

21.1. Бази от данни: Релационен модел.

Релационен модел на данните: домейн; релация; кортеж; атрибути; схема на релация; схема на релационна база от данни; реализация на релационната база от данни; видове операции върху релационната база от данни; заявки към релационната база от данни. Релационна алгебра: основни (обединение; разлика; декартово произведение; проекция; селекция) и допълнителни (сечение; частно; съединение; естествено съединение) операции.

Релационен модел на данните: домейн; релация; кортеж; атрибути.

В основата на релационния модел стои математическото понятие n-членна релация. Всяка релация е множество от елементи, които се състоят от n компоненти и се наричат n-торки. Възшност чрез една релация се моделира даден клас от обекти, а всяка n-торка от релацията представлява конкретен обект от този клас.

Предимства на релационния модел пред другите модели на данни:

- Простота на модела – възможност за опериране с релациите както с числата, като се използват операциите на т. нар. релационна алгебра
- Висока степен на независимост на данните (няма необходимост от познаване на вътрешното представяне на данните и начина на достъп до тях)
- Съответства на начина на човешкото мислене
- Еднообразен начин за представяне на данните и връзките между тях във вид на таблица

Нека $D_1, D_2, \dots, D_n$ са множества, не непременно различни. Всяко от тези множества е именувано, разглежда се като множество от допустими стойности на някаква величина и се нарича още област или домейн. Домейните могат да бъдат безкрайни или да съдържат краен брой елементи. Всяко подмножество на декартовото произведение $D_1 \times D_2 \times \dots \times D_n$ на n области се нарича релация. В контекста на БД, релацията е двумерна таблица, в която се поместват данните. Т.е. релацията може да се разглежда като таблица от елементи, в която редовете са n-торки, а стълбовете съдържат елементи само от една и съща област.

Всеки именуван стълб от табличното представяне на дадена релация е прието да се нарича атрибут на релацията. Т.е. атрибутите са означенията (имената) на колоните на релацията (например: *title, year, length* и *filmType*). С всеки атрибут е асоцииран домейн (например: *Movies (title: string, year: integer, length: integer, filmType: string)*). Редовете в релацията (без заглавната част – атрибутите) се наричат кортежи (tuples) (например (*Star Wars, 1977, 124, color*)). С други думи, релацията е множество от кортежи, откъдето следва, че никой кортеж не може да се появява повече от веднъж. Релационният модел изисква всеки компонент на кортеж да бъде атомарен (неделим), т.е. се разрешават само прости типове (integer, string), а сложните типове като списъци и масиви са забранени.

Схема на релация; схема на релационна база от данни.

Релациите са такива структури от данни, които са променливи по съдържание и във всеки момент представят текущото състояние на класа от обекти. Същественото в случая е това, че те запазват структурата си през цялото време от съществуването си. Структурата на всяка релация се задава с нейната схема. Схемата на релацията се определя от името на релацията и множеството от атрибутите ѝ. Изписваме схемата на релацията, като указваме името ѝ, следвано от атрибутите в скоби – например *Movies (title, year, length, filmType)*.

Независимо от това, че атрибутите в релационната схема са множество, а не списък, ние трябва да укажем стандартна подредба на атрибутите един спрямо друг. В релационния модел проектът се състои от една или няколко релационни схеми. Съвкупността от всичките тези релационни схеми, с които се описват класовете от обекти и връзките между тях, се нарича **релационна схема на базата от данни** (накратко **схема на базата от данни**).

Схема на релация = име на релация + атрибути

База от данни = колекция от релации

Схема на база от данни = множество от схемите на релациите в базата от данни

Множеството от кортежи за дадена релация ще наричаме **екземпляр на релацията**. Екземплярите на релациите се променят с времето. Кардиналността на един екземпляр на дадена релация е равен на текущия брой кортежи в релацията. Всяко множество от конкретни екземпляри на релации се нарича **текущо състояние** на релационната БД.

Реализация на релационната база от данни.

Процесът на **проектиране** на една релационна база от данни се състои от следните стъпки:

- Определяне на данните (обектите и техните характеристики), които ще се съхраняват;
- Определяне на връзките между обектите;
- Определяне на ключовите и на свързващите колони;
- Определяне на ограниченията върху обектите и връзките между тях;
- Отстраняване на евентуални недостатъци (излишества и пр.);
- Реализиране на базата от данни.

Дефинирането на отделните релационни схеми е твърде свободно, но най-често при създаването им се спазват следните две правила:

- Всеки клас от обекти се представя с релация, чиято схема включва всички негови атрибути, които стават атрибути и на релацията. Всяка n-торка от релацията представя конкретен обект/екземпляр от класа. Ключовият атрибут или списъкът от ключовите атрибути на класа от обекти, т.е. тези, които еднозначно идентифицират отделните екземпляри в класа, се приемат за ключ на релацията.
- Връзките между два или повече класове от обекти се представят чрез релация, чиято релационна схема включва ключовите атрибути на всеки от тези класове от обекти.

Видове операции върху релационната база от данни. Заявки към релационната база от данни.

Както беше посочено, една реална релация не може да бъде статична, а се променя с времето. Възможни са два вида **операции върху релационната база от данни**:

- Selection, insertion, updating, deletion of tuples (често) – **DML** (Data Manipulation Language). Чрез тези операции се извличат, добавят, изтриват и променят **данните** в БД. DML операциите са **SELECT**, **INSERT**, **UPDATE** и **DELETE**. След като се изпълни един DML statement, трябва да се направи **COMMIT**, за да станат промените постоянни. **Примери**:

```
SELECT *  
FROM books
```

21.1. Релационен модел.

```
SELECT books.title, count(*) AS Authors
FROM books
JOIN book_authors
ON books.isbn = book_authors.isbn
WHERE books.price > 100.00
GROUP BY books.title;

INSERT INTO my_table (field1, field2, field3) VALUES ('test', 'N', NULL);

UPDATE my_table SET field1 = 'updated value' WHERE field2 = 'N';

DELETE FROM my_table WHERE field2 = 'N';
```

- Промени в схемата (не много често) – **DDL** (Data Definition Language). Тези операции служат за дефиниране и променяне на структурата на таблиците и другите обекти в базата от данни (редове, колони, индекси и т.н.). Когато се изпълни един DDL statement, промените се прилагат незабавно. DDL операцията са **CREATE** (за създаване), **DROP** (за премахване) и **ALTER** (за промяна). Примери:

```
CREATE TABLE employees (
    id            INTEGER    PRIMARY KEY,
    first_name    CHAR(50)   NULL,
    last_name     CHAR(75)   NOT NULL,
    date_of_birth DATE       NULL
);
DROP TABLE employees;
ALTER TABLE employees ADD place_of_birth CHAR(50);
ALTER TABLE employees DROP COLUMN place_of_birth;
```

Релационна алгебра.

Алгебрата представлява математическа система, при която чрез използването на различни оператори и операнди могат да се получат нови такива. **Релационната алгебра** е такава алгебра, чиито операнди са релации или променливи, представляващи релации. Операторите са проектирани да извършват такива действия, каквито бихме искали да извършваме с релации в една база от данни. РА е в основата на всеки език за заявки към база от данни – всички системи за управление на бази от данни (СУБД) използват РА като междинен език за изчисление на заявките. В зависимост от това дали релациите представляват множества (един кортеж не може да се среща повече от веднъж) или мултимножества (може да се съдържат повторения на кортежи) от кортежи, говорим за **ядро на РА** или за **разширена РА**.

