

Лекція 7-2020_2021

- На попередній лекції
- Маніпуляційна частина реляційної моделі.
- Реляційна алгебра, операції
 - Вибірка
 - Проекція
 - Декартовий добуток. З'єднання.
 - Об'єднання, перетин, різниця
 - Додаткові операції
- Реляційне числення

На прошлой лекции

Проектирование РБД на основе нормализации – метод последовательных приближений к удовлетворительному набору схем отношений. Метод нормализации – декомпозиция отношения, находящегося в предыдущей нормальной форме, на два или более отношений, которые удовлетворяют требованиям следующей нормальной формы.

Правильные декомпозиции отношения обратимы, т.е. имеется возможность собрать исходное отношение из декомпозированных отношений без потери информации. Такие декомпозиции называются декомпозициями без потерь.

На прошлой лекции

Аномалия – ситуацию в таблицах БД, которая приводит к противоречиям в БД либо существенно усложняет обработку данных.

- Если удаляя факты, относящиеся к одной сущности, удаляются факты, относящиеся к другой сущности – аномалия удаления.
- Если не возможно записать в таблицу некоторый факт об одной сущности, не указав дополнительно некоторый факт о другой сущности – аномалия вставки.
- Если изменение значения одного данного может повлечь за собой просмотр всей таблицы и соответствующее изменение некоторых других записей таблицы – аномалия модификации.

На предыдущей лекции

1. **1NF** – выполнены все требования, чтобы таблица была отношением;
2. **2NF** – отношение в **1NF** и устранены частичные функциональные зависимости;
3. **3NF** – отношение в **2NF** и нет транзитивных зависимостей;
4. **BCNF** – отношение в **3NF** и имеет в качестве детерминанта один возможный ключ отношения;
5. **4NF** – отношение в **3NF (BCNF)** и устранены многозначные зависимости;
6. **5NF** – отношение в **4NF** и необходима декомпозиция из-за наличия нетривиальных аномалий обновления.

Три части реляционной модели данных

Трактовка реляционной модели данных – Кристофер Дейт. Реляционная модель состоит из трех частей:

структурной, целостной и манипуляционной.

1. Структурная часть: единственной родовой структурой данных, используемой в реляционных БД, является нормализованное n-арное отношение.

Определяются понятия *доменов, атрибутов, кортежей, заголовка, тела и переменной отношения*.

2. Манипуляционная часть – *два основных механизма манипулирования реляционными БД (реляционная алгебра и реляционное исчисление)*.

3. Целостная часть – целостность сущности и целостность ссылок.

Манипуляционная часть

Реляционная алгебра и реляционное исчисление. На основе одного или нескольких отношений можно создать другое отношение без изменения исходных.

- Язык реляционной алгебры – процедурный, все кортежи обрабатываются одной командой, без циклов.

- Язык реляционного исчисления – непроцедурный, описательный. Запрос с помощью предиката первого порядка (высказывания в виде функции). При подстановке значений аргументов функция становится выражением, называемым суждением, которое может быть истинным или ложным.

Алгебра Кодда: выборка, проекция, объединение, вычитание и произведение;.

Основы реляционной алгебры.

Отношение R

A1	A2
A	1
A	2
B	1
B	3
B	4

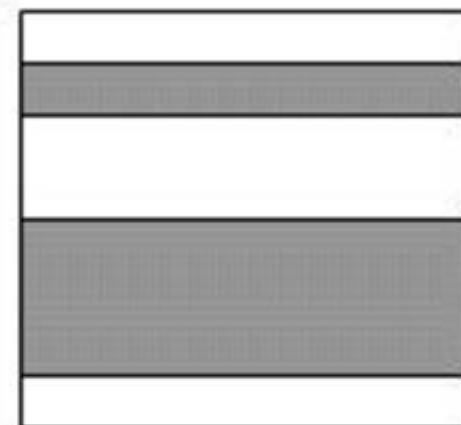
Отношение S

B1	B2
1	h
2	g
3	h

Операция выборки

Построение горизонтального подмножества

Выборка работает с **одним** отношением R и определяет результирующее отношение, которое содержит только те кортежи (строки) отношения R, которые удовлетворяют заданному условию F (предикату).



