

9. Бази от данни. Релационен модел на данните.

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

Примерни задачи: Съставяне на SQL заявки, DDL и DML команди.

База данни

Всяка база от данни е изградена на базата на модел, който определя логическата ѝ структура и по какъв начин данните се съхраняват, организират и манипулират. Някой от моделите са същност-въръзки (Entity-Relationship), обектно ориентиран модел (Object-Oriented) и релационен модел (Relational).

Базите от данни са създадени да съхраняват данни за голям период от време. Данните в тях се управляват от система за управление на бази от данни (СУБД), като например MySQL, Microsoft SQL Server, Oracle DB, Maria DB и други.

Релационен модел на данните: домейн, релация, кортежи, атрибути, схема на релация, схема на релационна база от данни, реализация на релационна база от данни, видове операции върху релационна база от данни, заявки към релационна база от данни

Релационният модел се базира на таблици, пример за които е фиг. 1 показана по-долу.

Movies

Title	Year	Length	Genre
Swept Away	2002	89	Drama
Gattaca	1997	106	Sci-Fi

фиг. 1

Релация

Релационният модел ни дава един-единствен начин да представим данни: като двуизмерна таблица, наречена релация. Таблицата по-горе ще наричаме релацията Movies. Всеки ред представлява филм и всяка колона представлява атрибут/свойство на филм.

Атрибути

Колоните на релация са именувани с атрибути. На фиг. 1 атрибутите са заглавие, година, продължителност и жанр (Title, Year, Length, Genre). Всеки домейн е асоцииран с атрибут, който описва значението на записите в колоната под него.

Схема на релация и схема на релационна база от данни

Името на релацията и наборът от атрибути за релация се нарича схема на релацията. Изписва се като името на релацията се слага първо и след него в скоби се изреждат атрибутите). Например, схемата на релацията от фиг. 1 е Movies(title, year, length, genre). Атрибутите в релация нямат наредба и в този смисъл те са множество, а не списък, но когато говорим за схема, трябва да имаме предвид, че тя задава някаква стандартна наредба по подразбиране на дадените атрибути. Релационният модел база данни се състои от една или повече релации. Множеството от схеми на релациите на база от данни се нарича релационна схема на база от данни или просто схема на база от данни.

Кортежи

Редовете на релация, различни от заглавния ред, съдържащ имената на атрибутите се наричат кортежи. Кортежът има един компонент за всеки атрибут на релацията. Например, първият от трите кортежи от фиг.1 има четири компонента (Swept Away, 2002, 89, Drama). Тъй като, когато кортеж е даден в изолация – имената на атрибутите не съществуват, то е редно да даваме компонентите му в реда, в който са атрибутите.

Домейн

Релационният модел изисква всеки компонент на всеки кортеж да бъде атомарен. Тоест трябва да е от някакъв елементарен (не разбиваем) цялостен тип. Не е разрешено

стойността да бъде под формата на структура от данни, множество, масив или всеки друг тип, който може да се разбие логически на отделни по-малки компоненти. Стойностите на компонентите във всеки кортеж трябва да принадлежат на домейна на съответната колона. Например филмите от примера от фиг. 1 трябва да имат първи и четвърти компонент – стринг, втори и трети компонент – цяло число. Възможно е да добавим домейна или типа на данните за всеки атрибут в релационната схема. Например: `Movies(title:string, year:int, length: int, genre: string)`

Реализация на релационна 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 бъдат съхранявани;
- Определяне на връзките между обектите;
- Определяне на ключовете и на свързващите колони;
- Определяне на ограниченията върху обектите и връзките между тях;
- Отстраняване на евентуални недостатъци (например излишества);
- 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. Всеки кортеж от релацията представя конкретен обект от класа. Ключовият атрибут или списък от ключовите атрибути на класа (тези атрибути, които еднозначно определят даден кортеж) 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 релационна база от данни. Заявки към релационна база от данни

1. Операции, които променят броя кортежите или ги видоизменят. Това са операциите **SELECT**, **INSERT**, **UPDATE**, **DELETE** и те са част от **DML** (Data Manipulation Language). Чрез тях се извличат, добавят, променят или изтриват кортежи. След тяхното изпълнение промените не се запазват. За да се запазят е необходимо да се направи **COMMIT**. Примери:

SELECT * FROM Movies;

Извлича всички кортежи (филми) от релацията `Movies`, като в резултатната таблица ще са всички атрибути на релацията.