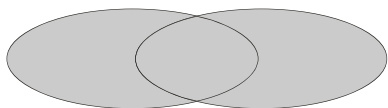
Нека $D_1, D_2, \dots, D_n$ са n области. Разглеждаме списъка от атрибути $A = A_1, A_2, \dots, A_n$, при който атрибутът A_i взема стойности от областта D_i , $i = 1, 2, \dots, n$. Нека B е друг списък от m атрибута $B_1, B_2, \dots, B_m$. Релациите $R(A_1, A_2, \dots, A_n)$ и $S(B_1, B_2, \dots, B_m)$ са съвместими, ако: 1) са с еднаква степен (т.е. $m = n$); 2) $\text{dom}(A_k) = \text{dom}(B_k)$, т.е. A_k и B_k са от един и същи тип за всяко $k = 1, 2, \dots, n$; 3) редът на атрибутите в двете релации съвпада.

Основни (обединение; разлика; декартово произведение; проекция; селекция) и допълнителни (сечение; частно; съединение; естествено съединение) операции в релационната алгебра.

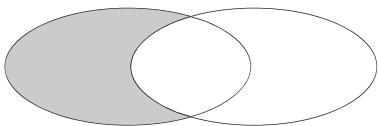
Освен на основни и допълнителни, операцията в релационната алгебра могат и да се класифицират по следния начин:

1. Операции върху множества – в случай, че R и S са с една и съща релационна схема:

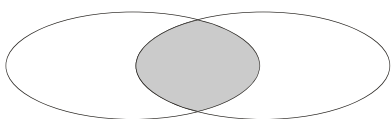
- **Обединение** (union) – бинарна (двучленна), комутативна, асоциативна операция – $R \cup S$



- **Разлика** (set difference) – бинарна операция; за разлика от другите операции, разликата не е комутативна, т.е. при размяна на местата на операндите резултатът не е един и същ $(R - S \neq S - R) - R - S$



- **Сечение** (intersection) – бинарна, комутативна, асоциативна операция – $R \cap S$: Сечението на две релации с обща реляционна схема е трета релация със същата реляционна схема, която съдържа общите за двете релации кортежи.



- **Частно** – бинарна, комутативна, асоциативна операция – $R : S$: Нека са дадени две релации $P(A_1, A_2, \dots, A_k, B_1, B_2, \dots, B_m)$ и $Q(B_1, B_2, \dots, B_m)$. Частното на P и Q е трета релация $R(A_1, A_2, \dots, A_k)$, която има свойството, че за всяко $t \in R$ и всяко $s \in Q$, конкатенацията $ts \in P$.

Пример:

R			
NAME	ADDRESS	GENDER	BIRTHDATE
Carrie Fisher	123 Maple St., Hollywood	F	9/9/99
Mark Hamil	456 Oak Rd., Brentwood	M	8/8/88

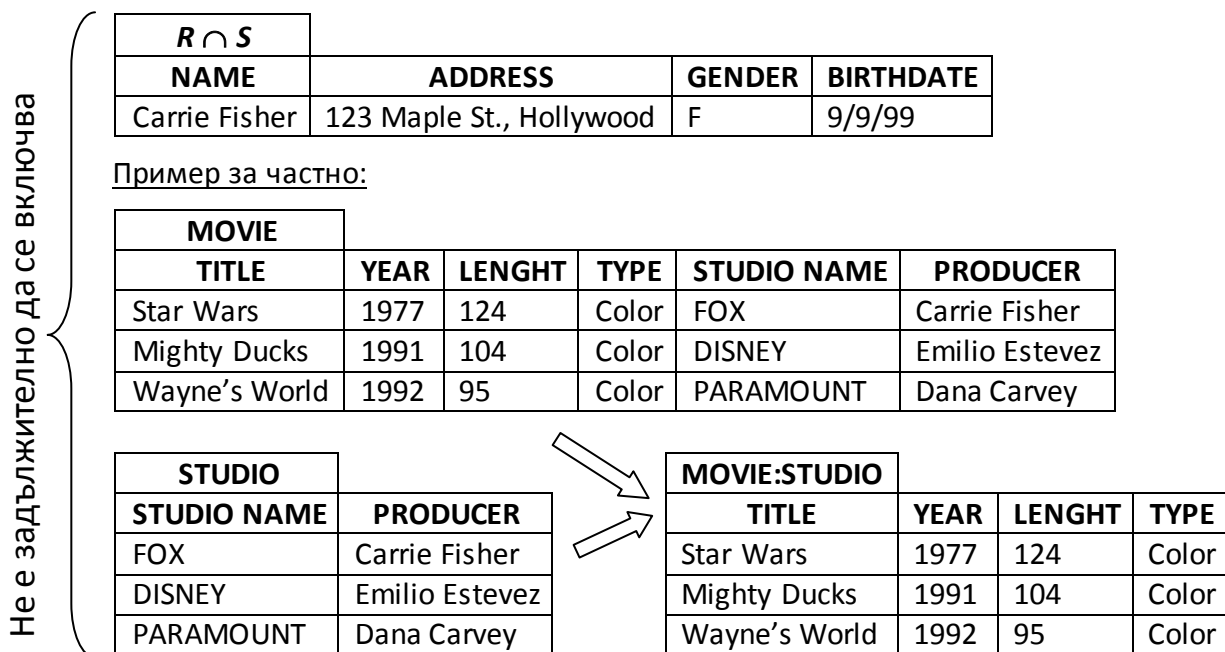
S			
NAME	ADDRESS	GENDER	BIRTHDATE
Carrie Fisher	123 Maple St., Hollywood	F	9/9/99
Harrison Ford	789 Palm Dr., Beverly Hills	M	7/7/77

Резултатите от прилагането на трите основни операции:

$R \cup S$			
NAME	ADDRESS	GENDER	BIRTHDATE
Carrie Fisher	123 Maple St., Hollywood	F	9/9/99
Harrison Ford	789 Palm Dr., Beverly Hills	M	7/7/77
Mark Hamil	456 Oak Rd., Brentwood	M	8/8/88

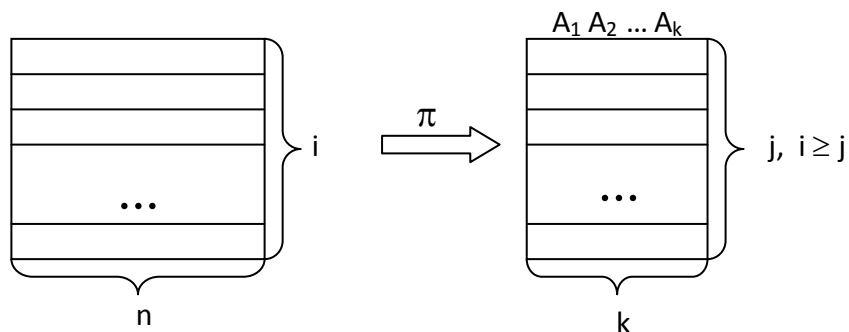
$R - S$			
NAME	ADDRESS	GENDER	BIRTHDATE
Mark Hamil	456 Oak Rd., Brentwood	M	8/8/88

