

Многотабличные и вложенные запросы

Три варианта запроса к множеству таблиц:

1. Соединение

- Явное (JOIN..ON)
- Неявное (список таблиц)

2. Подзапрос

- Запрос, заключенный в скобки и вложенный в предложение SELECT, FROM, WHERE или HAVING основного запроса

3. Объединение (UNION)

Неявное соединение таблиц

- `SELECT .. FROM <таблица1> [псевдоним1], <таблица2> [псевдоним2]... [WHERE <условие_соединения> [AND <условие_поиска>]...],`
- где
- `<условие_соединения> ::= <таблица1>.столбец <операция_сравнения> <таблица2>.столбец,`
- `<операция_сравнения> ::= { = | < | > | <= | >= | <> }.`

Неявное соединение таблиц

- Алгоритм выполнения запросов с неявным соединением:
 1. вычисляется декартово произведение таблиц, входящих в соединение, т.е. для каждой строки одной из таблиц берутся все возможные сочетания строк из других таблиц;
 2. производится отбор строк из полученной таблицы согласно условию поиска в предложении WHERE;
 3. осуществляется проекция (вывод) по столбцам, указанным в списке возвращаемых элементов.

Пример неявного соединения

Таблица Поставщики

Код	Наименование
1	ООО Один
2	ООО Два
3	ООО Три
4	ООО Четыре

Таблица Поставки

Код	КП	КД
1	1	20
2	3	30

SELECT * FROM Поставщики А,Поставки Б

А.Код	А.Наименование	Б.Код	Б.КП	Б.КД
1	ООО Один	1	1	20
2	ООО Два	1	1	20
3	ООО Три	1	1	20
4	ООО Четыре	1	1	20
1	ООО Один	2	3	30
2	ООО Два	2	3	30
3	ООО Три	2	3	30
4	ООО Четыре	2	3	30

Пример неявного соединения

Таблица Поставщики

Код	Наименование
1	ООО Один
2	ООО Два
3	ООО Три
4	ООО Четыре

Таблица Поставки

Код	КП	КД
1	1	20
2	3	30
3	1	10
4	4	40

Таблица Детали

Код	Наименование
10	Гвоздь 50
20	Гвоздь 100
30	Болт М3
40	Болт М5

SELECT Поставщики.Наименование AS П,
Детали.Наименование AS Д FROM
Поставщики,Поставки,Детали WHERE
Поставки.КП=Поставщики.Код AND
Поставки.КД=Детали.Код

П	Д
ООО Один	Гвоздь 100
ООО Три	Болт М3
ООО Один	Гвоздь 50
ООО Четыре	Болт М5

Явное соединение таблиц

- `SELECT .. FROM {<таблица1> [псевдоним1]
<тип_соединения1> <таблица2> [псевдоним2] [{ ON
<условие_соединения1> | USING (<список_столбцов>) }] }
[<тип_соединения2> <таблица3> [псевдоним3] [{ ON
<условие_соединения2> | USING (<список_столбцов>) }]]...`
- где
- `<тип_соединения> ::= {CROSS JOIN | [NATURAL] [{INNER | {LEFT
| RIGHT } [OUTER]}] JOIN }.`

Явное соединение таблиц

- Перекрестное соединение CROSS JOIN
.. FROM A CROSS JOIN B то же, что и .. FROM A,B
- INNER JOIN – «внутреннее» соединение. В таблицах соединяются только те строки, для которых выполняется <условие_соединения>
- OUTER JOIN – «внешнее» соединение.
- NATURAL JOIN - Естественное соединение. Стандарт SQL определяет это соединение как результат соединения таблиц по всем одноименным столбцам.

Пример явного соединения

```
SELECT Поставщики.Наименование AS П, Детали.Наименование AS Д  
FROM Поставщики,Поставки,Детали  
WHERE Поставки.КП=Поставщики.Код AND Поставки.КД=Детали.Код  
– неявное соединение
```

```
SELECT Поставщики.Наименование AS П, Детали.Наименование AS Д  
FROM Поставщики CROSS JOIN Поставки CROSS JOIN Детали  
WHERE Поставки.КП=Поставщики.Код AND Поставки.КД=Детали.Код
```

```
SELECT А.Наименование AS П, В.Наименование AS Д  
FROM Поставщики А JOIN Поставки Б ON Б.КП=А.Код  
JOIN Детали В ON Б.КД=В.Код
```