Title	Year	Length	Genre
Swept Away	2002	89	Drama
Gattaca	1997	106	Sci-Fi

SELECT Title, Year FROM Movies;

Извлича всички кортежи от релацията `Movies`, но в резултатната таблица ще са само атрибутите `title` и `year`.

Title	Year
Swept Away	2002
Gattaca	1997

SELECT Title, Year FROM Movies WHERE Year > 2000 AND Title LIKE "S%";

Извлича всички филми, чиито атрибут `year` е със стойност по-голяма от 2000 и атрибут `title` започва с буквата `S`. В резултатната таблица ще са селектирани само атрибутите `title` и `year`.

Title	Year
Swept Away	2002

INSERT INTO Movies(Title, Year, Genre) VALUES ("Heat", 1995, 170, Crime);

Добавя нов филм в релацията с посочените атрибути. Липсващият атрибут `length` е със стойност `NULL`.

Title	Year	Length	Genre
Swept Away	2002	89	Drama
Gattaca	1997	106	Sci-Fi
Heat	1995	170	Crime

UPDATE Movies **SET** Length = 170
WHERE Title **LIKE** "Heat";

Актуализира дължината на филма с име Heat, като я преправя да бъде 170.

Title	Year	Length	Genre
Swept Away	2002	89	Drama
Gattaca	1997	106	Sci-Fi
Heat	1995	170	Crime

DELETE FROM Movies **WHERE** Year < 2000 ;

Изтрива всички филми, които са пуснати преди 2000 г.

Title	Year	Length	Genre
Swept Away	2002	89	Drama

2. Операции, които променят схемата на релацията. Това са операциите **CREATE**, **DROP**, **ALTER** и те са част от **DDL** (Data Definition Language). Те служат за дефиниране и промяна на структурата на релацията. Когато се изпълни една DDL операция, промените се прилагат незабавно. Примери:

CREATE TABLE Employees(
 Id **INTEGER** **PRIMARY KEY**,
 FirstName **VARCHAR**(45) **NULL**,
 LastName **VARCHAR**(45) **NOT NULL**,
 Age **INTEGER** **NULL**
);

Създава таблица (релация) със следната схема: Employees(id, first_name, last_name, age), ключов атрибут id и атрибут age от целочислени типове и атрибути first_name и last_name с ограничение на дължината от 45 символа.

DROP TABLE Employees;

Изтрива таблицата employees.

ALTER TABLE Employees **ADD** City
VARCHAR(45);

Добавя атрибут city в релацията employees.

ALTER TABLE Employees **DROP COLUMN** City;

Премахва атрибута city от релацията employees

Релационна алгебра: основни (обединение, разлика, декартово произведение, проекция, селекция) и допълнителни (сечение, частно, съединение, естествено съединение) операции

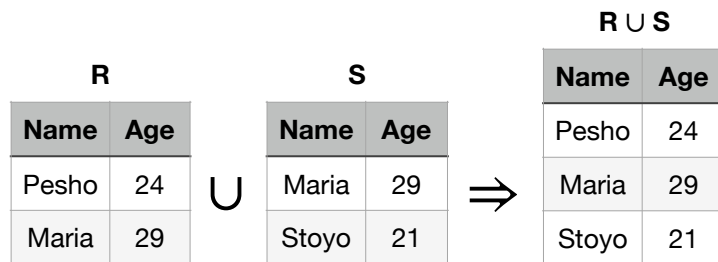
Операциите на традиционната релационна алгебра попадат в 4 широки класа:

