

Управление данными

Лекция 6

Операторы соединения и вложенные запросы

Свечников С.В.

Содержание занятия

- 01. Операторы соединения
- 02. Вложенные запросы
- 03. Коррелированные подзапросы

01. Операторы соединения

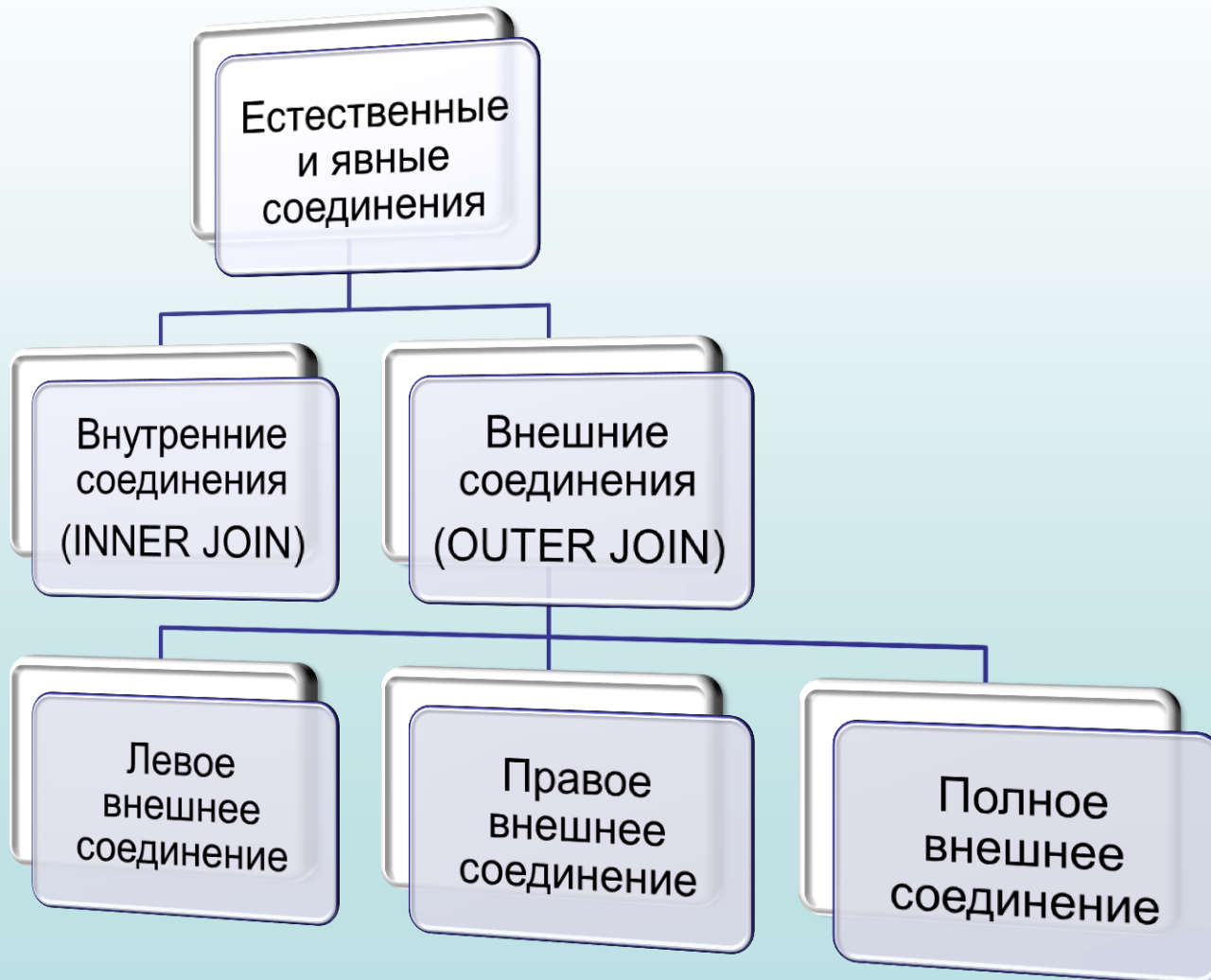
Специальные операторы соединения

В SQL определены специальные операторы, облегчающие выполнение некоторых распространенных операций соединения.