Не е задължително да се включва



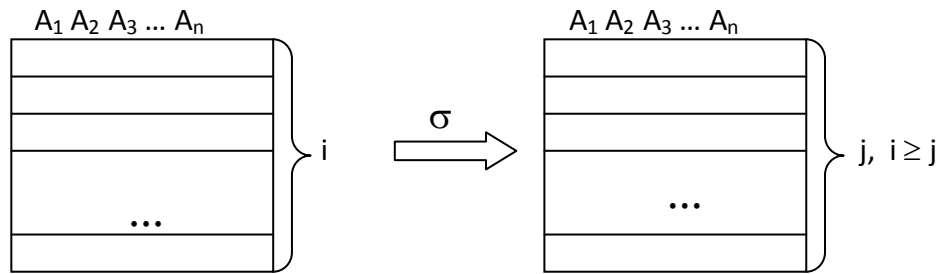
2. Операции, които отстраняват части от релациите

• **Проекция** (хоризонтална рестрикция, намаляване в ширина): Нека R е n -членна релация с релационна схема $P(A_1, A_2, \dots, A_n)$. Проекцията на релацията R по отношение на атрибутите $A_{i_1}, A_{i_2}, \dots, A_{i_k}$ е k -членна релация $R(A_{i_1}, A_{i_2}, \dots, A_{i_k})$, която се получава от R чрез отстраняване на всички атрибути, нефигуриращи в схемата на R и в която повтарящите се елементи (кортежи) са представени само по един път. Записва се по следния начин: $\pi_{\langle \text{attr_list} \rangle}(R)$, където $\langle \text{attr_list} \rangle$ е списък от атрибутите, които ще бъдат резултат от операцията. Въпреки че основният смисъл тук е намаляване в ширина, възможно е и намаляване в дълбочина, когато избраните атрибути имат много повтарящи се стойности, а, тъй като тук се работи само с множества, резултатът е унифициран. Това е един от минусите на работата с множества – може да се получи загуба на информация.



• **Селекция** (рестрикция, вертикална рестрикция, намаляване в дълбочина) – унарна, комутативна операция: Нека R е n -членна релация с релационна схема $P(A_1, A_2, \dots, A_n)$. Селекцията на релацията R по отношение на дадено условие F е друга релация R със същата релационна схема, всеки елемент на която удовлетворява условието F . Записва се по следния начин: $\sigma_{\langle \text{predicate} \rangle}(R)$, където $\langle \text{predicate} \rangle$ е условието, на което трябва да отговарят атрибутите. Предикатите имат следния вид: $\langle \text{attribute} \rangle \langle \text{op} \rangle \langle \text{attribute} \rangle$ или $\langle \text{attribute} \rangle \langle \text{op} \rangle \langle \text{constant} \rangle$, където $\langle \text{attribute} \rangle$ са атрибути на R , $\langle \text{constant} \rangle$ са константи, а $\langle \text{op} \rangle$ са оператори като $=, \neq, <, >, \leq, \dots$, AND, OR.

21.1. Релационен модел.



Пример:

MOVIE					
TITLE	YEAR	LENGHT	FILM TYPE	STUDIO NAME	PRODUCER
Star Wars	1977	124	Color	FOX	Carrie Fisher
Mighty Ducks	1991	104	Color	DISNEY	Emilio Estevez
Wayne's World	1992	95	Color	PARAMOUNT	Dana Carvey

Примери за проекция:

$\pi_{\text{title, year, length}}(\text{Movie})$:

TITLE	YEAR	LENGHT
Star Wars	1977	124
Mighty Ducks	1991	104
Wayne's World	1992	95

$\pi_{\text{film type}}(\text{Movie})$:

FILM TYPE
Color

Пример за селекция:

$\sigma_{\text{length} \geq 100}(\text{Movie})$

MOVIE					
TITLE	YEAR	LENGHT	FILM TYPE	STUDIO NAME	PRODUCER
Star Wars	1977	124	Color	FOX	Carrie Fisher
Mighty Ducks	1991	104	Color	DISNEY	Emilio Estevez

Не е задължително да се включва

3. Операции, които комбинират кортежи от две или повече релации

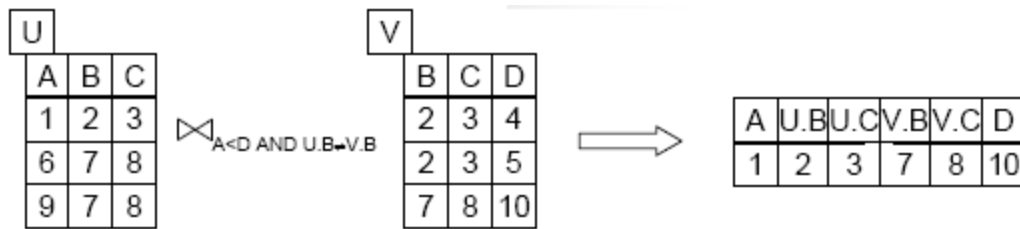
• **Декартово произведение** (Cartesian product) – бинарна, комутативна, асоциативна операция – $R \times S$: Нека са дадени две релации $P(A_1, A_2, \dots, A_m)$ и $Q(B_1, B_2, \dots, B_n)$. Декартово произведение на P и Q е трета $(m+n)$ -членна релация $R(A_1, A_2, \dots, A_m, B_1, B_2, \dots, B_n)$, съдържаща всички възможни конкатенации на елементи от P и Q . Ако в P и Q има атрибут с едно и също име (например A), в резултата се записва $P.A$ и $Q.A$ за уточняване.

R		S									
A	B	B	C	D		A	R.B	S.B	C	D	
1	2	2	5	6	x	1	2	2	5	6	
1	2	4	7	8		1	2	4	7	8	
1	2	9	10	11		1	2	9	10	11	
3	4	2	5	6		3	4	2	5	6	
3	4	4	7	8		3	4	4	7	8	
3	4	9	10	11		3	4	9	10	11	

• **Съединение** (θ -съединение, theta join) – $R \bowtie_C S$: Съединение на релациите $P(A_1, A_2, \dots, A_m)$ и $Q(B_1, B_2, \dots, B_n)$ по отношение на условието C е $(m+n)$ -членна релация с релационна схема $R(A_1, A_2, \dots, A_m, B_1, B_2, \dots, B_n)$, която е такова подмножество на декартовото произведение $P \times Q$, в което всеки кортеж отговаря на условието C . Този вид съединение

21.1. Релационен модел.

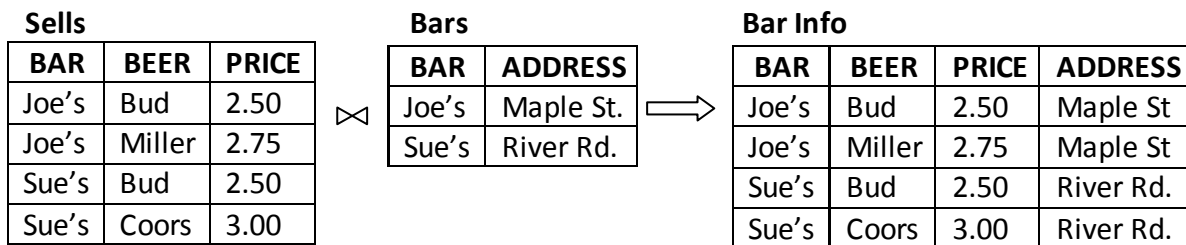
се извършва на два етапа: първо се прави декартово произведение на двете релации; след това се прави селекция върху получената релация по отношение на условието С.



• **Еквисъединение** (equijoin) – частен случай на θ -съединението, когато θ е отношението равенство (=), т.е. условието на съединението включва само съвпадение по атрибутите.