1. Прилагат се обичайните операции над множества – обединение, сечение и разлика върху релации;
2. Операции, които премахват част от релацията: селекция, която премахва някои от редовете (кортежите) и проекция, която премахва някои от колоните (атрибутите);
3. Операции, които комбинират кортежите на две релации, както декартовото произведение, което комбинира кортежите на две релации по всички възможни начини, така и различни видове на съединителни операции, които избирателно сдвояват кортежи от две релации.
4. Операцията наречена преименуване, която не засяга кортежите на релация, но променя схемата на релацията, т.е. имената на атрибутите и/или името на самата релация.

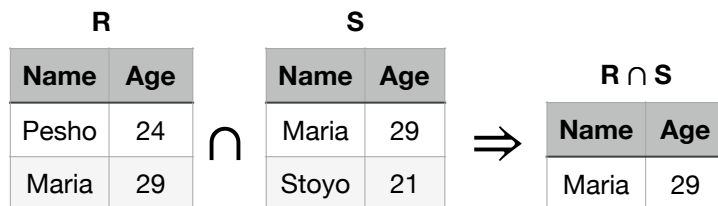
Обикновено ще наричаме изрази на релационната алгебра като заявки.

Операциите обединение, сечение и разлика се прилага върху релации, които имат еднакви схеми и еднакви домейни за всеки атрибут. Преди да се изчисли резултата от операцията, колоните на двете релации трябва да бъдат подредени така, че реда на атрибутите да е един и същ. Преименуващият оператор е много подходящ, когато искаме да приложим някой от операторите обединение, сечение и разлика върху две релации, които имат равен брой атрибути с еднакви домейни, но с различни имена.

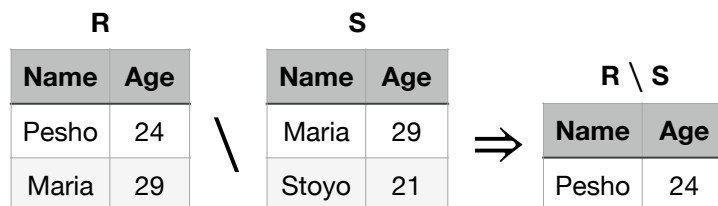
- $R \cup S$ е обединението на релациите R и S . Операцията е асоциативна и комутативна. Резултатът е множеството от елементи (кортежи), които са или в R или в S или и в двете. Елемент се появява само веднъж в обединението, дори и да присъства и в R и в S ;



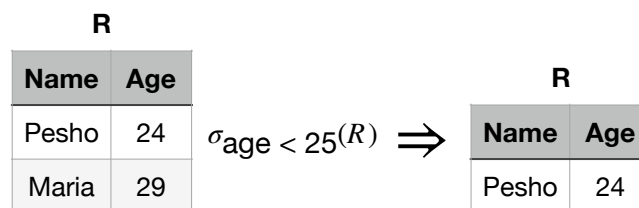
- $R \cap S$ е сечението на релациите R и S . Операцията е асоциативна и комутативна. Резултатът е множеството от елементи (кортежи), които са едновременно и в R и в S . Общият елемент се появява само веднъж в сечението.



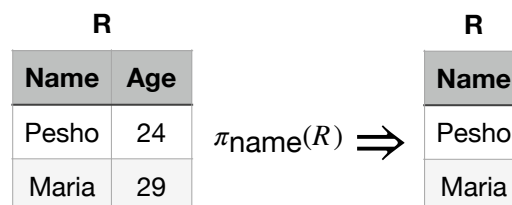
- $R \setminus S$ е разликата на релациите R и S . Операцията не е комутативна (резултатът се различава, ако разменим реда на операндите). $R \setminus S$ е множеството от всички кортежи от R , които не са в S .