Естественные и явные соединения

Естественные и явные соединения в свою очередь могут быть разделены на следующие категории.



CROSS JOIN

Перекрестное соединение (CROSS JOIN) – это декартово произведение, в котором не выполняется никакого отбора на основе сравнения в соединяемых таблицах.

Выводятся все комбинации строк.

Пример

```
SELECT a.pid, a.lname, a.fname, b.name  
FROM person a CROSS JOIN gruppa b;
```

NATURAL JOIN

Естественное соединение (NATURAL JOIN) – эквисоединение двух таблиц, выполняемое по столбцам с одинаковыми именами.

Пример

```
SELECT a.pid, a.lname, a.fname, b.name  
FROM person a NATURAL JOIN gruppa b;
```

Необходимо, чтобы в обеих таблицах были столбцы с одинаковыми названиями.

SPECIFIED JOIN

Явные соединения (SPECIFIED JOIN) – основываются на явно указанном в запросе условии.

Имеют 2 формы:

- ❑ **ON** – использует для определения предикат
- ❑ **USING** – эквисоединение по столбцам с одинаковыми именами

Примеры

```
SELECT p.pid, p.lname, p.fname, g.name  
FROM person p JOIN gruppа g  
ON p.group_id= g.gid;
```

```
SELECT *  
FROM person JOIN gruppа  
USING (gid);
```


UNION JOIN

Объединяющие соединения (UNION JOIN) – отличаются от остальных соединений тем, что они не основаны на декартовом произведении и не используют предикаты.

Результатом объединяющего соединения будут строки одной таблицы дополненные столбцами другой таблицы с NULL-значениями.

Пример

```
SELECT p.pid, p.lname, p.fname, g.name  
FROM person p UNION JOIN группа g;
```

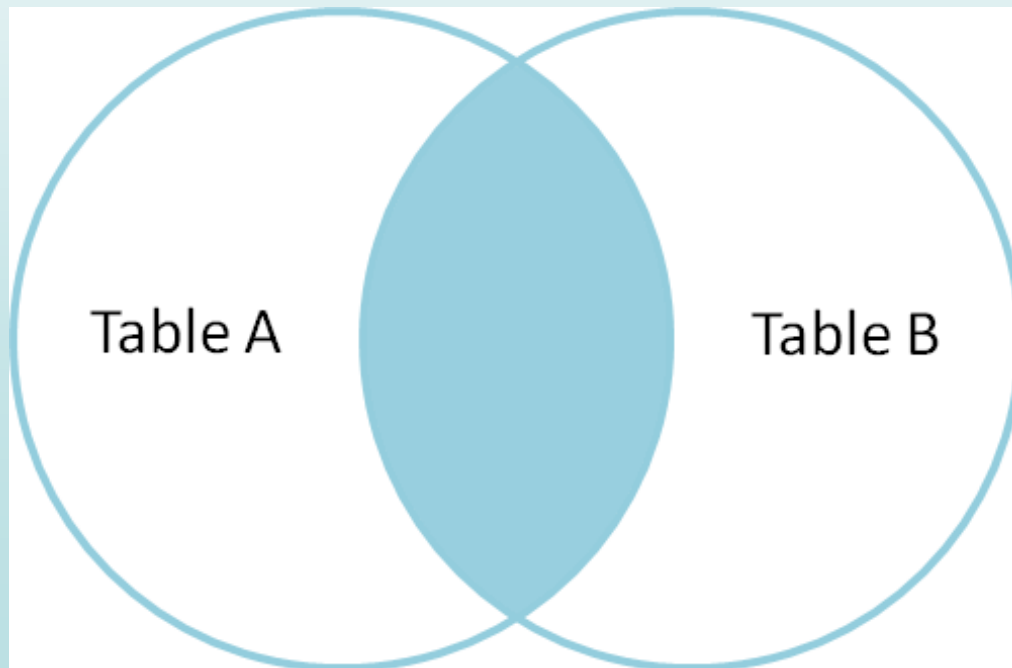
UNION JOIN в MySQL не поддерживается.