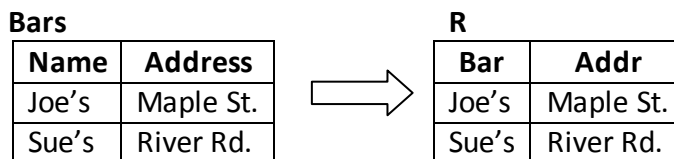
• **Естествено съединение** (natural join) – $R \bowtie S$: Разширение на еквисъединението, при което се извършва свързване по всички атрибути с еднакви имена. Автоматично се отстранява повтарящата се колона.

Пример:



4. **Преименуване** (rename) – тези операции не оказват влияние върху конструкцията на релацията, а само променят имената на таблиците или на техните атрибути. Операторът ρ дава нова схема на релацията. $R_1 := \rho_{R_1(A_1, \dots, A_n)}(R_2)$ превръща R_1 в релация с атрибути $A_1, \dots, A_n$ и същите кортежи като на R_2 . Записва се още $R_1(A_1, \dots, A_n) := R_2$.

Пример: $R(\text{bar}, \text{addr}) := \text{Bars}$



Представяне на допълнителните операции чрез основните.

- **Сечение:** $R \cap S = R - (R - S)$
- **Съединение:** $R \bowtie_c S = \sigma_c (R \times S)$
- **Естествено съединение:** $R \bowtie S = \pi_L (\sigma_c (R \times S))$, където L е списък от атрибутите на R , последвани от атрибутите на S , които не присъстват в R ; а C е условие от вида $R.A_1 = S.A_1 \text{ AND } R.A_2 = S.A_2 \text{ AND } \dots R.A_n = S.A_n$, където $A_1, A_2, \dots, A_n$ са всички общи за R и S атрибути.

Приоритет на релационните оператори.

- Унарни оператори – селекция, проекция, преименуване
- Декартово произведение и съединение
- Сечение
- Обединение и разлика

21.2. БАЗИ ОТ ДАННИ: Нормални форми.

Нормални форми. Проектиране схемите на релационните бази от данни. Аномалии, ограничения, ключове. Функционални зависимости, аксиоми на Армстронг. Първа, втора, трета нормална форма, нормална форма на Бойс-Код. Многозначни зависимости; аксиоми на функционалните и многозначните зависимости; съединение без загуба; четвърта нормална форма.

Аномалии.

Да разгледаме един пример, с който се илюстрират различни недостатъци на една конкретна релационна схема, известни като **аномалии** на излишество, обновяване, добавяне и отстраняване.

Дадена е релационната схема **БИБЛИОТЕКА** (*Биб#*, *Адрес*, *Книга*, *Брой*), с която всяка библиотека е представена чрез своя номер (*Биб#*) и адрес (*Адрес*), а всяка книга е представяна с името си (*Книга*) и броя екземпляри (*Брой*), които са налични в съответната библиотека. Естествено е да се приеме, че всяка библиотека е еднозначно определена от своя номер, както и, че една и съща книга може да се съхранява в няколко библиотеки. Като следствие от това могат да се направят следните изводи:

1. Аномалия от излишество – За всяка книга от една и съща библиотека ще има повторение на адреса на библиотеката.
2. Аномалия при обновяване – От (1) следва, че при промяна на адреса на библиотеката ще е необходимо да се обновят толкова реда на релацията, колкото са книгите, които се съхраняват в нея. При тази операция не е изключена възможността някои от редовете да останат необновени, при което ще настъпи противоречивост в данните на БД.
3. Аномалия при добавяне – Адресът на една новосъздаваща се библиотека не може да бъде въведен в БД, докато не се посочи поне една книга, която ще се съхранява в тази библиотека. Причината за това е, че ключовите атрибути в тази релационна схема са атрибутите *Биб#* и *Книга* и следователно те трябва да имат точно определени стойности за всеки елемент от релацията.
4. Аномалия при изтриване – Ако поради ремонт на дадена библиотека книгите, които са се съхранявали в нея, бъдат преместени в друго книгохранилище или библиотека, ще се наложи всички редове от релацията, отнасящи се до тази библиотека, да бъдат отстранени от релацията. Като следствие от това ще бъде "изгубен" и адресът на библиотеката, независимо че тя съществува и адресът ѝ остава същият.

Ключове.

Понеже всяка релация е множество от неповтарящи се n -торки, то всяка релация има **ключ**, който определя еднозначно кортежите в релацията. Естествено е една релация да има и повече от един ключ. Ако едно множество от атрибути A е ключ за една релация R , то всяко съдържащо A множество X също ще идентифицира еднозначно елементите на R . Затова към всеки ключ се предявява и изискването за минималност: Ако K е ключ на релацията R , тогава: 1) За всеки два елемента $r_1 \in R$ и $r_2 \in R$ е в сила $r_1[K] \neq r_2[K]$; 2) Не съществува подмножество от атрибути на K , за което (1) да остава в сила.

Един от всички възможни ключове на всяка релация се избира за ключ на релацията и той се нарича **първичен ключ**.

Външният ключ K^* за една релация R е такъв атрибут или списък от атрибути, който не е ключ за R , но съществува релация Q , за която K^* е първичен ключ.

Ограничения.

Ограниченията дефинират правила за стойностите, допустими за съответните колони, и представляват стандартен механизъм за налагане на цялост. С други думи, ограничението е свойство на таблица или колона от таблица, което предотвратява въвеждането на невалидни стойности на данните. Ето основните ограничения в релационния модел:

Може по-накратко да се споменат

- Ограниченията UNIQUE и PRIMARY KEY (първичен ключ) не допускат да бъде въведена стойност, която дублира съществуваща стойност;
- Стойностите на атрибутите, съставлящи първичния ключ, не могат да бъдат нулеви;
- Ограничението CHECK не допуска да бъде въведена стойност, която не отговаря на определени условия (например $\text{Age} > 0$ или $\text{length(EGN)} = 10$);
- Ограничението FOREIGN KEY (външен ключ) налага връзка между данните в две таблици – за всяка ненулева стойност на атрибут, който е външен ключ, трябва да съществува ключов атрибут от друга релация, който съдържа тази стойност;
- Ограничението NOT NULL задава, че не се допуска стойността на атрибут (не непременно ключов) да е нулева (null). Стойност null не означава нула, интервал или символен низ с нулева дължина, като "". Null означава, че не са били въведени данни. Наличието на null обикновено означава, че стойността е или неизвестна, или недефинирана.

Освен тези ограничения, друг широк клас от ограничения за цялостност се въвеждат с помощта на функционални зависимости.

Функционални зависимости.

Нека $A_1, A_2, \dots, A_n$ е един какъв да е списък от n атрибута, $n \geq 1$, който накратко ще означаваме с $A_1A_2\dots A_n$ (без запетайки) или само с A . Предполага се, че списъкът A е неподреден и няма повтарящи се елементи, т.е. A е множество от имена на атрибути. Знае се, че ако A има n елемента, то броят на подмножествата на A е 2^n .