- Селекция. Този оператор приложен към релация R връща резултат, който е подмножество от кортежите на R , които удовлетворяват някакъв предварително зададен предикат.

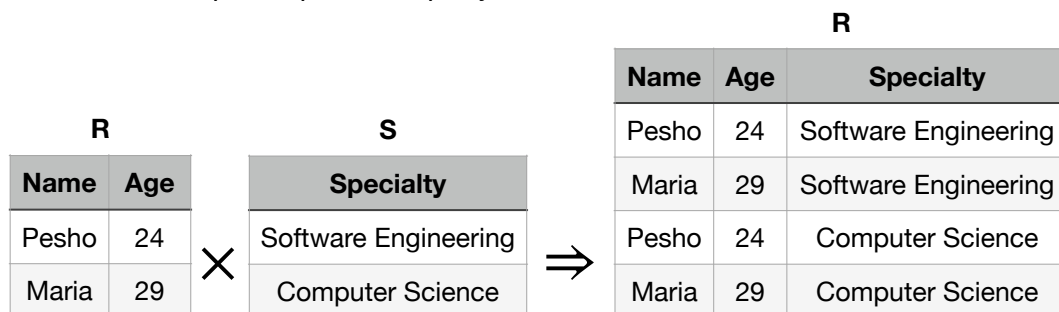


- Проекция. Проектиращият оператор се използва когато искаме да създадем нова релация от релацията R , която използва само част от нейните атрибути.

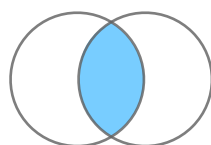
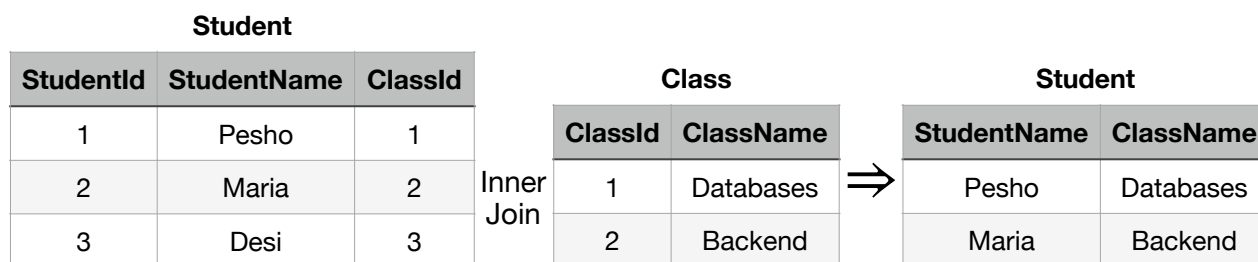


- $R \times S$ е декартовото произведение на релациите R и S . Резултатът е релация, която съдържа кортежи с по-голяма дължина, които комбинират всеки кортеж от R с всеки кортеж от S . По конвенция, атрибутите на релацията от ляво на операнда предшестват атрибутите на релацията от дясно на операнда в резултатната релация. Ако някои от атрибутите на R и S имат общи имена, тогава е необходимо да ги преименуваме, така че всички имена на атрибути да са единствени. За да няма двусмислие относно атрибут A ,

който е едновременно и в R и в S , се използват означенията $R.A$ и $R.S$ в схемата $R \times S$, за да се детерминира кой атрибут от коя схема е взет.

