

Практическое задание 5

Язык SQL

Операция соединения по двум отношениям (Join). ДЕЛЕНИЕ

Соединение - это процесс, когда две или более таблицы объединяются в одну с изменением схемы отношения.

JOIN — оператор языка SQL, который является реализацией операции соединения реляционной алгебры. Входит в раздел FROM операторов SELECT, UPDATE или DELETE.

Операция соединения, как и другие бинарные операции, предназначена для обеспечения выборки данных из двух таблиц и включения этих данных в один результирующий набор. Отличительной особенностью операции соединения является следующее:

1. в схему таблицы-результата входят столбцы обеих исходных таблиц (таблиц-операндов), то есть схема результата является «сцеплением» схем операндов;
2. каждая строка таблицы-результата является «сцеплением» строки из одной таблицы-операнда со строкой второй таблицы-операнда.

При необходимости соединения не двух, а нескольких таблиц, операция соединения применяется несколько раз (последовательно).

Синтаксис

```
SELECT имя_поля [... n]
FROM
имя_таблицы_1
[INNER | [LEFT | RIGHT | FULL] OUTER | CROSS ] JOIN
имя_таблицы_2
ON <условие_соединения>;
```

типы операций соединения:

- тета-соединение $R \bowtie_F S$;
- соединение по эквивалентности $R \bowtie_{=S}$;
- естественное соединение $R \bowtie S$;
- внешнее соединение $R \Join S$; $R \Join_{\supset} S$;
- полусоединение $R \Join_{\supseteq} S$.

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

Операция тета-соединения

Тета-соединение $R \bowtie_F S$ определяет отношение, которое содержит кортежи из *декартова произведения* отношений R и S , удовлетворяющие предикату F .

Предикат F имеет вид $R.ai \ \Theta \ S.bj$, где вместо Θ может быть указан один из операторов сравнения ($>$, $>=$, $<$, $<=$, $=$, $<>$).

Соединение по эквивалентности - предикат F содержит только оператор равенства ($=$)

Внутреннее соединение - т.ит.т, когда в предложении **WHERE** сравниваются значения полей из разных таблиц. При этом строится *декартово произведение* строк таблиц, а из полученного набора данных отбираются записи, удовлетворяющие условиям соединения.

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

или

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

Таблица 2

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

Естественное соединение

Естественным соединением называется соединение по эквивалентности двух отношений **R** и **S**, выполненное по всем общим атрибутам, из результатов которого исключается по одному экземпляру каждого общего атрибута.

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

Дана БД, моделирующая работу торговой фирмы со схемой БД(Товар, Клиент, Сделка)

Товар(ID_тов, Назв, Цена, Тип, сорт, гор_Тов)
Клиент(ID_кл, Фирма, Фам, Имя, Отч, гор_Кл, Тел)
Сделка(ID_сд, ID_Тов, ID_Кл, Кол-во, Дата))

Пример 7.1. Вывести **всю** информацию о проданных товарах (См.пред. занятия).

Можно создать вложенные соединения, добавив третью таблицу к результату объединения двух других таблиц.

Пример 7.2. Получить сведения о товарах, дате сделок, количестве проданного товара и покупателей.

Использование общих имен таблиц для идентификации столбцов неудобно из-за их громоздкости. Каждой таблице можно присвоить какое-нибудь краткое обозначение, **псевдоним**.

Пример 7.3. Получить сведения о товарах, дате сделок, количестве проданного товара и покупателях. В запросе использовать псевдонимы таблиц.

Внешнее соединение похоже на внутреннее, но в результирующий набор данных включаются также записи **ведущей** таблицы соединения, которые объединяются с пустым множеством записей другой таблицы.

Какая из таблиц будет **ведущей**, определяет вид соединения. **LEFT** - левое внешнее соединение, ведущей является таблица, расположенная слева от вида соединения; **RIGHT** - правое внешнее соединение, ведущая таблица расположена справа от вида соединения.

Таблица 3

$R \bowtie S, F=(R.a2=S.b1)$			
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

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

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

Правое внешнее соединение $R \rightarrow S$ - в результирующем отношении содержатся все кортежи правого отношения.

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

Полное внешнее соединение - в его результирующее отношение помещаются все кортежи из обоих отношений, а для обозначения несовпадающих значений кортежей в нем используются определители **NULL**.

Таблица 4.

$R \bowtie_F S, F=(R.a2=S.b1)$

R.a1	R.a2
a	1
a	2
b	3
b	1

Пример 7.4. Вывести информацию о всех товарах. Для проданных товаров будет указана дата сделки и количество. Для непроданных эти поля останутся пустыми.