Нека r е релация, а $R(A)$ е нейната релационна схема. $r(R)$ е математическа релация от степен n върху домейните $\text{dom}(A_1), \text{dom}(A_2), \dots, \text{dom}(A_n)$ – подмножество на декартовото произведение на домейните, които дефинират R . Това може да се изрази чрез $r(R) \subseteq (\text{dom}(A_1) \times \text{dom}(A_2) \times \dots \times \text{dom}(A_n))$. $r(R)$ е множество от n -торки (tuples), $r = \{t_1, t_2, \dots, t_m\}$. Всяка n -торка е подреден списък от n стойности $t = \langle v_1, v_2, \dots, v_n \rangle$, където $v_i, 1 \leq i \leq n$, е елемент от $\text{dom}(A_i)$ или специалната стойност NULL.

Нека $X, Y \in A$. Казва се, че атрибутът X **функционално определя** атрибута Y , ако за всяка релация r с релационна схема $R(A)$ при наличието на кои да е два кортежа $t_1, t_2 \in r$, за които $t_1.X = t_2.X$, следва, че $t_1.Y = t_2.Y$, т.е. при равенство на кортежи по атрибута X следва равенството им по атрибута Y . X и Y могат да бъдат както прости атрибути, така и съставни, т.е. да се състоят от няколко прости атрибута (ако $A_1A_2\dots A_n \rightarrow B_1$ и $A_1A_2\dots A_n \rightarrow B_2$ и $A_1A_2\dots A_n \rightarrow B_m$, то $A_1A_2\dots A_n \rightarrow B_1B_2\dots B_m$). Когато X функционално определя Y , казва се също, че Y **зависи функционално** от X или че е налице **функционална зависимост** (ФЗ) от X в Y ,

като се използва означението $R.X \rightarrow R.Y$. Множеството от ФЗ се означава с F , а с малка буква f се означава единична ФЗ. Зависимостта $A_1A_2...A_n \rightarrow B$ се нарича функционална, защото има функция, която на списък от стойности (по една за всяко $A_1A_2...A_n$) съпоставя уникална стойност за B . Тук функцията обаче не се изчислява по стандартния начин, а „изчислението“ става чрез търсене в релацията.

ФЗ са твърдения за схемата на релацията, не за конкретен екземпляр, т.е. те са свойства на семантиката на атрибутите и всички данни ги удовлетворяват. Затова те не могат да се определят чрез просто преглеждане на данните.

Графично представяне на ФЗ $A \rightarrow B$:



Например, в релацията *Movies* се наблюдават следните функционални зависимости: *title year* $\rightarrow$ *length*; *title year* $\rightarrow$ *filmType*; *title year* $\rightarrow$ *studioName*; но НЕ! *title year* $\rightarrow$ *starName*.

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

1. Множеството K функционално определя всички атрибути на R ;
2. За нито едно подмножество на K не е вярно (1).

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

Например, за релацията *Movies* (*title*, *year*, *length*, *filmType*, *studioName*, *starName*) ключът е $\{title, year, starName\}$ и няма други ключове, но има много суперключове – всички супермножества на $\{title, year, starName\}$, например $\{title, year, starName, length, type\}$.

Аксиоми на Армстронг.

Зависимостите, които е възможно да се определят в рамките на една реляционна схема, макар и краен брой, понякога могат да бъдат твърде много. Затова естествено е да се постави въпросът, как по дадено множество от ФЗ F могат да се получат всичките ФЗ.

Една ФЗ $X \rightarrow Y$ се нарича логическо следствие на множеството от ФЗ F , ако за всяка релация r , удовлетворяваща зависимостите F , следва, че $X \rightarrow Y$ се удовлетворява от r .

Армстронг показва, че като се използват т.нар. правила или аксиоми за извод при дадено множество от ФЗ, се получават нови ФЗ, при това могат да се получат всичките ФЗ. При формулировката на аксиомите, дадени по-долу, се предполага, че: $R(A)$ е реляционна схема; F е множество от ФЗ в A ; X, Y, Z и W са списъци от атрибути на A ; r е произволна релация със схема $R(A)$.

A1. Рефлексивност (reflexivity): Ако $Y \subseteq X$, то $X \rightarrow Y$.

Пример: *title year* $\rightarrow$ *title*

A2. Разширение, попълнение (augmentation): Ако $X \rightarrow Y$, то $XW \rightarrow YW$.

Пример: От $title\ year \rightarrow length$ получаваме $title\ year\ filmType \rightarrow length\ filmType$

A3. Транзитивност (transitivity): Ако $X \rightarrow Y$ и $Y \rightarrow Z$, то $X \rightarrow Z$.

Може да не се доказват { Доказателства на аксиомите на Армстронг:

A1. Рефлексивност: Всеки 2 кортежа t и u съвпадат по всички атрибути на X , $\Rightarrow$ те съвпадат и по всяко подмножество на X , включително Y .

A2. Разширение, попълнение: Да допуснем, че има кортежи t и u , които съвпадат по всички атрибути на XW , но не съвпадат по YW . t и u задължително съвпадат по W , $\Rightarrow$ се различават по някой от атрибутите на Y , което противоречи на $X \rightarrow Y$.

A3. Транзитивност: Да допуснем, че има 2 кортежа (x, y_1, z_1) и (x, y_2, z_2) , които съвпадат по всички атрибути на X . $X \rightarrow Y$, следователно, щом съвпадат по всички атрибути на X , задължително съвпадат по всички атрибути на Y , т.е. $y_1 = y_2$. Аналогично, от $Y \rightarrow Z$ следва, че $z_1 = z_2$. $\Rightarrow$ двата кортежа съвпадат.

Следствия от аксиомите на Армстронг:

Сл.1. Обединение: Ако $X \rightarrow Y$ и $X \rightarrow Z$, то $X \rightarrow YZ$.

Сл.2. Псевдотранзитивност: Ако $X \rightarrow Y$ и $WY \rightarrow Z$, то $XW \rightarrow Z$.

Сл.3. Декомпозиция: Ако $X \rightarrow Y$ и $Z \subseteq Y$, то $X \rightarrow Z$.

Може без док-ва { Доказателства на следствията от аксиомите на Армстронг:

Сл.1. Обединение: $X \rightarrow Y$, следователно $X \rightarrow XY$ (A2). $X \rightarrow Z$, $\Rightarrow XY \rightarrow ZY$ (A2). От получените $X \rightarrow XY$ и $XY \rightarrow ZY$ следва, че $X \rightarrow ZY$ (A3).

Сл.2. Псевдотранзитивност: $X \rightarrow Y$, следователно $WX \rightarrow WY$ (A2). Но имаме, че $WY \rightarrow Z$, следователно $WX \rightarrow Z$ (A3).

Сл.3. Декомпозиция: $Z \subseteq Y$, следователно $Y \rightarrow Z$ (от A1). Но имаме, че $X \rightarrow Y$, следователно $X \rightarrow Z$ (A3).

Правило за декомпозиция: Ако $AA \rightarrow B_1B_2...B_n$, то $AA \rightarrow B_1$, $AA \rightarrow B_2$, ..., $AA \rightarrow B_n$.

Правило за обединение: Ако $AA \rightarrow B_1$, $AA \rightarrow B_2$, ..., $AA \rightarrow B_n$, то $AA \rightarrow B_1B_2...B_n$.

Функционалната зависимост $A_1A_2...A_n \rightarrow B$ се нарича тривиална, ако атрибутът B съвпада с някой от атрибутите $A_1A_2...A_n$. В противен случай се нарича нетривиална.