Пример явного соединения

Таблица Поставщики

Код	Наименование
1	ООО Один
2	ООО Два
3	ООО Три
4	ООО Четыре

Таблица Детали

Код	Наименование
10	Гвоздь 50
20	Гвоздь 100
30	Болт М3
40	Болт М5

```
SELECT Поставщики.Наименование AS П,  
Детали.Наименование AS Д  
FROM Поставщики JOIN Детали USING(Код)
```

эквивалентно

```
SELECT Поставщики.Наименование AS П,  
Детали.Наименование AS Д  
FROM Поставщики JOIN Детали ON  
Поставщики.Код=Детали.Код
```

Декартово произведение

Таблица Поставщики

Код	Наименование
1	ООО Один
2	ООО Два
3	ООО Три
4	ООО Четыре

Таблица Поставки

Код	КП	КД
1	1	20
2	3	30

- SELECT A.*,Б.*

FROM Поставщики А, Поставки Б

- SELECT A.*,Б.* FROM

Поставщики А CROSS JOIN Поставки Б

А.Код	А.Наименование	Б.Код	Б.КП	Б.КД
1	ООО Один	1	1	20
2	ООО Два	1	1	20
3	ООО Три	1	1	20
4	ООО Четыре	1	1	20
1	ООО Один	2	3	30
2	ООО Два	2	3	30
3	ООО Три	2	3	30
4	ООО Четыре	2	3	30

Эквисоединение

Таблица Поставщики

Код	Наименование
1	ООО Один
2	ООО Два
3	ООО Три
4	ООО Четыре

Таблица Поставки

Код	КП	КД
1	1	20
2	3	30
3	1	10
4	4	40

```
SELECT Поставщики.Наименование AS П, Поставки.Код AS К  
FROM Поставщики,Поставки,Детали  
WHERE Поставки.КП=Поставщики.Код
```

```
SELECT Поставщики.Наименование AS П, Поставки.Код AS К  
FROM Поставщики JOIN Поставки  
ON Поставки.КП=Поставщики.Код
```

Тета-соединение

Таблица Поставщики

Код	Наименование
1	ООО Один
2	ООО Два
3	ООО Три
4	ООО Четыре

Таблица Поставки

Код	КП	КД
1	1	20
2	3	30
3	1	10
4	4	40

```
SELECT Поставщики.Наименование AS П, Поставки.Код AS К  
FROM Поставщики,Поставки,Детали  
WHERE Поставки.КП>Поставщики.Код
```

А.Код	А.Наименование	Б.Код	Б.КП	Б.КД
1	ООО Один	2	3	30
2	ООО Два	2	3	30
1	ООО Один	4	4	40
2	ООО Два	4	4	40
3	ООО Три	4	4	40

Соединение со своей копией

Таблица Поставщики

Код	Наименование
1	ООО Один
2	ООО Два
3	ООО Три
4	ООО Четыре

Таблица Поставки

Код	КП	КД
1	1	20
2	3	30
3	1	10
4	4	40

```
SELECT П1.Наименование AS ПН1, П2.Наименование AS ПН2
FROM Поставщики П1, Поставщики П2
WHERE П1.Код>П2.Код
```

ПН1	ПН2
ООО Два	ООО Один
ООО Три	ООО Один
ООО Три	ООО Два
ООО Четыре	ООО Один
ООО Четыре	ООО Два
ООО Четыре	ООО Три

Соединение со своей копией

Таблица students

cod_st	name_st	name2
1	Иванов	Иван
2	Петров	Петр
3	Сидоров	Петр
4	Иванов	Дмитрий

```
SELECT DISTINCT s1.name_st  
FROM students s1,students s2  
WHERE s1.name_st=s2.name_st AND  
s1.cod_st<>s2.cod_st
```

или

```
SELECT name_st FROM students  
GROUP BY name_st  
HAVING COUNT(name_st)>1
```

Вложенные запросы

- Marks(cod_m,cod_st,mark)
- Students(cod_st,name_st,born)
- Subjects(cod_sub,name_sub)
- Во фразе FROM: SELECT .. FROM (SELECT..) ..
SELECT MIN(avg_mark),MAX(avg_mark) FROM
(SELECT AVG(mark) avg_mark FROM marks GROUP BY cod_st)
- В условиях отбора
 - SELECT mark FROM marks WHERE cod_st IN
(SELECT cod_st from students WHERE name_st='Иванов')
AND cod_sub IN (SELECT cod_sub FROM subjects WHERE
name_sub='Математика')