Полусоединение - определяет отношение, содержащее те кортежи отношения **R**, которые входят в **соединение** отношений **R** и **S**.

```
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
```

Пример 7.5

Построить запрос с **INNER JOIN**. Результат отобразить в виде таблицы. Дать пояснения.

Дано: отношения Город, Доктор (Врачи, проживающие в городах)

Город	
ID_Гор	Название
1	Москва
2	Тула
3	Рязань

Доктор	
Имя	ID_Гор
Андрей	1
Иван	2
Алексей	1
Игорь	4

OUTER JOIN (ВНЕШНЕЕ СОЕДИНЕНИЕ)

Соединение двух таблиц, в результат которого в обязательном порядке входят все строки либо одной, либо обеих таблиц.

LEFT [OUTER] JOIN

Оператор левого внешнего соединения LEFT JOIN соединяет две таблицы. Порядок таблиц для оператора важен, поскольку оператор не является симметричным. Заголовок таблицы-результата является объединением (конкатенацией) заголовков соединяемых таблиц.

В результате, если выполняется соединение левой и правой таблиц по условию р.

1. происходит внутреннее соединение (INNER JOIN) левой и правой таблиц по предикату р;
2. в результат добавляются те записи левой таблицы, которые не вошли во внутреннее соединение на шаге 1. Для таких записей поля, соответствующие правой таблице, заполняются значениями NULL.

Пример 7.6

Построить запрос с **LEFT JOIN**. Результат отобразить в виде таблицы. Дать пояснения.

RIGHT [OUTER] JOIN

АНАЛОГИЧЕН LEFT JOIN В результат добавляются те записи правой таблицы, которые не вошли во внутреннее соединение. Для таких записей поля, соответствующие левой таблице, заполняются значениями NULL.

Пример 7.7

Построить запрос с **RIGHT JOIN**. Результат отобразить в виде таблицы. Дать пояснения.

FULL [OUTER] JOIN

Оператор полного внешнего соединения FULL OUTER JOIN соединяет две таблицы. Порядок таблиц для оператора неважен, поскольку оператор является симметричным. Заголовок таблицы-результата является объединением (конкатенацией) заголовков соединяемых таблиц.

Результатирующее отношение формируется последовательно как в LEFT и RIGHT

Пример 7.8

Построить запрос с **FULL JOIN**. Результат отобразить в виде таблицы. Дать пояснения.

CROSS JOIN

Оператор перекрёстного соединения, или декартова произведения CROSS JOIN соединяет две таблицы. Порядок таблиц для оператора неважен, поскольку оператор является симметричным.

Пример 7.9

Построить два запроса с декартовым произведением. **Дать пояснения.**
(используя CROSS и без него с использованием WHERE)

Если в предложении WHERE добавить условие соединения, то есть ограничения на сочетания кортежей, то результат эквивалентен операции INNER JOIN с таким же условием:

Таким образом, CROSS JOIN + WHERE(предикат) и INNER JOIN(предикат) синтаксически являются альтернативными формами записи одной и той же логической операции внутреннего соединения. Синтаксис CROSS JOIN + WHERE для операции соединения называют устаревшим, поскольку его не рекомендует стандарт SQL ANSI.

Операция пересечения

$R \cap S = R / (R / S)$

определяет отношение, которое содержит кортежи, присутствующие как в отношении R , так и в отношении S . Отношения R и S должны быть совместимы по объединению.

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

Или

```
SELECT R.a1, R.a2
FROM R WHERE R.a1 IN
(SELECT S.b1 FROM S
WHERE S.b1=R.a1) AND R.a2 IN
(SELECT S.b2 FROM S WHERE S.b2=R.a2)
```

Операция разности

Разность R/S двух отношений R и S состоит из кортежей, которые имеются в отношении R , но отсутствуют в отношении S . Причем отношения R и S должны быть совместимы по объединению.

```
SELECT R.a1, R.a2
FROM R
WHERE NOT EXISTS
(SELECT S.b1, S.b2
FROM S
WHERE S.b1=R.a1 AND
S.b2=R.a2)
```

Операция деления

$T = R : S$ -

отношение R на

отношение S - те кортежи,

которые делятся на делителем S

комбинации всех

Операция деления одного отношения (делимого) на другое отношение (делитель) может быть выполнена, если все множество атрибутов схемы делителя является подмножеством атрибутов схемы делимого.