Когато B е съставен атрибут, т.е. $B = B_1B_2...B_m$, то ФЗ $A_1A_2...A_n \rightarrow B_1B_2...B_m$ е:

- тривиална, ако $B_1B_2...B_m \subseteq A_1A_2...A_n$;
- нетривиална, ако поне един атрибут от $B_1B_2...B_m$ не е от $A_1A_2...A_n$.
- напълно нетривиална, ако никой от атрибутите от $B_1B_2...B_m$ не е от $A_1A_2...A_n$.

Правило на тривиалната зависимост: Имаме право от дясната част на една ФЗ да премахнем тези атрибути, които принадлежат на лявата част – така от $A_1A_2...A_n \rightarrow B_1B_2...B_m$ получаваме $A_1A_2...A_n \rightarrow C_1C_2...C_k$, където $\{C_1, C_2, ..., C_k\} \subseteq \{B_1, B_2, ..., B_m\}$ и нито един от атрибутите C не е от $A_1A_2...A_n$.

Първа, втора, трета нормална форма, нормална форма на Бойс-Код.

Виждаме, че основен проблем при изграждането на модели на дадена предметна област е определянето на същностите и на свойствата, които ги характеризират. Сложността на проблема е следствие от нееднозначността на неговото решение. Естествено е в такъв случай да се поставят въпроси като: кога една релационна схема е добра; кога две схеми са еквивалентни или коя от двете схеми е по-добрата. Всеки от тези въпроси всъщност се докосва до необходимостта от формален инструмент за анализ на релационни схеми. Такъв инструмент е т.нар. **нормализация** – процес, насочен към преобразуването на релационни схеми, при който на новополучените релационни схеми се налагат известни ограничения, елиминиращи някои нежелани свойства.

Една релационна схема е в **първа нормална форма** (1НФ), ако областите на съставлящите я атрибути са атомарни (прости), т.е. атрибутите ѝ са атомарни. Искане се и, естествено, да няма повтарящи се кортежи, което е основно изискване за една релация.

Един атрибут X от релационната схема $R(A)$ с множество от ФЗ F се нарича **първичен атрибут**, ако влиза в състава на ключа (първичния ключ). В противен случай се нарича **непървичен**. Една релационна схема е във **втора нормална форма** (2НФ) относно множеството от ФЗ F , ако тя е в 1НФ и всеки непървичен атрибут е в **пълна ФЗ** от ключа, т.е. зависи от целия ключ, а не от някакво негово подмножество.

Нека $R(A)$ е релационна схема със съответно множество от ФЗ F , а X и Z са подмножества на A . Казва се, че атрибутът Z е **транзитивно зависим** от X , ако съществува такова множество от атрибути Y , $Y \subset A$, $Z \notin XY$, за което са в сила свойствата: $X \rightarrow Y$, $Y \rightarrow Z$. Една релационна схема $R(A)$ е в **трета нормална форма** (3НФ) относно множеството от ФЗ F , ако тя е в 1НФ и нито един първичен атрибут не е транзитивно зависим от ключа (а всеки непървичен атрибут е не-транзитивно, т.е. директно зависим от ключа). Друга дефиниция на **3НФ**: R е в 3НФ тогава и само тогава, когато за всяка нетривиална ФЗ или лявата страна е суперключ, или дясната е първичен атрибут.

В определението на 2НФ се иска наличие на 1НФ, т.е. всяка релационна схема, която е във 2НФ, е и в 1НФ. В определението на 3НФ не се изисква релационната схема да е във 2НФ, но може да се покаже, че всяка релационна схема, която е в 3НФ, е и във 2НФ.

Релацията R е в **нормална форма на Бойс-Код** (Boyce-Codd Normal Form, BCNF) тогава и само тогава, когато за всяка нетривиална зависимост $A_1A_2...A_n \rightarrow B_1B_2...B_m$ от R , съответното множество от атрибути $\{A_1, A_2, ..., A_n\}$ е суперключ за R .

Многозначни зависимости

Терминът “**многозначна зависимост**” се използва, когато два атрибута или множество атрибути са независими помежду си. Това състояние обобщава идеята за ФЗ в този смисъл, че всяка функционална предполага съответна многофункционална зависимост.

Съществуват схеми в BCNF, които съдържат излишни данни. Най-често това се получава при опит да се поставят две или повече връзки “много към много” в една релация. Това излишество е резултат от независимостта на атрибутите. Например, в Tutor/Student Cross-Reference без проблем могат да се въведат два различни номера на социална осигуровка за един и същ ръководител, а това не е желателно.

Пример:

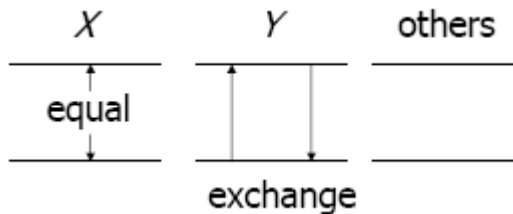
Ще предположим, че известните личности имат по няколко адреса. Разделяме тези адреси на части (град, улица). Редом до информацията за “звездите” ще включим и познатата ни информация за заглавията и годините на филмите, в които са участвали те.

Name	Street	City	Title	Year
Carrie Fisher	123 Maple Str	Hollywood	Star Wars	1977
Carrie Fisher	5 Locust Ln.	Malibu	Star Wars	1977
Carrie Fisher	123 Maple Str	Hollywood	Empire Strikes Back	1980
Carrie Fisher	5 Locust Ln.	Malibu	Empire Strikes Back	1980
Carrie Fisher	123 Maple Str	Hollywood	Return or the Jedi	1983
Carrie Fisher	5 Locust Ln.	Malibu	Return or the Jedi	1983

Нека обърнем внимание на двата адреса и трите най известни филма на Carrie Fisher. Няма причина да асоциираме определен адрес с определен филм, а не с друг филм. Поради това единственият начин да покажем, че адресите и филмите са независими свойства, е да добвим всеки адрес към всеки филм. Но, когато повторим адресите и данните за филмите във всички комбинации, се получава явно излишество на данни. Например, всеки от адресите на Carrie Fisher се повтаря по три пъти (веднъж за всеки от филмите) и всеки филм се повтаря по два пъти (веднъж за всеки от адресите).

Въпреки това, няма нарушение на BCNF в тази релация. Няма и никакви нетривиални ФЗ. Например атрибутът City не е функционално определен от другите 4 атрибута. Може да има звезда с два дома, които имат един и същ адрес на улица в различни градове. Тогава ще има два кортежа, чиито атрибути си съответстват напълно, освен по атрибута City. Така че $name\ street\ title\ year \rightarrow city$ НЕ е ФЗ за нашата релация. Никой от петте атрибута не е функционално определен от другите четири. Тъй като няма нетривиални ФЗ, следва, че всички пет атрибута формират един суперключ и няма нарушение на BCNF.

Многозначна зависимост (multivalued dependency, MVD, M3) е твърдение за релация R, за която, при фиксиране на стойностите за определени атрибути, стойностите в точно определени други атрибути са независими от стойностите на всички други атрибути в релацията. **Многозначната зависимост** $X \twoheadrightarrow Y$ утвърждава, че, ако 2 кортежа в една релация съвпадат по всички атрибути на X, техните компоненти от множеството атрибути Y могат да бъдат разменени и резултатът ще даде 2 нови кортежа, които също принадлежат на релацията.



