

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 реляционната схема са множество, а не списък, ние трябва да укажем стандартна подредба на атрибутите един спрямо друг. В реляционния модел проектът се състои от една или няколко реляционни схеми. Съвкупността от всичките тези реляционни схеми, с които се описват класовете от обекти и връзките между тях, се нарича **реляционна 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 реляционната БД.

Реализация на реляционната база от данни.

Процесът на **проектиране** на една реляционна 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 реляционни 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 реляционна схема включва ключовите атрибути на всеки от тези класове от обекти.

Видове операции върху реляционната база от данни. Заявки към реляционната 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 реляционната 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 операциите 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**.

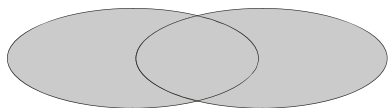
Нека D_1, D_2, \dots, D_n са n области. Разглеждаме списъка от атрибути $A = A_1, A_2, \dots, A_n$, при който атрибутът A_i взема стойности от областта D_i , $i = 1, 2, \dots, n$. Нека B е друг списък от m атрибута B_1, B_2, \dots, B_m . Релациите $R(A_1, A_2, \dots, A_n)$ и $S(B_1, B_2, \dots, B_m)$ 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.

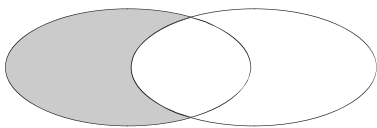
Освен на основни и допълнителни, операциите в релационната алгебра могат и да се класифицират по следния начин:

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

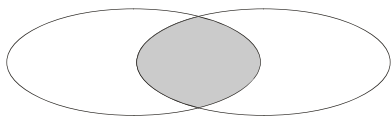
- **Обединение** (union) – бинарна (двучленна), комутативна, асоциативна операция – $R \cup S$



- **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

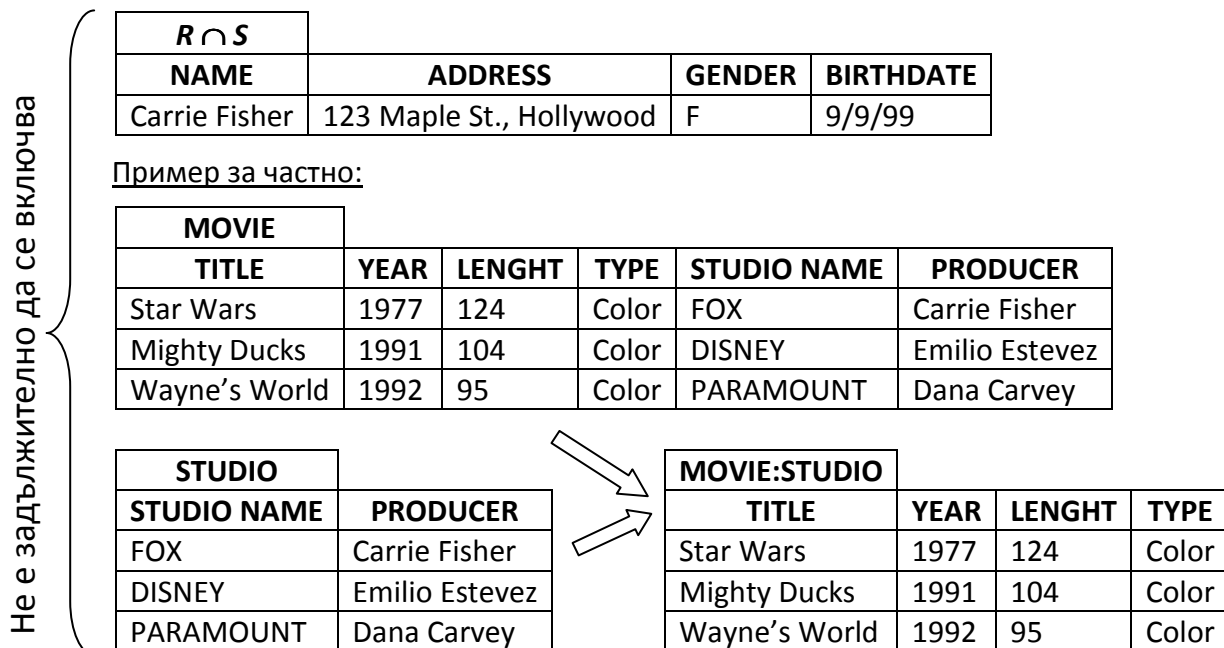
S			
NAME	ADDRESS	GENDER	BIRTHDATE
Carrie Fisher	123 Maple St., Hollywood	F	9/9/99
Harrison Ford	789 Palm Dr., Beverly Hills	M	7/7/77