Результирующее отношение содержит только те атрибуты делимого, которых нет в делителе.

В него включаются только те кортежи, декартово произведение которых с делителем содержится в делимом (является подмножеством делимого).

Пусть $R = (\text{имя, пол, рост, возраст, вес})$; $S = (\text{имя, пол, возраст})$;

$T = (\text{рост, вес})$.

Отношение R				
имя	пол	рост	возраст	вес
a	ж	160	20	60
b	м	180	30	70
c	ж	150	16	40

T1=Π _c (R)	
рост	Вес
160	60
180	70

Отношение S		
имя	пол	возраст
а	ж	20

150	40
-----	----

T2=Π _c ((S X T1) - R)	
рост	вес
180	70
150	40

TT=(S X T1) - R				
имя	пол	возраст	рост	вес
а	ж	20	180	70
а	ж	20	150	40

T=T1-T2	
рост	вес
160	60

Пример 7.10. Деление отношений в SQL.

- Создание отношения R
- `CREATE TABLE R`
- `(i int primary key,`
- `имя varchar(3),`
- `пол varchar(3),`
- `рост int,`
- `возраст int,`
- `вес int)`

Создание отношения S

- `CREATE TABLE S`
- `(i int primary key,`
- `имя varchar(3),`
- `пол varchar(3),`
- `возраст int)`

Самостоятельно: Создать

отношение T1 (CREATE VIEW T1 AS SELECT рост, вес FROM R)

отношение TT (CREATE VIEW TT AS SELECT...)

отношение T2 (CREATE VIEW T2 AS SELECT...)

отношение T (Select ...) (

Пример 7.10 можно решить «вручную» на бумаге или в электронном виде)

Рассмотрим еще один пример (для полноты понимания)

Даны отношения: A

A	
S#	P#
S1	P1
S1	P2
S1	P3
S1	P4
S2	P1
S2	P3
S3	P2
S3	P3

B	
P#	
P1	

B1	
B1	P#
P2	
P3	

B2	
B2	P#
P1	
P2	
P3	

Результаты:

A:B	A:B1	A:B
		2
S#	S#	S#
S1	S1	S1
S2	S3	

Пример 7.11

Найти мультфильмы, которые показывают на R2 каналах (последовательно показать содержимое получающихся промежуточных отношений).

R1 Мультфильмы		
Код_мульт	Название_мульт	Название_канала
0	<i>The Simpsons</i>	<i>RenTV</i>
0	<i>The Simpsons</i>	<i>2x2</i>
0	<i>The Simpsons</i>	<i>CTC</i>
1	<i>Family Guy</i>	<i>RenTV</i>
1	<i>Family Guy</i>	<i>2x2</i>
2	<i>Duck Tales</i>	<i>CTC</i>
2	<i>Duck Tales</i>	<i>2x2</i>

R2 Каналы
Название_канала
<i>RenTV</i>
<i>2x2</i>

Пример 7.12

Вывести фамилии людей, которые владеют языками из R2 (аналогично 7.6)

R1

Ф.И.О.	Иностр. язык	Степ.влад. языком
<i>Иванов</i>	английский	свободно
Сидоров	английский	свободно
Сидоров	японский	разговорный
Сидоров	турецкий	со словарем
Кузнецова	английский	со словарем
Кузнецова	турецкий	свободно
Иванова	немецкий	разговорный
Иванова	турецкий	свободно
Иванова	английский	со словарем

R2

Иностр.язык	Степ.влад языком
английский	со словарем
турецкий	свободно

Пример 7.13

Отношение R содержит информацию о пиццерии и изготавливаемой ею пицце. Необходимо найти все пиццерии, где подают и cheese и supreme.

R		S
pizzeria	pizza	pizza
Chicago	cheese	cheese
Chicago	supreme	supreme
Dominos	supreme	

Самостоятельная работа №2

(Все операции реализовать в СУБД и представить результаты в электронном виде)

1. Создать две таблицы R и S и заполнить их данными

(R: 8-10кортежей, S: 4-5 кортежей.)

2. Реализовать запросы на SQL, обеспечивающие все виды операций соединения по двум отношениям (

- тета-соединение,
- соединение по эквивалентности
- Внутреннее соединение ,
- Естественное соединение,
- Левое внешнее соединение,
- Полное внешнее соединение

2. Реализовать запросы для операций

- объединения, - пересечения - разности - деления

(по аналогии с примером 7.5 - последовательно показать содержимое получающихся промежуточных отношений)