INNER JOIN

Внутреннее соединение (INNER JOIN) – соединение, исключающее несовпадающие строки. Другими словами, внутренне соединение – это любое соединение, не являющееся внешним. INNER подразумевается по умолчанию.

Пример

```
SELECT p.pid, p.lname, p.fname, g.name  
FROM person p INNER JOIN группа g  
ON p.group_id= g.gid;
```



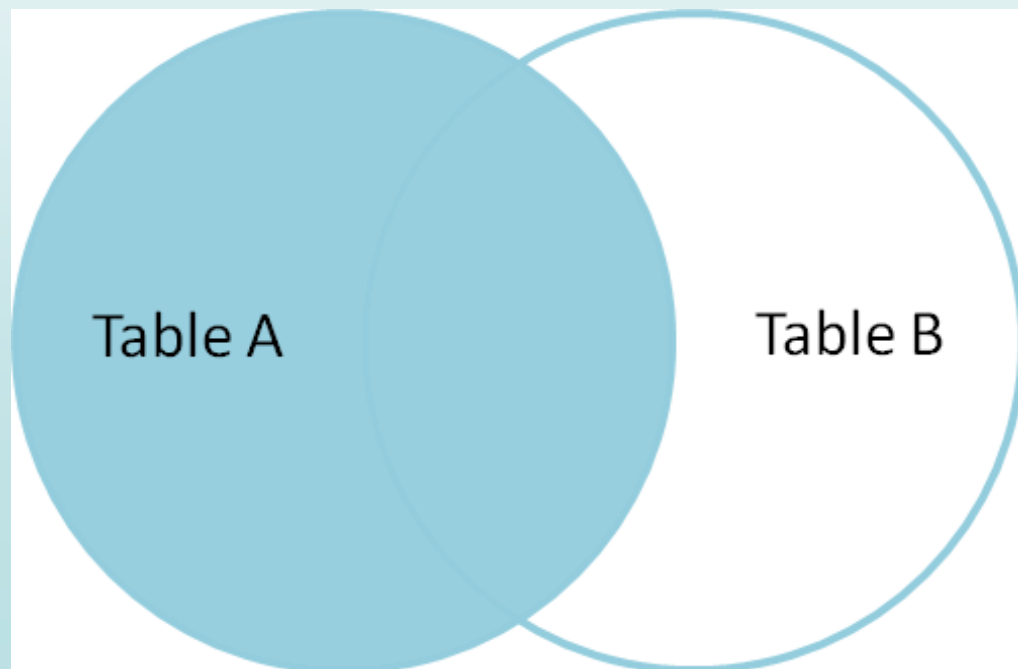
LEFT OUTER JOIN

Левое внешнее соединение (LEFT OUTER JOIN) содержит все строки из таблицы A, вместе с совпадающими значениями из таблицы B, в которых не было найдено совпадений.

Для строк таблицы A, в которых не было найдено совпадений

Пример

```
SELECT p.pid, p.lname, p.fname, g.name  
FROM person p LEFT OUTER JOIN gruppа g  
ON p.group_id= g.gid;
```