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

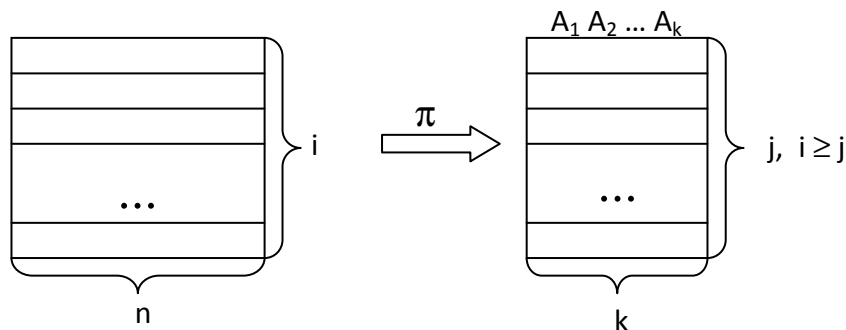
$R - S$			
NAME	ADDRESS	GENDER	BIRTHDATE
Mark Hamil	456 Oak Rd., Brentwood	M	8/8/88

Не е задължително да се включва



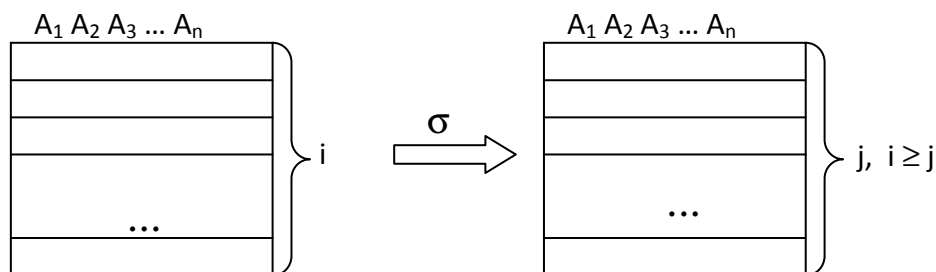
2. Операции, които отстраняват части от релациите

• **Проекция** (хоризонтална рестрикция, намаляване в ширина): Нека P е n -членна релация с релационна схема $P(A_1, A_2, \dots, A_n)$. Проекцията на релацията P 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 P 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 P е n -членна релация с релационна схема $P(A_1, A_2, \dots, A_n)$. Селекцията на релацията P 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

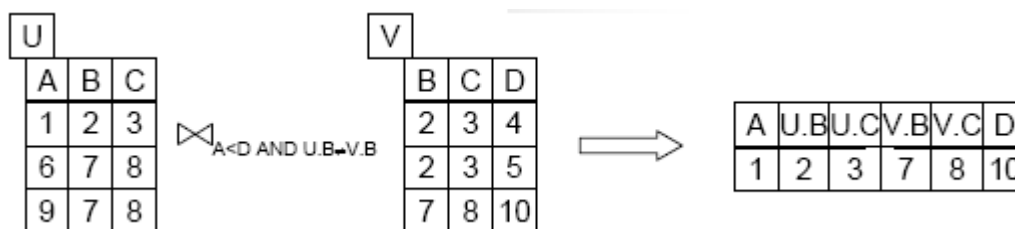
3. Операции, които комбинират кортежи от две или повече релации

• **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	\times	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

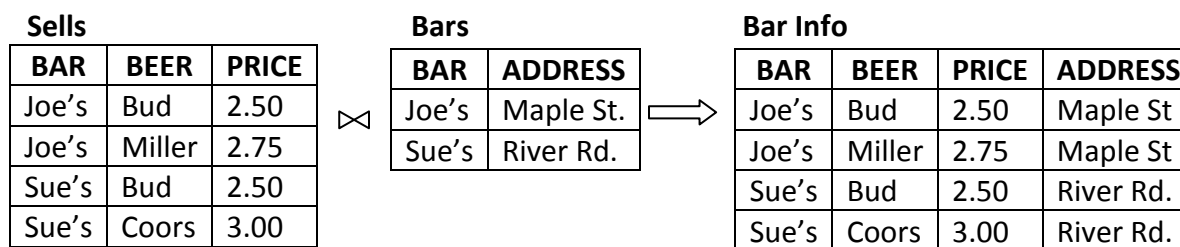
се извършва на два етапа: първо се прави декартово произведение на двете релации; след това се прави селекция върху получената релация по отношение на условието C.



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

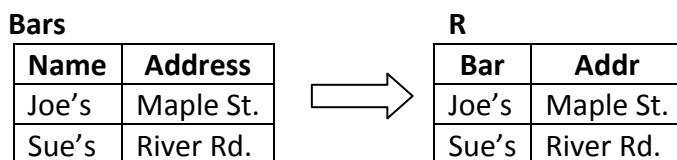
• **Естествено съединение** (natural join) – $R \bowtie S$: Разширение на еквисъединението, при което се извършва свързване по всички атрибути с еднакви имена. Автоматично се отстранява повтарящата се колона.

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 реляционните оператори.

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