атрибути на R
 - K е минимално

Какво е функционалното при FD

$A_1 A_2 \dots A_n \rightarrow B$ се нарича функционална зависимост, защото има **функция**, която на

списък от стойности (по една за всяко A_1, A_2, \dots, A_n) съпоставя уникална стойност за B

Тук функцията не се изчислява по стандартния начин

- "изчислението" става чрез търсене в релацията

Откриване на ключ в релации

Когато релационната схема е получена от преобразуването на E/R диаграма в релация, структурата на ключа може да се предвиди:

- Преобразуване на същност
 - Ако релацията е получена от m -во същности, ключът на релацията се формира от атрибутите на ключа на m -вото същности

- Преобразуване на бинарна връзка
 - Чрез включване на двата ключа на двете множества същности в релацията
 - Атрибутите на връзката не влизат в ключа на релацията

Определяне на всички FDs		
A	B	C
a	b	c
a	b1	c
a1	b	c1

$A \rightarrow C ?$

Не може да се определя функционална зависимост само по данните, трябва да мислим спрямо знанията си за атрибутите си, техния смисъл и връзки

Добри и лоши проекти
<ul style="list-style-type: none"> • Защо проектът не е добър? <div>Data(Id#, Name, Address, C#, Description, Grade)</div> <ul style="list-style-type: none"> • Защо трябва да предпочетем този проект? <div> Student(Id#, Name, Address) Course(C#, Description) Enrolled(Id#, C#, Grade) </div> <p>student-course database</p>

Slide 19

Причината второто да е по-добро от първото е, че можем да въведем информация за курсове без да е нужно някой вече да е взел този курс.

В долният вариант имаме само по една функционална зависимост - единствено между ключът и другите атрибути на релацията.

Пример за лошо проектиране

Id#	Name	Address	C#	Description	Grade
124	Jones	Phila	Phil7	Plato	A
456	Smith	NYC	Phil7	Plato	B
789	Brown	Boston	Math8	Topology	C
124	Jones	Phila	Math8	Topology	A
789	Brown	Boston	Eng12	Chaucer	B

- Излишество на информация
 - Name - Address
- Информацията за курса зависи от наличието на студент

Slide 20

Аксиоми на Армстронг

1. Рефлексивност

Ако $Y \subseteq X$ то $X \rightarrow Y$

Пример: Name, Address \rightarrow Address

► Доказателство

2. Разширение, попълнение (Augmentation)

Ако $X \rightarrow Y$ то $XW \rightarrow YW$

Пример : от $C\# \rightarrow Description$ получаваме

$C\#, Id\# \rightarrow Description, Id\#$

► Доказателство

3. Транзитивност

Ако $X \rightarrow Y$ и $Y \rightarrow Z$ то $X \rightarrow Z$

Пример : от $Id\#, C\# \rightarrow C\#$ и $C\# \rightarrow Description$,
получаваме $Id\#, C\# \rightarrow Description$

► Доказателство

Чрез тях могат да се докажат другите зависимости.

Чрез тях може да създаваме други зависимости

Следствия от Аксиомите на Армстронг

Обединение

Ако $X \rightarrow Y$ и $X \rightarrow Z$ то $X \rightarrow YZ$

$X \rightarrow Y$, следователно $X \rightarrow XY$ (A2)

$X \rightarrow Z$, следователно $XY \rightarrow ZY$ (A2)

$XY \rightarrow ZY$ (A3)

Псевдотранзитивност

Ако $X \rightarrow Y$ и $WY \rightarrow Z$ то $XW \rightarrow Z$

$X \rightarrow Y$, следователно $WX \rightarrow WY$ (A2)

но

$WY \rightarrow Z$, следователно $WX \rightarrow Z$ (A3)

Декомпозиция

Ако $X \rightarrow Y$ и $Z \subseteq Y$ то $X \rightarrow Z$

$X \rightarrow Y$

$Z \subseteq Y$, следователно, $Y \rightarrow Z$ (A1)

$X \rightarrow Z$ (A3)

Правила за разделяне и обединение

Имаме право да разделим множеството атрибути в дясната част на FD и да поставим всеки от тях в дясната част на нова FD.

Правило за декомпозиция:

Ако $AA \rightarrow B_1, B_2, \dots, B_n$, то

$AA \rightarrow B_1$

$AA \rightarrow B_2$

, ...,

$AA \rightarrow B_n$

Правило за обединение:

Ако $AA \rightarrow B_1$

$AA \rightarrow B_2$

, ...,

$A \rightarrow B_n$

то $A \rightarrow B_1, B_2, \dots, B_n$

Тривиални зависимости

Функционалната зависимост $A_1A_2\dots A_n \rightarrow B$ се нарича **тривиална**, ако атрибутът B съвпада с някой от атрибутите A_1, A_2, \dots, A_n .

В противен случай - **нетривиална**

Тривиални, нетривиални, напълно нетривиални

- Тривиална: атрибутите $B_1B_2\dots B_n$ са подмножество на $A_1A_2\dots A_n$
 - title year \rightarrow title
- Нетривиална: поне един атрибут от $B_1B_2\dots B_n$ не е подмножество на $A_1A_2\dots A_n$
 - title year \rightarrow year, length
- Напълно нетривиална: нито един от атрибутите B 's не е част от A 's

Това е от 04.04.2023:

Обвивка на множество от функционални зависимости

Нека F е мн-во от FD's, F^+ - обвивка на F е множеството от FD's, които логически следват от F .

(Първите 3 - аксиоми на Армстронг пораждат всички други)

$$F^+ = \{X \rightarrow Y \mid\}$$

Пример

$AB \rightarrow C, CD \rightarrow E$

$ABD \rightarrow E$

2 аксиома - разширение с D:

$ABD \rightarrow CD \rightarrow E$

От транзитивност - вярно

Пример

$A \rightarrow B, B \rightarrow C, A \rightarrow D, CE \rightarrow HG$

A определяя функц. A,B,C,D

$A \rightarrow BD$

$AE \rightarrow HG$

$BCD \rightarrow BCDE$