

Профессиональная программа “Анализ данных на языке SQL”

Блок 2. Простые операции с одной таблицей

Содержание блока

- Инструкция SELECT
- Фильтрация таблицы по столбцам и строкам
- Использование функций
- Типы данных
- Обработка неопределенных значений

Содержание блока

- Сортировка
- Операция TOP
- Устранение дубликатов
- Порядок операций в запросе

В конце блока предусмотрено решение ряда практических задач.

1-ый учебный вопрос

Инструкция SELECT

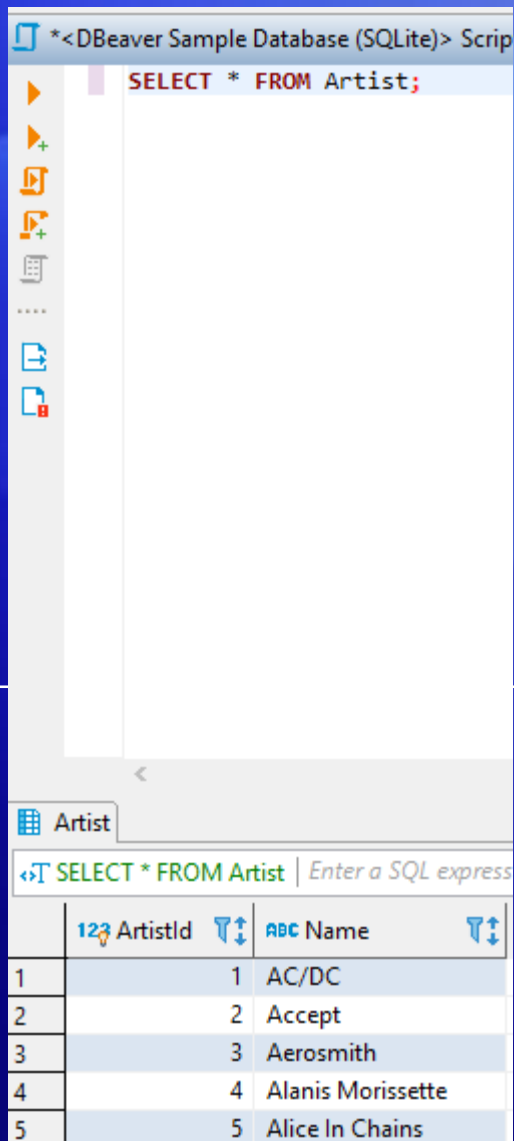
Общие сведения: инструкция SELECT



Возвращает строки из базы данных и позволяет делать выборку одной или нескольких строк или столбцов из одной или нескольких таблиц в SQL Server. Полный синтаксис инструкции SELECT сложен, однако основные предложения можно вкратце описать следующим образом:

```
[ WITH { [ XMLNAMESPACES ,] [ <common_table_expression> ] } ]  
SELECT select_list [ INTO new_table ]  
[ FROM table_source ] [ WHERE search_condition ]  
[ GROUP BY group_by_expression ]  
[ HAVING search_condition ]  
[ ORDER BY order_expression [ ASC | DESC ] ]
```

Простейшая инструкция SELECT



The screenshot shows the DBeaver interface with a SQL editor at the top containing the query `SELECT * FROM Artist;`. Below the editor, the 'Artist' table is selected, and its data is displayed in a table view. The table has two columns: 'ArtistId' and 'Name'. The results show five rows of data.

	ArtistId	Name
1	1	AC/DC
2	2	Accept
3	3	Aerosmith
4	4	Alanis Morissette
5	5	Alice In Chains

На сервере физически существует таблица Artist. Известно, что она заполнена данными о рок группах, альбомы которых есть в коллекции. При соблюдении этих условий, для просмотра таблицы Artist полностью, со всеми элементами и экземплярами, что в ней есть, используется команда `SELECT * FROM [имя_таблицы]`

Символ * после команды SELECT означает, что результатом выборки станет вся таблица, к которой идет обращение.

2-ый учебный вопрос

Фильтрация таблицы по
столбцам и строкам

Фильтрация таблицы по столбцу

12	ABC FirstName	ABC LastName	ABC Company	ABC Address	ABC City
1	Luís	Gonçalves	Embraer - Empresa Brasile	Av. Brigadeiro Faria Lima, 2170	São José dos Campos
2	Leonie	Köhler	[NULL]	Theodor-Heuss-Straße 34	Stuttgart
3	François	Tremblay	[NULL]	1498 rue Bélanger	Montréal
4	Bjørn	Hansen	[NULL]	Ullevålsveien 14	Oslo
5	František	Wichterlová	JetBrains s.r.o.	Klanova 9/506	Prague
6	Helena	Holý	[NULL]	Rilská 3174/6	Prague
7	Astrid	Gruber	[NULL]	Rotenturmstraße 4, 1010 Innere Stadt	Vienne
8	Daan	Peeters	[NULL]	Grétrystraat 63	Brussels
9	Kara	Nielsen	[NULL]	Sønder Boulevard 51	Copenhagen
10	Eduardo	Martins	Woodstock Discos	Rua Dr. Falcão Filho, 155	São Paulo
11	Alexandre	Rocha	Banco do Brasil S.A.	Av. Paulista, 2022	São Paulo

Приведенная на слайде таблица имеет несколько столбцов и строк со значениями. В данном случае это информация о покупателях музыкальных альбомов (имя, фамилия, место работы, адрес, город..). В приведенном примере таблица снова выведена через конструкцию `SELECT * FROM`.