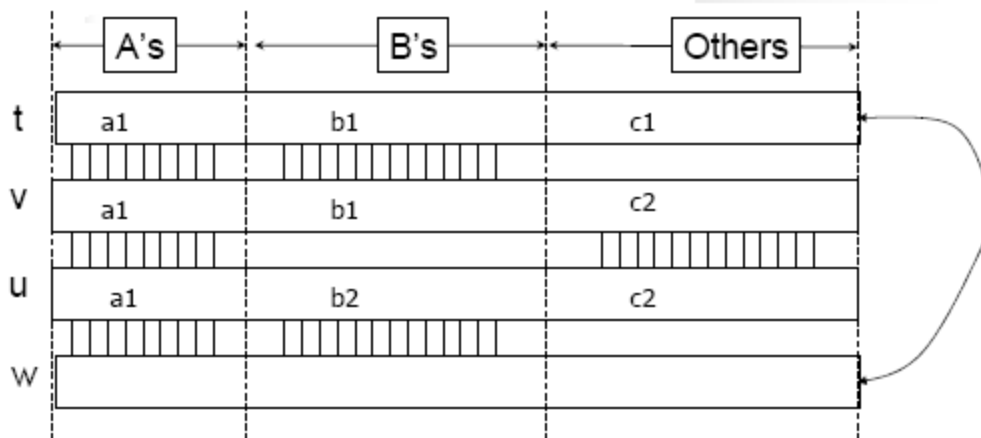
Дефиниция: $A_1A_2...A_n \twoheadrightarrow B_1B_2...B_m$ е **многозначна зависимост** в R, ако за всяка двойка кортежи t, u от R, за които $t[A_1A_2...A_n] = u[A_1A_2...A_n]$, съществува кортеж v от R, за който:

- $v[A_1A_2...A_n] = t[A_1A_2...A_n] = u[A_1A_2...A_n]$
- $v[B_1B_2...B_m] = t[B_1B_2...B_m]$
- $v[C_1C_2...C_k] = u[C_1C_2...C_k]$,

където $C_1C_2...C_k$ са всички атрибути от R, с изключение на $(A_1A_2...A_n \cup B_1B_2...B_m)$.

Можем да използваме това правило с разменени t и u, за да подскажем съществуването на четвърти кортеж w, който се съгласува с u и t. В последствие за всички фиксирани

стойности на A съответните стойности от B и останалите атрибути се появяват във всички възможни комбинации в различни кортежи.



Пример: name $\rightarrow\rightarrow$ street city

Name	Street	City	Title	Year
C. Fisher*	123 Maple Str	Hollywood	Star Wars*	1977
C. Fisher*	5 Locust Ln.*	Malibu*	Star Wars*	1977
C. Fisher	123 Maple Str	Hollywood	Empire Strikes Back	1980
C. Fisher*	5 Locust Ln.*	Malibu*	Empire Strikes Back	1980
C. Fisher	123 Maple Str	Hollywood	Return or the Jedi	1983
C. Fisher	5 Locust Ln.	Malibu	Return or the Jedi	1983

Аксиоми на многозначните зависимости.

МЗ $A_1A_2...A_n \rightarrow\rightarrow B_1B_2...B_m$ се нарича **тривиална**, когато $B_1B_2...B_m \subseteq A_1A_2...A_n$ или $(A_1A_2...A_n \cup B_1B_2...B_m)$ съдържа всички атрибути на R.

МЗ $A_1A_2...A_n \rightarrow\rightarrow B_1B_2...B_m$ е **нетривиална**, когато нито един от атрибутите $B_1B_2...B_m$ не съвпада с $A_1A_2...A_n$ и не всички атрибути на R принадлежат на $(A_1A_2...A_n \cup B_1B_2...B_m)$.

Има няколко правила за МЗ, които са подобни на тези за ФЗ:

Пр.1. Транзитивно правило: Ако $A_1A_2...A_n \rightarrow\rightarrow B_1B_2...B_m$ и $B_1B_2...B_m \rightarrow\rightarrow C_1C_2...C_k$, то $A_1A_2...A_n \rightarrow\rightarrow C_1C_2...C_k$.

Пр.2. Правило на обединението: Ако $X_1X_2...X_n \rightarrow\rightarrow Y_1Y_2...Y_m$ и $X_1X_2...X_n \rightarrow\rightarrow Z_1Z_2...Z_k$, то $X_1X_2...X_n \rightarrow\rightarrow (Y_1Y_2...Y_m \cup Z_1Z_2...Z_k)$.

Пр.3. Правило на допълнението: Ако $A_1A_2...A_n \rightarrow\rightarrow B_1B_2...B_m$, то $A_1A_2...A_n \rightarrow\rightarrow C_1C_2...C_k$, където $C_1C_2...C_k$ е м-то от всички атрибути на R с изключение на $(A_1A_2...A_n \cup B_1B_2...B_m)$.

Пример: От name $\rightarrow\rightarrow$ street city, съгласно правилото на допълнението name $\rightarrow\rightarrow$ title year също трябва да е в сила за тази релация, защото title и year са атрибути, които не са споменати в първата МЗ. Втората МЗ означава, че всяка всяка известна личност е участвала в няколко филма, които са независими от адресите на звездата.

Забележка: Подобно на ФЗ, не може да се разделя лявата част на МЗ. За разлика от ФЗ обаче, не може да се разделя и дясната част – понякога се налага да се оставят няколко атрибута в дясната част. Например, ако върху МЗ name $\rightarrow\rightarrow$ street city се приложи правилото за разделяне, тогава трябва да очакваме, че е вярно и следното: name $\rightarrow\rightarrow$

street. Тази МЗ показва, че всеки адрес на звезда е независим от другите атрибути, включително и от града. Обаче това твърдение не е вярно. Например – първите 2 кортежа. Предполагаемата МЗ ще ни позволи да загатнем, че кортежите с улиците са се разменили:

Name	Street	City	Title	Year
C. Fisher	5 Locust Ln.	Hollywood	Star Wars	1977
C. Fisher	123 Maple Str	Malibu	Star Wars	1977

Но тези кортежи не са правилни, защото домът на 5 Locust Ln. се намира в Малибу, а не в Холивуд.

Пр.4. Правило FD-IS-AN-MVD (всяка функционална зависимост е многозначна зависимост):

Ако $A_1A_2...A_n \rightarrow B_1B_2...B_m$ то: $A_1A_2...A_n \twoheadrightarrow B_1B_2...B_m$.

Доказателство: Да предположим, че R е релация, за която е вярна ФЗ $A_1A_2...A_n \rightarrow B_1B_2...B_m$, и да предположим, че t и u са кортежите, които съвпадат в A. За да покажем, че е вярна МЗ $A_1A_2...A_n \twoheadrightarrow B_1B_2...B_m$, трябва да докажем, че R също съдържа кортеж v, който се съгласува с t и u по A, с t по B и с u по всички останали атрибути. Но v може да е u. Със сигурност u съвпада с t и u в A. ФЗ $A_1A_2...A_n \rightarrow B_1B_2...B_m$ предполага, че u съвпада с t в B. И разбира се u се съгласува със себе си по другите атрибути. Така че, когато е в сила функционалната зависимост, то е в сила и многозначната зависимост.

Декомпозиция на релации. Съединение без загуба.

Декомпозиция на релацията $R(A_1, ..., A_n)$ е заместването ѝ с множество релации $R_1, ..., R_n$, получени чрез проекции така, че R и $R_1 \cup R_2 \cup ... \cup R_n$ имат една и съща схема.