$\sigma_F(R)$ или $\sigma_{\text{предикат}}(R).$

Например: выборка $\sigma_{(a2=1)}(R) = \{(a, 1), (b, 1)\}$

SQL: SELECT a1, a2 FROM R WHERE a2 = 1;

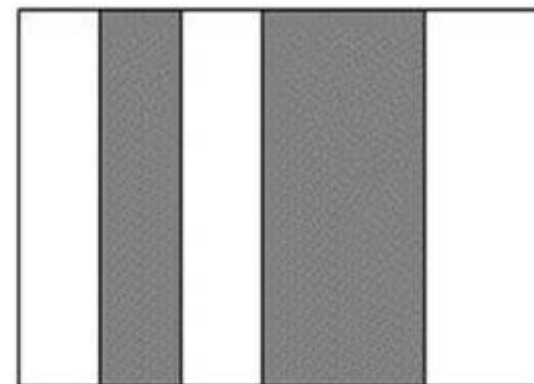
Результат:

A1	A2
A	1
B	1

Операция проекции

Построение вертикального подмножества

Проекция работает с одним отношением R и определяет новое отношение, которое содержит вертикальное подмножество отношения R, создаваемое посредством извлечения значений указанных атрибутов и исключения из результата строк-дубликатов



$$\Pi_{a_1, a_2, \dots, a_n} (R)$$

Например: проекция $\Pi_{b_2}(S) = \{(h), (g)\}$

SQL: `SELECT b2 FROM S;`

Результат:

B2
h
g
h

Декартово произведение

Декартово

произведение $R \times S$

двух отношений

определяет новое

отношение -

результат

конкатенации

каждого кортежа

из отношения R с

каждым кортежем

из отношения S.

$R1 \rightarrow N$ стр и K стб

$R2 \rightarrow M$ стр и L стб

$R1 * R2 \rightarrow$

$N * M$ строк и

$K + L$ столбцов

$R \times S = \{(a, 1, 1, h), (a, 2, 1, h), (b, 1, 1, h), \dots\}$

SQL: SELECT R.a1, R.a2, S.b1, S.b2 FROM R, S

A1	A2	B1	B2
a	1	1	h
a	1	2	g
a	1	3	h
a	2	1	h
a	2	2	g
a	2	3	h
b	1	1	h
b	1	2	g
b	1	3	h
b	3	1	h
b	3	2	g
b	3	3	h
b	4	1	h
b	4	2	g
b	4	3	h

Соединение

Вместо декартова произведения обычно используется операция соединения, которая является производной от операции декартова произведения. **Соединение** – это процесс, когда две или более таблицы объединяются в одну.

Операция **тета-соединения** $R \bowtie_F S$ определяет отношение, которое содержит кортежи из декартова произведения отношений R и S , удовлетворяющие предикату F . Предикат F имеет вид $R.a_i \Theta S.b_j$, где вместо Θ может быть указан один из операторов сравнения ($>$, $>=$, $<$, $<=$, $=$, $<>$).

Соединение по эквивалентности

Если предикат F содержит только оператор равенства ($=$), то соединение называется **соединением по эквивалентности**.

$R \bowtie_F S, F=(R.a2=S.b1)$			
R.a1	R.a2	S.b1	S.b2
a	1	1	h
a	2	2	g
b	3	3	h
b	1	1	h

SQL:

```
SELECT R.a1, R.a2, S.b1, S.b2
FROM R, S
WHERE R.a2=S.b1;
```

или

```
SELECT R.a1, S.b1, S.b2
FROM R INNER JOIN S
ON R.a2=S.b1;
```

Левое (правое) внешнее соединение

Левым внешним соединением $R \bowtie S$ называется соединение, при котором кортежи отношения R , не имеющие совпадающих значений в общих столбцах отношения S , также включаются в результат.

$R \bowtie S, F=(R.a2=S.b1)$			
R.a1	R.a2	S.b1	S.b2
a	1	1	h
a	2	2	g
b	1	1	h
b	3	3	h
b	4	null	null

SQL:

```
SELECT R.a1, R.a2, S.b1, S.b2  
FROM R LEFT JOIN S  
ON R.a2=S.b1; (все из первого)
```

Правое внешнее соединение:

```
SELECT R.a1, R.a2, S.b1, S.b2  
FROM R RIGHT JOIN S ON  
R.a2=S.b1; (все из второго)
```

Полусоединение

Операция полусоединения $R \bowtie_F S$ определяет отношение, содержащее те кортежи отношения R, которые входят в соединение отношений R и S.

$R \bowtie_F S, F=(R.a2=S.b1)$	
R.a1	R.a2
a	1
a	2
b	3
b	1

SQL:

```
SELECT R.a1, R.a2 FROM R,S  
WHERE R.a2=S.b1
```

или

```
SELECT R.a1, R.a2  
FROM R INNER JOIN S  
ON R.a2=S.b1
```


Объединение, пересечение, разность

Совместимость отношений по объединению:

два отношения совместимы по объединению в том и только в том случае, когда обладают **одинаковыми заголовками**. Это означает, что оба отношения содержат один и тот же набор имен атрибутов.