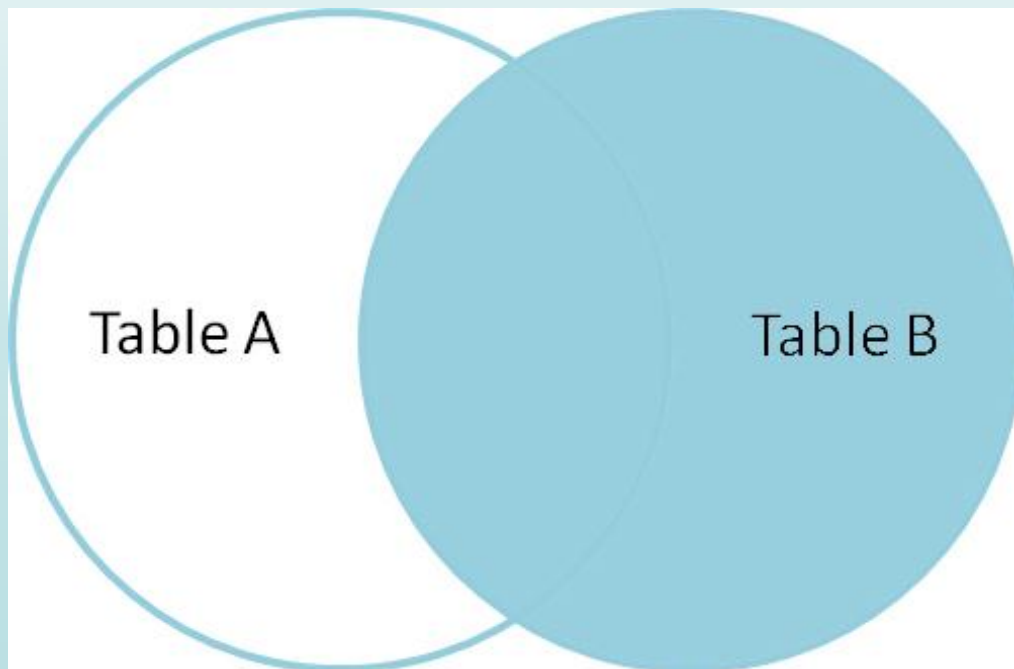


RIGHT OUTER JOIN

Правое внешнее соединение (RIGHT OUTER JOIN) является обратным по отношению к левому внешнему соединению. Все строки таблицы B будут дополнены совпадениями из таблицы A, а при отсутствии совпадений NULL-значениями.

Пример

```
SELECT p.pid, p.lname, p.fname, g.name  
FROM person p RIGHT OUTER JOIN groupa g  
ON p.group_id= g.gid;
```



