

Оператор SELECT (напоминание)

SELECT

Определение заголовка целевой таблицы запроса

Обязательные

FROM

Определение источника данных запроса

WHERE

Определение условий отбора строк целевой таблицы

GROUP BY

Определение признаков группирования строк целевой таблицы

HAVING

Определение условий отбора групп строк целевой таблицы

ORDER BY

Определение условий упорядочения строк целевой таблицы

Строгое описание

<https://intuit.ru/studies/courses/75/75/info>

<http://citforum.ru/database/sqlbook/index.shtml>

Общий вид оператора SELECT

SELECT [**DISTINCT** | **ALL**] <целевой список столбцов>
FROM <табличное выражение> ; **МИНИМАЛЬНЫЙ**
[**WHERE** <предикат>]
[**GROUP BY** <список столбцов группирования строк>]
[**HAVING** <предикат>]
[**ORDER BY** <список столбцов упорядочения строк>];

СИМВОЛ ; ЕСТЬ ПРИЗНАК ЗАВЕРШЕНИЯ ОПЕРАТОРА.
ОБЯЗАТЕЛЕН!!!

Порядок обработки разделов SELECT

FROM <табличное выражение>
[**WHERE** <предикат>]
[**GROUP BY** <список столбцов группирования строк>]
[**HAVING** <предикат>]
SELECT [**DISTINCT** | **ALL**] <целевой список столбцов>
[**ORDER BY** <список столбцов упорядочения строк>];

8.8. Предикаты SQL

предикат ::= [() [NOT] элементарный_предикат
[{AND | OR} предикат] ()] [IS [NOT] 3VL-значение]

элементарный_предикат ::= предикат_сравнение
| предикат_BETWEEN
| предикат_IN
| предикат_LIKE
| предикат_IS NULL
| предикат_EXISTS
| предикат_QUANTIFIED COMPARISON
| предикат_UNIQUE
| предикат_MATCH
| предикат_OVERLAP
| предикат_SIMILAR
| предикат_IS DISTINCT

В SQL используется трёхзначная логика (3VL)
3VL-значение ::= TRUE | FALSE | UNKNOWN

Таблицы истинности 3VL

X	NOT X
TRUE	FALSE
FALSE	TRUE
UNKNOWN	UNKNOWN

AND	TRUE	FALSE	UNKNOWN
TRUE	TRUE	FALSE	UNKNOWN
FALSE	FALSE	FALSE	FALSE
UNKNOWN	UNKNOWN	FALSE	UNKNOWN

IS	TRUE	FALSE	UNKNOWN
TRUE	TRUE	FALSE	FALSE
FALSE	FALSE	TRUE	FALSE
UNKNOWN	FALSE	FALSE	TRUE

OR	TRUE	FALSE	UNKNOWN
TRUE	TRUE	TRUE	TRUE
FALSE	TRUE	FALSE	UNKNOWN
UNKNOWN	TRUE	UNKNOWN	UNKNOWN

Предикат сравнения

сравнение ::= выражение оператор_сравнения выражение
оператор_сравнения ::= < | = | > | <= | >= | <>

'Томск' <> 'Яя' \Rightarrow TRUE

'Томск' = 'Яя' \Rightarrow FALSE

Никакие два NULL-значения не совпадают, НО NULL <> NULL \Rightarrow UNKNOWN

выражение – любое допустимое выражение для вычисления значения

Получить номера изделий, для которых выполняет поставки поставщик 'Иван'.

```
SELECT DISTINCT SPJ.Jnum  
FROM S, SPJ  
WHERE S.Snum = SPJ.Snum and S.name = 'Иван';
```

Предикат BETWEEN ::=

выражение [NOT] BETWEEN выражение AND выражение

Предикат

равносилен предикату

expr1 BETWEEN expr2 AND expr3

expr1 >= expr2 AND expr1 <= expr3

Примеры

ПРЕДИКАТ	ЗНАЧЕНИЕ
5 BETWEEN 3 AND 8	TRUE
5 BETWEEN 6 AND NULL	FALSE
5 BETWEEN NULL AND 8	UNKNOWN

Получить номера деталей, объёмы разовых поставок которых лежат в диапазоне [1200, 2000] штук.

SELECT DISTINCT Pnum

FROM SPJ

WHERE Qt BETWEEN 1200 AND 2000;

Предикат IN. Проверка принадлежности значения заданному множеству.