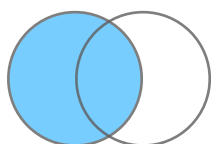
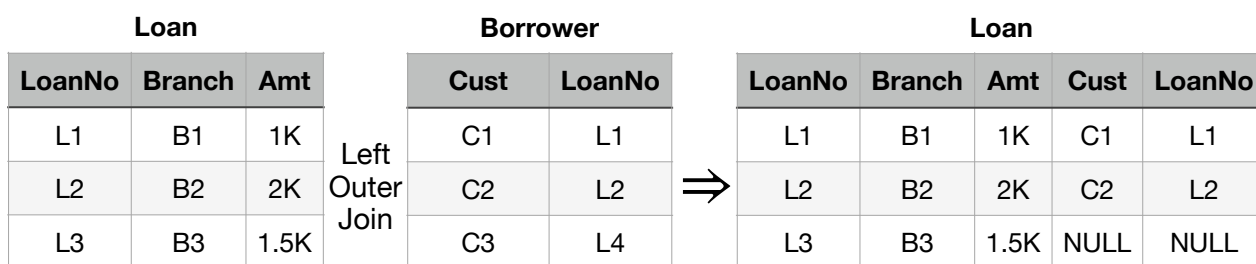


- Вътрешно съединение. Връща нова релация, която обединява кортежи, които имат общи стойности.



```
SELECT studentName, className
FROM Student
INNER JOIN Class
ON Class.ClassId = Student.ClassId
```

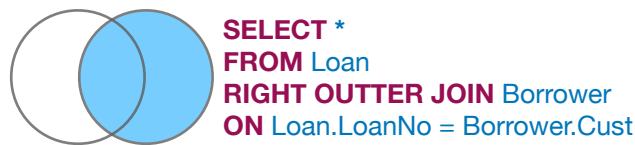
- Външно ляво съединение. Връща нова релация, която има всички кортежи от лявата релация и само тези кортежи от дясната релация, които имат общи стойности с кортежи от лявата релация.



```
SELECT *
FROM Loan
LEFT OUTER JOIN Borrower
ON Loan.LoanNo = Borrower.Cust
```

- Външно дясно съединение. Дуално на външното ляво съединение, но като взима всички кортежи от дясната релация и само тези от лявата, които имат общи стойности с кортежи от дясната релация.

Loan			Right Outer Join	Borrower		⇒	Loan				
LoanNo	Branch	Amt		Cust	LoanNo		LoanNo	Branch	Amt	Cust	LoanNo
L1	B1	1K		C1	L1		L1	B1	1K	C1	L1
L2	B2	2K		C2	L2		L2	B2	2K	C2	L2
L3	B3	1.5K		C3	L4		NULL	NULL	NULL	C3	L4



- Външно цялостно съединение. Взява кортежите от двете релации, като липсващите атрибути ги запълва с NULL.

Loan			Full Outer Join	Borrower		⇒	Loan				
LoanNo	Branch	Amt		Cust	LoanNo		LoanNo	Branch	Amt	Cust	LoanNo
L1	B1	1K		C1	L1		L1	B1	1K	C1	L1
L2	B2	2K		C2	L2		NULL	NULL	NULL	C2	L2
L3	B3	1.5K		C3	L4		L3	B3	1.5K	NULL	NULL



- Естествено съединение. Естественото съединение на релациите $R(a_1, \dots, a_m)$ и $S(b_1, \dots, b_n)$ по отношението на условието C е $m + n$ членна релация с релационна схема $R(a_1, \dots, a_m, b_1, \dots, b_n)$, която е такова подмножество на декартовото произведение $R \times S$, в което всеки кортеж отговаря на условието C . Бележим съединението с нотацията $R \bowtie_C S$. Този вид съединение се извършва на два етапа: първо се прави декартовото произведение на двете релации и след това се прави селекция върху получената релация по отношение на условието C . Пример:

R × S					S ⋈ _{C<D and R.B≠S.B} R				
R		S			A	R.B	S.B	C	D
1	2	2	5	6	1	2	2	5	6
1	2	4	6	8	1	2	4	6	8
1	2	9	10	9	1	2	9	10	9
3	4	2	5	6	3	4	2	5	6
3	4	4	6	8	3	4	4	6	8
3	4	9	10	9	3	4	9	10	9