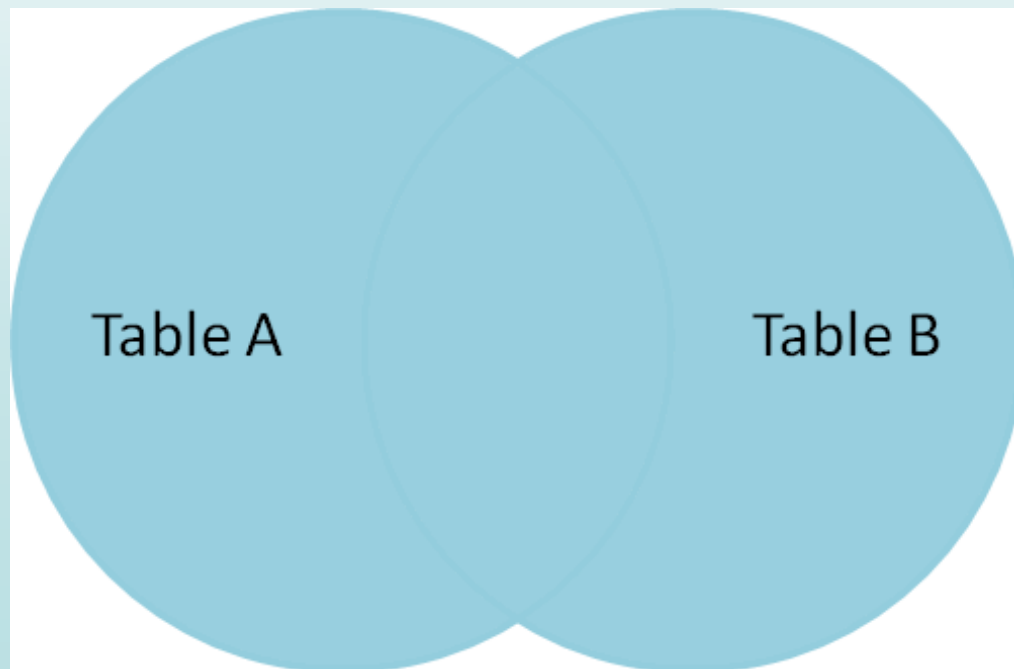
FULL OUTER JOIN

Полное внешнее соединение (FULL OUTER JOIN) является комбинацией левого и правого соединений. Оно показывает все строки обеих таблиц: при наличии совпадений – объединенные, а при их отсутствии – с NULL-значениями.

Пример

```
SELECT p.pid, p.lname, p.fname, g.name  
FROM person p FULL OUTER JOIN groupa g  
ON p.group_id= g.gid;
```

В MySQL FULL OUTER JOIN не поддерживается.



FULL OUTER JOIN в MySQL

Для того чтобы представить предыдущий запрос в MySQL можно воспользоваться объединением двух запросов: левого и правого соединений.

Пример

```
SELECT p.pid, p.lname, p.fname, g.name  
FROM person p RIGHT OUTER JOIN gruppa g  
ON p.group_id= g.gid
```

UNION

```
SELECT p.pid, p.lname, p.fname, g.name  
FROM person p LEFT OUTER JOIN gruppa g  
ON p.group_id= g.gid;
```

02. Вложенные запросы

Подзапросы

Запросы могут управлять другими запросами.

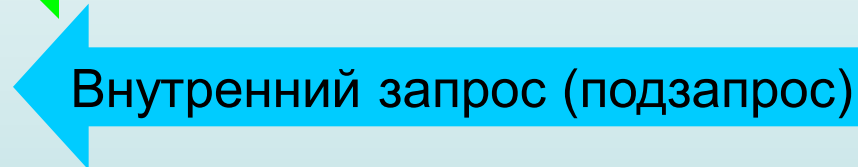
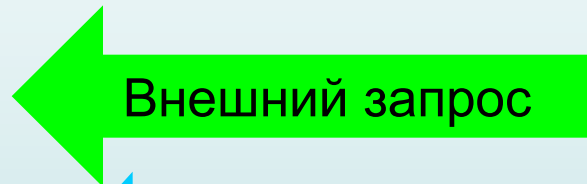
Для этого нужно поместить запрос внутри предиката другого запроса и использовать выходные данные внутреннего запроса в предикате внешнего.

SQL разрешает вкладывать запросы друг в друга.

Пример

```
SELECT * FROM person  
WHERE group_id =
```

```
(SELECT gid FROM группа  
WHERE name='ИТС-1-10');
```



Для осуществления внешнего запроса SQL сперва должен выполнить внутренний запрос.

Использование IN в подзапросах

Если подзапрос будет содержать несколько значений, то предыдущий пример не будет выполнен.

Пример

```
SELECT * FROM person  
WHERE group_id =  
(SELECT gid FROM grupp  
WHERE name like 'ИТ%');
```

Подзапрос выводит
несколько значений

```
SELECT * FROM person  
WHERE group_id IN  
(SELECT gid FROM grupp  
WHERE name like 'ИТ%');
```

Применяем оператор
IN

При использовании специального оператора IN можно формировать подзапросы, выводящие любое количество строк. В отличие от IN, операторы BETWEEN, LIKE и IS NULL не могут применяться с подзапросами.

Допустимые результаты подзапросов

Предикаты с подзапросами обычно применяют форму «скалярное выражение – оператор – подзапрос», но можно применять и другие формы.

Пример

Форма «подзапрос – оператор – скалярное выражение»:

```
SELECT * FROM person
```

```
WHERE (SELECT gid FROM gruppа WHERE name='ИТС-1-10') = group_id;
```

Использование агрегатов в подзапросах

Одним из типов функций, которые автоматически выдают единственное значение для любого количества строк являются агрегатные функции.