Вложенные запросы

Вывести Фамилии самых молодых студентов

- `SELECT name_st FROM students WHERE
born=(SELECT MAX(born) FROM students)`

Cod_st	Name_st	Born
1	Иванов	11.01.1995
2	Петров	12.06.1996
3	Сидоров	23.09.1994
4	Кузнецов	12.06.1996



Name_st
Петров
Кузнецов

Вложенные запросы

Найти студентов с максимальным средним баллом

- `SELECT name_st FROM students WHERE cod_st IN (SELECT cod_st FROM marks GROUP BY cod_st HAVING AVG(mark)=(SELECT MAX(avg_mark) max_avg_mark FROM (SELECT AVG(mark) avg_mark FROM marks GROUP BY code_st)))`

Cod_st	Name_st	Born
1	Иванов	11.01.1995
2	Петров	12.06.1996
3	Сидоров	23.09.1994
4	Кузнецов	12.06.1996

Cod_m	Cod_st	Cod_sub	Mark
01	1	1	4
02	1	2	3
03	2	1	3
04	2	2	5
05	3	1	4
06	3	2	5



cod_st
3

Вложенные запросы в качестве вычисляемых столбцов

- Вычислить среднюю оценку по каждому студенту
- ```
SELECT cod_st,name_st,
(SELECT AVG(mark) FROM marks WHERE
cod_st=students.cod_st) avg_mark FROM students
```

| Cod_st | Name_st  | Born       |
|--------|----------|------------|
| 1      | Иванов   | 11.01.1995 |
| 2      | Петров   | 12.06.1996 |
| 3      | Сидоров  | 23.09.1994 |
| 4      | Кузнецов | 12.06.1996 |

| Cod_m | Cod_st | Cod_sub | Mark |
|-------|--------|---------|------|
| 01    | 1      | 1       | 4    |
| 02    | 1      | 2       | 3    |
| 03    | 2      | 1       | 3    |
| 04    | 2      | 2       | 5    |
| 05    | 3      | 1       | 4    |
| 06    | 3      | 2       | 5    |



| Cod_st | Name_st | Avg_mark |
|--------|---------|----------|
| 1      | Иванов  | 3,5      |
| 2      | Петров  | 4        |
| 3      | Сидоров | 4,5      |

# Оператор EXISTS

- SELECT cnum, cname, city FROM Customers WHERE EXISTS (SELECT \* FROM Customers WHERE city = 'Москва')

| cnum | cname                            | city      |
|------|----------------------------------|-----------|
| 2001 | ТОО Рога и копыта                | Москва    |
| 2002 | АО Бендер и К                    | Одесса    |
| 2003 | фирма ХХХ                        | Рязань    |
| 2004 | концерн "дети лейтенанта Шмидта" | Бобруйск  |
| 2006 | С1emens                          | Лондон    |
| 2007 | ОАО "ООО"                        | ТОМСК     |
| 2008 | ОАО "Валют-транзит"              | Караганда |

# Оператор EXISTS

- `SELECT DISTINCT snum FROM Customers couter WHERE EXISTS  
(SELECT * FROM Customers cinner WHERE cinner.snum =  
couter.snum AND cinner.cnum <> couter.cnum)`

# ИСПОЛЬЗОВАНИЕ ОПЕРАТОРОВ ANY, ALL

- <выражение> <оператор сравнения> SOME | ANY (<подзапрос>)
- <выражение> <оператор сравнения> ALL (<подзапрос>)
- SELECT DISTINCT maker FROM Product WHERE type = 'pc' AND NOT model = ANY (SELECT model FROM PC)
- SELECT DISTINCT model, price FROM Laptop WHERE price > ALL (SELECT price FROM PC)

# Формальные правила оценки истинности предикатов, использующих параметры ANY|SOME и ALL.

- Если определен параметр ALL и все результаты сравнения значения выражения и каждого значения, полученного из подзапроса, являются TRUE, истинностное значение равно TRUE.
- Если результат выполнения подзапроса не содержит строк и определен параметр ALL, результат равен TRUE. Если же определен параметр SOME, результат равен FALSE.
- Если определен параметр ALL и результат сравнения значения выражения хотя бы с одним значением, полученным из подзапроса, является FALSE, истинностное значение равно FALSE.
- Если определен параметр SOME и хотя бы один результат сравнения значения выражения и значения, полученного из подзапроса, является TRUE, истинностное значение равно TRUE.
- Если определен параметр SOME и каждое сравнение значения выражения и значений, полученных из подзапроса, равно FALSE, истинностное значение тоже равно FALSE.
- В любом другом случае результат будет равен **UNKNOWN**.