Если два отношения совместимы по объединению, то при обычном выполнении над ними операций **объединения, пересечения и взятия разности** результатом операции является отношение с корректно определенным заголовком, совпадающим с заголовком каждого из отношений-операндов

Объединение, пересечение, разность

Два отношения СЛУЖАЩИЕ_В_ПРОЕКТЕ_1 (Пр1) и СЛУЖАЩИЕ_В_ПРОЕКТЕ_2 (Пр2) с одинаковыми схемами {НОМЕР, ИМЯ, ЗРП, ОТД}

Пр1

Пр2

Номер	Имя	Зрп	Отд		Номер	Имя	Зрп	Отд
2934	Иванов		310		2934	Иванов		310
2935	Петров		310		2935	Петров		310
2936	Сидоров		313		2939	Сидоренко		313
2937	Федоров		310		2940	Федоренко		310
2938	Иванова		315		2941	Иваненко		315

Объединение (UNION)

Служащие, участвующие хоть в одном проекте.

Номер	Имя	Зрп	Отд
2934	Иванов		310
2935	Петров		310
2936	Сидоров		313
2937	Федоров		310
2938	Иванова		315
2939	Сидоренко		313
2940	Федоренко		310
2941	Иваненко		315

SQL:
SELECT Пр1.Номер,
Пр1.Имя, Пр1.Зрп,
Пр1.отд FROM Пр1
UNION
SELECT Пр2.Номер,
Пр2.Имя, Пр2.Зрп,
Пр2.Отд FROM Пр2;

Пересечение (INTERSECT)

Служащие, участвующие в обоих проектах.

Номер	Имя	Зрп	Отд
2934	Иванов		310
2935	Петров		310

SQL:

```
SELECT Пр1.Номер,  
Пр1.Имя, Пр1.Зрп,  
Пр1.Отд  
FROM Пр1, Пр2  
WHERE  
(Пр1.Номер= Пр2.Номер)  
AND (Пр1.Имя= Пр2.Имя)  
AND (Пр1.Зрп= Пр2.Зрп)  
AND (Пр1.Отд= Пр2.Отд)
```

Взятие разности (MINUS)

Служащие, участвующие только в первом проекте.

Номер	Имя	Зрп	Отд
2936	Сидоров		313
2937	Федоров		310
2938	Иванова		315

SQL:

```
SELECT Пр1.Номер,  
Пр1.Имя, Пр1.Зрп, Пр1.Отд  
FROM Пр1  
WHERE Пр1.Номер  
NOT IN (  
SELECT Пр2.Номер  
FROM Пр1, Пр2  
WHERE  
(Пр1.Номер= Пр2.Номер));
```

Дополнительные операции реляционной алгебры

Кристофер Дейт: переименование, расширение, вставка, обновление и удаление.

Переименование – изменение имени атрибута отношения. $\langle \text{СтароеИмя} \rangle \text{ AS } \langle \text{НовоеИмя} \rangle$.

Расширение – добавление нового атрибута, значение которого получается путем вычислений.

$\text{Табл1.Атр1} * \text{Табл1.Атр2 AS Имя1}$.

Дополнительные операции реляционной алгебры

Вставка – добавление одной или группы строк в отношение `INSERT <выражение-источник> INTO <выражение-цель>`

Обновление – изменение значений атрибута

`UPDATE <выражение-цель> <список элементов>`

Удаление – удаление одной или нескольких строк

`DELETE <выражение-цель>.`

Подведение итогов – вертикальные или групповые вычисления:

`SELECT Sum (Пр1.Зрп) AS Вся_Зрп_Пр1 FROM Пр1;`

Основы реляционного исчисления

Реляционное исчисление – высказывания реализуются в виде предиката первого порядка (функции). При подстановке значений аргументов значение функции (суждением) может быть истинным или ложным.

В реляционном исчислении кортежей задача состоит в нахождении таких кортежей, для которых предикат является истинным.

Предикат:

Wff – **well formulated formula** (правильно построенная формула)

Основы реляционного исчисления

$wff ::= \langle \text{условие} \rangle \mid \text{NOT } wff \mid$
 $\quad \langle \text{условие} \rangle \text{ AND } wff \mid$
 $\quad \langle \text{условие} \rangle \text{ OR } wff \mid$
 $\quad \text{IF } \langle \text{условие} \rangle \text{ THEN } wff \mid$
 $\quad \text{EXISTS } \langle \text{переменная} \rangle (wff) \mid$
 $\quad \text{FORALL } \langle \text{переменная} \rangle (wff) \mid$
 $\quad (wff)$

$\langle \text{условие} \rangle$ – это либо формула **wff**, заключенная в скобки, либо простое сравнение вида:

$\langle \text{операнд } 1 \rangle \square \langle \text{операнд } 2 \rangle$

$\square \rightarrow =, <, >, \geq, \leq$

NOT, **AND**, **OR** – (не, и, или),

IF **THEN** – если ,то

Основы реляционного исчисления

EXISTS, FORALL – кванторы

(**EXISTS** – квантор существования,

FORALL – квантор всеобщности)

Exists f (x) – существует по крайней мере одно такое значение переменной **x**, что вычисление формулы **f** дает значение истина.

ForAll f (x) – для всех значений переменной **x**, вычисление формулы **f** дает значение истина