Например, дадена е релация R със схема $\{A_1, A_2, ..., A_n\}$. За да няма загуба на информация и да се намери най-оптималното решение за представяне, трябва да се декомпозира в две релации S и T със схеми $\{B_1, B_2, ..., B_m\}$ и $\{C_1, C_2, ..., C_k\}$, така, че: $\{A_1, A_2, ..., A_n\} = \{B_1, B_2, ..., B_m\} \cup \{C_1, C_2, ..., C_k\}$. Кортежите в релацията S са проекции на всички кортежи в R върху $\{B_1, B_2, ..., B_m\}$ (за всеки кортеж t от текущия екземпляр на R се избират компонентите, които съответстват на атрибутите $B_1, B_2, ..., B_m$; тези компоненти образуват нов кортеж, който принадлежи на текущия екземпляр на S; прави се само по 1 копие на кортеж). Аналогично, кортежите в релацията T са проекции на всички кортежи в R върху атрибутите $\{C_1, C_2, ..., C_k\}$.

Нека $R(A_1, A_2, ..., A_n)$ е релация и R се декомпозира на две релации S ($B_1, B_2, ..., B_m$) и T ($C_1, C_2, ..., C_k$). Казваме, че декомпозицията е със съединение без загуба, ако $R = S \bowtie T$.

Декомпозицията на релацията $R(A_1, A_2, ..., A_n)$ на 2 релации S ($B_1, B_2, ..., B_m$) и T ($C_1, C_2, ..., C_k$) е със съединение без загуба тогава и само тогава, когато за R е изпълнена поне една от следните функционални зависимости: $S \cap T \rightarrow S$ или $S \cap T \rightarrow T$.

Нека $R(A_1, A_2, ..., A_n)$ е релация, за която е изпълнено множеството от функционални зависимости F и R се декомпозира на две релации S ($B_1, B_2, ..., B_m$) и T ($C_1, C_2, ..., C_k$). Казваме, че декомпозицията е със съединение без загуба на функционалните зависимости, ако $F_1 \cup F_2 = F$, където F_1 и F_2 са проекциите на F съответно върху S и T.

Чрез подходящи декомпозиции, всяка схема на релация може да се декомпозира на няколко схеми, така че да са изпълнени следните условия: 1) Всички получени релации да са в BCNF; 2) Декомпозицията да е със съединение без загуба.

Стратегията за декомпозиция, която възприемаме, е следната.

21.1. Реляционен модел.

Нека $A_1 A_2 \dots A_n \rightarrow B_1 B_2 \dots B_m$ е нетривиална функционална зависимост и $\{A_1, A_2, \dots, A_n\}$ не е суперключ. Тогава декомпозираме релацията R на следните две релации:

1. Първата релация има атрибути $A_1, A_2, \dots, A_n, B_1, B_2, \dots, B_m$.
2. Втората релация има атрибути $A_1, A_2, \dots, A_n$ и всички останали атрибути на R , които не участват във функционалната зависимост.

Ако новополучените релации не са в BCNF, то към тях прилагаме същата процедура. При това, функционалните зависимости в новите релации се изчисляват чрез проектиране на ФЗ от изходната релация (т.е. функционалните зависимости в новите релации са точно онези, които следват от предишните ФЗ и в които участват само новите атрибути).

Процесът на декомпозиране ще е краен, тъй като винаги получаваме релации с по-малко атрибути, а всяка релация с два атрибута е в BCNF.

В общия случай декомпозицията в BCNF не е със съединение без загуба на функционалните зависимости. Съществува обаче алгоритъм за декомпозиция в 3НФ, който е със съединение без загуба и запазва ФЗ. Този алгоритъм в повечето случаи се справя с излишествата, породени от функционални зависимости.

Пример за съединение със загуба:

T	Employee	Project	Branch
	Brown	Mars	L.A.
	Green	Jupiter	San Jose
	Green	Venus	San Jose
	Hoskins	Saturn	San Jose
	Hoskins	Venus	San Jose

Функционалните зависимости са: $\text{Employee} \rightarrow \text{Branch}$ и $\text{Project} \rightarrow \text{Branch}$.

Декомпозиция на релацията T :

T1	Employee	Branch
	Brown	L.A.
	Green	San Jose
	Hoskins	San Jose

T2	Project	Branch
	Mars	L.A.
	Jupiter	San Jose
	Saturn	San Jose
	Venus	San Jose

След прилагане на естествено съединение резултатът е различен от първоначалната релация и информацията не може да бъде възстановена:

T	Employee	Project	Branch
	Brown	Mars	L.A.
	Green	Jupiter	San Jose
	Green	Venus	San Jose
	Hoskins	Saturn	San Jose
	Hoskins	Venus	San Jose
	Green	Saturn	San Jose
	Hoskins	Jupiter	San Jose

Не е задължително да се включва

Четвърта нормална форма.

21.1. Реляционен модел.

Излишеството, което произтича от МЗ, не може да се отстрани чрез привеждане в НФ на Бойс-Код. Необходима е по-строга НФ, наречена **четвърта нормална форма** (4НФ), която третира МЗ като ФЗ по отношение на декомпозицията, но не и по отношение на ключовете. В тази НФ всички МЗ са елиминирани, както всички ФЗ, които нарушават BCNF. В резултат, декомпозираните релации нямат излишък нито от ФЗ, нито от МЗ. В основата си четвъртата нормална форма е BCNF, но е приложена върху МЗ вместо върху ФЗ.

Релацията R е в **четвърта нормална форма**, ако за всяка нетривиална МЗ $A_1A_2...A_n \twoheadrightarrow B_1B_2...B_m$ е изпълнено, че $A_1A_2...A_n$ е суперключ.

Пример: Релацията, която разглеждахме досега, нарушава условието за 4НФ. Например, $name \twoheadrightarrow street \text{ city}$ е нетривиална МЗ, въпреки това $name$ не е суперключ. Всъщност единственият ключ за релацията са всички атрибути.

Декомпозиция към 4НФ: Ако $X \twoheadrightarrow Y$ нарушава 4НФ, правим следната декомпозиция: XY е едната от декомпозираните релации, а всички атрибути без $X \cup Y$ е другата.

Пример: Нека да продължим със същия пример. Видяхме, че $name \twoheadrightarrow street \text{ city}$ нарушава 4НФ. Правилото за декомпозиция ни кара да заменим схемата с пет атрибута със схема, която има само три атрибута от горната МЗ, и друга схема, състояща се от лявата страна ($name$) + атрибутите, които не се появяват в тази МЗ. Тези атрибути са $title$ и $year$. Така след декомпозирането получаваме следните две схеми: $R (name, street, city)$ и $S(name, title, year)$. Във всяка от схемите няма нетривиални многозначни (или функционални) зависимости, така че те са в 4НФ.

Обобщение на нормалните форми.



Свойство	3НФ	BCNF	4НФ
Отсъствие на FD излишество	В повечето случаи	Да	Да
Отсъствие на MVD излишество	Не	Не	Да
Запазване на FD	Да	Не винаги	Не винаги
Запазване на MVD	Не винаги	Не винаги	Не винаги

BCNF (следователно и 4НФ) елиминират излишеството и други аномалии, които са причинени от ФЗ, докато само 4НФ елиминира допълнителния излишък, причинен от наличието на нетривиални МЗ, които не са ФЗ. Често 3НФ е достатъчна за премахването на този излишък, но има примери, където не е. BCNF не гарантира запазването на ФЗ и никоя от нормалните форми не гарантира запазване на МЗ, въпреки че в типични случаи зависимостите се запазват.