**Предикат_IN ::= выражение [NOT] IN выражение
AND (выражение. , . .) | подзапрос**

Примеры

ПРЕДИКАТ	ЗНАЧЕНИЕ
5 IN (3, 5, 2)	TRUE
5 IN (3, 7, 2)	FALSE
5 IN (3, NULL, 2)	UNKNOWN
5 IN (NULL, 5, NULL)	TRUE

Получить номера и имена поставщиков, выполняющих поставки хотя бы одной из деталей 'P1', 'P5', 'P12'.

```
SELECT DISTINCT S.Snum, S.Snam  
FROM SPJ, S  
WHERE SPJ.Snum = S.Snum AND SPJ.Pnum IN ('P1', 'P5', 'P12');
```


Предикат LIKE. Работает только со строками символов.

Проверяет, входит ли заданная (искомая) строка в состав другой строки (образца).

Предикат LIKE ::= выражение [NOT] LIKE выражение
[ESCAPE символ]

проверяемое

образец

управляющий

Примеры

ПРЕДИКАТ	ЗНАЧЕНИЕ
'Томск' LIKE 'Томск '	TRUE
'Томск' LIKE 'Томск'	TRUE
'Томск ' LIKE 'Томск'	FALSE

Образец может содержать *знаки шаблона (трафаретные символы)*:
символ '_' заменяет любой **один** символ в проверяемом значении;
символ '%' заменяет **подстроку** символов любой (в том числе и нулевой) длины.

Примеры

ПРЕДИКАТ	ЗНАЧЕНИЕ
'Томск' LIKE '_омск '	TRUE
'Омск' LIKE '%омск'	TRUE
'Атомск' LIKE '%омск'	TRUE
'Омск' LIKE '_омск'	FALSE

ESCAPE символ – определение символа, отменяющего для ‘%’ и ‘_’ смысл знаков шаблона.

Пример

Предикат X LIKE ‘__|% %’ **ESCAPE** ‘|’

возвратит **TRUE** на любой последовательности символов X, которая начинается двумя любыми символами, имеет в третьей позиции символ ‘%’, в четвёртой – пробел, далее – всё, что угодно, в том числе и ничего.

Символ ‘%’ в первом вхождении считается литералом (перед ним помещён ‘|’), во втором – знаком шаблона.

Если *проверяемое* или *образец*, или *управляющий* символ есть **NULL**,
то значение предиката **UNKNOWN**.

Если *проверяемое* и *образец* эквивалентны,
то значение предиката **TRUE**.

Если в *проверяемом* значении можно выделить
сегменты, эквивалентные явно заданным подстрокам *образца*, и
сегменты, соответствующие знакам шаблона, и
эти сегменты расположены в *проверяемом* значении и
в *образце* в одинаковой последовательности,
то значение предиката **TRUE**.

Если не выполнено ни одно из перечисленных условий,
то значение предиката **FALSE**.

Пример

X LIKE ' __|%%' ESCAPE '|'

«истинными» будут признаны строки: '55% белков'

'30% жиров'

'05% углеводов'

«ложными» – строки: 'содержит 55% белков';

'30%жиров';

'5% углеводов'

Предикат IS NULL. Обнаружение NULL-значений.

предикат_IS NULL ::= выражение IS [NOT] NULL

Возвращает **TRUE**, если значение выражения не определено

FALSE – в противном случае.

Значение **UNKNOWN** невозможно.

Предикат EXISTS. Проверка наличия строк в таблице.

предикат_exists ::= EXISTS подзапрос

TRUE - если результат подзапроса содержит хотя бы одну строку;

FALSE - в противном случае.

Реализует *квантор существования* РИ с переменными-кортежами.

Предикат QUANTIFIED COMPARISON (специальный тип предиката сравнения)

предикат_quantified_comparison ::=

выражение оператор_сравнения {ALL|ANY|SOME} подзапрос

Ключевые слова (*кванторы*) **ANY** и **SOME** – синонимы.

Значение **выражения** сравнивается со значениями результата **подзапроса**.

Результат **всех** сравнений - **одно** истинностное значение, зависящее от *квантора*.