Пример

```
SELECT * FROM person  
WHERE year <  
(SELECT AVG(year) FROM person);
```

Если использовать GROUP BY с агрегатами, то функция выдаст несколько значений.

Количество столбцов в подзапросе

Для выполнения вложенных запросов подзапрос всегда должен выбирать единственный столбец.

Как следствие нельзя использовать следующий подзапрос, кроме случая, если таблица содержит только 1 столбец.

Пример

```
SELECT * FROM person  
WHERE group_id =  
(SELECT * FROM gruppa  
WHERE name='ИТС-1-10');
```

Не верно

Выражения в подзапросах

В предложении SELECT подзапроса можно указывать не только столбец, но и составные выражения.

Примеры

```
SELECT * FROM person  
WHERE year <  
(SELECT AVG(year)-10 FROM person);
```

```
SELECT * FROM person  
WHERE year <  
(SELECT AVG(year) FROM person)-10;
```

03. Коррелированные подзапросы

Коррелированные подзапросы

При использовании подзапросов в SQL разрешается ссылаться из внутреннего запроса на таблицу в предложении FROM внешнего запроса. Таким образом, формируется **коррелированный подзапрос**.

Пример

```
SELECT *  
FROM person vnesh  
WHERE 1 IN  
(SELECT gid FROM gruppa vnutr  
WHERE vnesh.group_id= vnutr.gid);
```

Выполнение коррелированного подзапроса

	pid	lname	fname	y...	phone	mf	city	group_id
1	1	Петров	Василий	45	9878376	муж	Москва	1
	2	Иванов	Дмитрий	17	2346899	муж	Самара	2
	3	Крылова	Ирина	18	5465556	жен	Ростов	1
	4	Жуков	Федор	34	4879830	муж	Самара	3
	5	Попов	Вадим	67	3098475	муж	Тула	1
	6	Астахов	Павел	29	<NULL>	муж	Тамбов	4
	7	Дронов	Артем	66	7321245	муж	Москва	1
	8	Козлова	Нелли	12	<NULL>	жен	Тула	2
	9	Куркина	Ольга	70	3746598	жен	Ростов	<NULL>
	10	Пончик	Игорь	22	1293845	муж	Москва	<NULL>

1	ИТС-1-10	ТИССУ	2010-09-01
2	ИТС-1-10	ТИССУ	2010-09-01
3	ИТО-1-11	ИТС	2011-09-01
4	ИТМ-1-10	ПМ	2010-09-01
5	ИСБ-1-12	ТОЭ	2012-09-03

1. Выбирается строка из таблицы во внешнем запросе.
2. Значения строки сохраняются в алиасе **vnesh**
3. Выполняется подзапрос, в котором есть совпадения **vnesh.group_id= vnutr.gid** и извлекается столбец **gid**.
4. **gid** сравнивается с предикатом внешнего запроса, т.е. значением «1».
5. Если значение **gid** совпадает, то оно выводится.
6. Процедура повторяется для следующей строки внешнего запроса.

Корреляция таблицы с собой

Коррелированные подзапросы могут быть основаны на той же таблице, что и основной запрос.

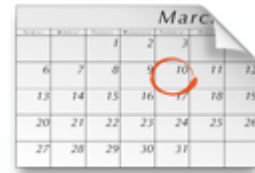
Пример

```
SELECT *  
FROM person vnesh  
WHERE year >  
(SELECT AVG(year) FROM person vnutr  
WHERE vnesh.city = vnutr.city  
and vnesh.year > vnutr.year);
```

Вопросы



Домашнее задание



- ☐ Изучить главы 10 и 11 книги М.Грабера Введение в SQL
- ☐ Подготовиться к контрольной работе по лекции
- ☐ Выполнить лабораторную работу №2

Контроль



Для выполнения контрольного теста используем ссылку:

app.startexam.com/Center/Web/student

Короткая ссылка:

clck.ru/2Undm

Пароль на тест:

5915