Если результат **подзапроса** - *непустая* таблица, то

exp θ ALL подзапрос

Равносильно **exp θ val1 AND exp θ val2 AND ... AND exp θ valN.**

exp θ ANY (подзапрос) ;

Равносильно **exp θ val1 OR exp θ val2 OR ... OR exp θ valN.**

Примеры

ПРЕДИКАТ	ЗНАЧЕНИЕ
5 >= ALL (3, 5, 2)	TRUE
5 > ALL (3, 5, 2)	FALSE
5 >= ALL (3, NULL, 2)	UNKNOWN
5 > ANY (3, 5, 2)	TRUE
5 < ANY (3, 5, 2)	FALSE
5 >= ANY (3, NULL, 2)	TRUE
5 >= ANY (6, NULL, 9)	UNKNOWN

Предложение WHERE

Пример 1.

Получить всю информацию о поставщиках, имеющих статус выше 90.

```
SELECT *  
FROM S  
WHERE St > 90;
```

Базовая таблица

Snum	Snam	St	Ci
S8	Владимир	30	Томск
S2	Николай	50	Асино
S5	Константин	100	Яя
S4	Петр	20	Рио-де-Жанейро
S3	Григорий	80	Яя
S9	Егор	100	Яя
S7	Сергей	90	Асино
S1	Иван	100	Томск
S6	Иван	100	Лесото

Результат выборки

Snum	Snam	St	Ci
S5	Константин	100	Яя
S9	Егор	100	Яя
S1	Иван	100	Томск
S6	Иван	100	Лесото

Пример 2.

Получить значения номеров и наименований поставщиков в Томске.

```
SELECT Snum, Snam  
FROM S  
WHERE Ci = Томск;
```

Базовая таблица

Snum	Snam	St	Ci
S8	Владимир	30	Томск
S2	Николай	50	Асино
S5	Константин	100	Яя
S4	Петр	20	Рио-де-Жанейро
S3	Григорий	80	Яя
S9	Егор	100	Яя
S7	Сергей	90	Асино
S1	Иван	100	Томск
S6	Иван	100	Лесото

Результат выборки

Snum	Snam
S8	Владимир
S1	Иван

Соединения таблиц

соединение ::= перекрёстное_соединение
 | естественное_соединение
 | левое_внешнее_соединение
 | правое_внешнее_соединение
 | полное_внешнее_соединение
 | соединение_по_именам_столбцов
 | соединение_по_условию
 | UNION_соединение

ЛЮБОЕ ИЗ ЭТИХ СОЕДИНЕНИЙ ЗАМЕЩАЕТ ИМЯ ТАБЛИЦЫ В ПРЕДЛОЖЕНИИ **FROM**

перекрёстное_соединение ::= таблица_A CROSS JOIN таблица_B

Таблица P (Детали)

Pnum	Pnam	We	Co	Ci
P1	корпус	300	белый	Томск
P3	кнопка	5	красный	Томск
P5	панель	100	серый	Асино

Таблица J (Изделия)

Jnum	Jnam	Ci
J1	процессор	Асино
J6	клавиатура	Томск

P CROSS JOIN J

Pnum	Pnam	We	Co	P.Ci	Jnum	Jnam	J.Ci
P1	корпус	300	белый	Томск	J1	процессор	Асино
P1	корпус	300	белый	Томск	J6	клавиатура	Томск
P3	кнопка	5	красный	Томск	J1	процессор	Асино
P3	кнопка	5	красный	Томск	J6	клавиатура	Томск
P5	панель	100	серый	Асино	J1	процессор	Асино
P5	панель	100	серый	Асино	J6	клавиатура	Томск

FROM A CROSS JOIN B ЭКВИВАЛЕНТНО FROM A, B

Операция естественного соединения

в SQL может быть реализована в четырёх разновидностях.

естественное_соединение ::= таблица_A [NATURAL] [тип_соединения]
JOIN таблица_B

тип_соединения ::= INNER | {{LEFT | RIGHT | FULL} [OUTER]}

Таблица P (Детали)

Pnum	Pnam	We	Co	Ci
P1	корпус	300	белый	Томск
P3	кнопка	5	красный	Томск
P5	панель	100	серый	Асино
P2	разъём	3	чёрный	Яя

Таблица J (Изделия)

Jnum	Jnam	Ci
J1	процессор	Асино
J6	клавиатура	Томск
J7	мышь	NULL

P INNER JOIN J (Внутреннее естественное соединение)

Pnum	Pnam	We	Co	Ci	Jnum	Jnam
P5	панель	100	серый	Асино	J1	Процессор
P3	кнопка	5	красный	Томск	J6	Клавиатура
P1	корпус	300	белый	Томск	J6	Клавиатура