Фильтрация таблицы по столбцу

Customer	
SELECT LastName FROM Customer	
	ABC LastName
1	Gonçalves
2	Köhler
3	Tremblay
4	Hansen
5	Wichterlová
6	Holý
7	Gruber
8	Peeters
9	Nielsen
10	Martins
11	Rocha

Для фильтрации данных по столбцу в конструкции SELECT символ * меняется на имя столбца таблицы, который нужно оставить в результате выполнения запроса.

Результатом выполнения инструкции SELECT LastName FROM Customer является один столбец LastName из таблицы Customer.

Фильтрация таблицы по столбцам

Customer		
SELECT FirstName, LastName FROM Customer		
	ABC FirstName	ABC LastName
1	Luís	Gonçalves
2	Leonie	Köhler
3	François	Tremblay
4	Bjørn	Hansen
5	František	Wichterlová
6	Helena	Holý
7	Astrid	Gruber
8	Daan	Peeters
9	Kara	Nielsen
10	Eduardo	Martins
11	Alexandre	Rocha
12	Roberto	Almeida

Для фильтрации данных по нескольким столбцам в конструкции SELECT символ * меняется на имена столбцов таблицы, которые **нужно оставить** в результате выполнения запроса, перечисленные без каких-либо символов за исключением запятой между ними.

Результатом выполнения инструкции SELECT FirstName, LastName FROM Customer является два столбца: Имя и Фамилия из таблицы Customer.

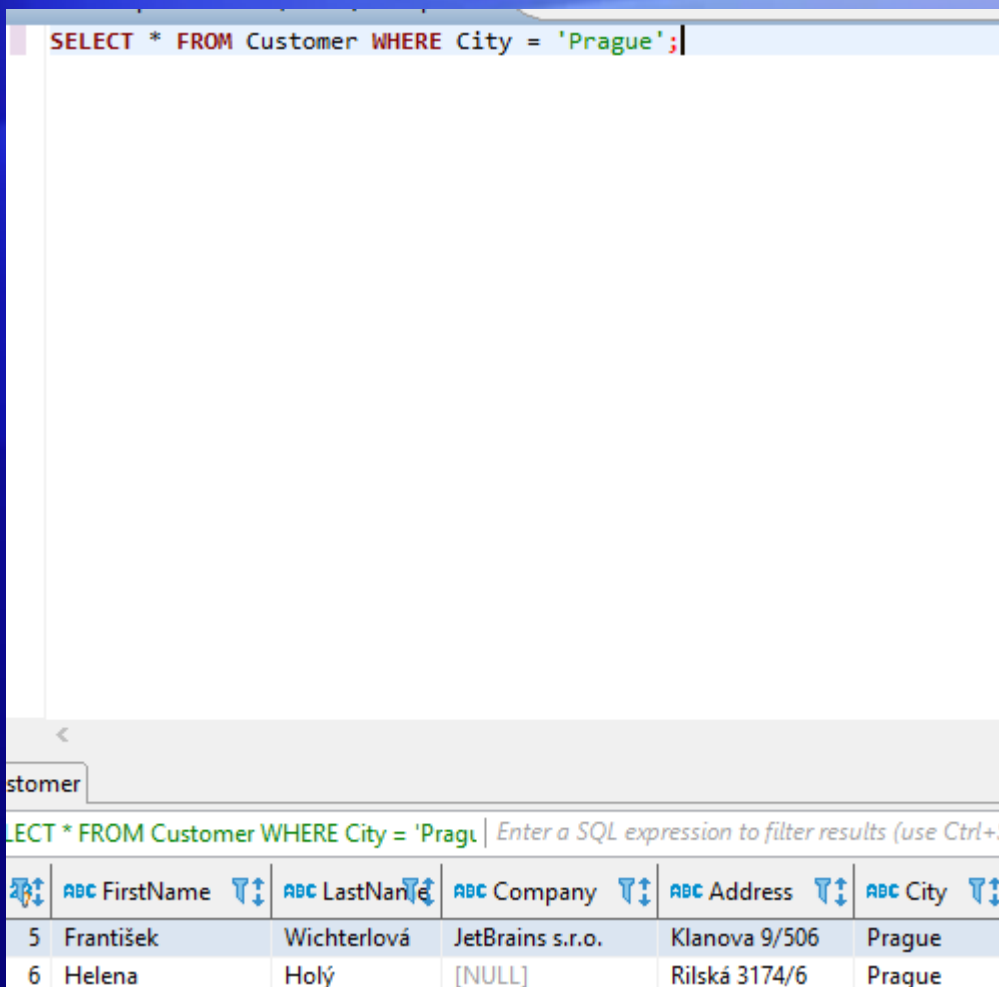
Порядок фильтрации таблицы по столбцам

Customer		
SELECT LastName, FirstName FROM Customer		
	ABC LastName	ABC FirstName
1	Gonçalves	Luís
2	Köhler	Leonie
3	Tremblay	François
4	Hansen	Bjørn
5	Wichterlová	František
6	Holý	Helena
7	Gruber	Astrid
8	Peeters	Daan
9	Nielsen	Kara
10	Martins	Eduardo
11	Rocha	Alexandre
12	Almeida	Roberto

Следует обратить внимание на то, что порядок перечисления столбцов в инструкции SELECT сохраняется и при выводе результата на экран.

Результатом выполнения инструкции SELECT LastName, FirstName FROM Customer является два столбца: Фамилия и Имя из таблицы Customer. В результате столбцы поменялись местами, несмотря на свое физическое расположение в таблице.

Фильтрация таблицы по строкам



The screenshot shows a SQL query editor with the following SQL statement entered:

```
SELECT * FROM Customer WHERE City = 'Prague';
```

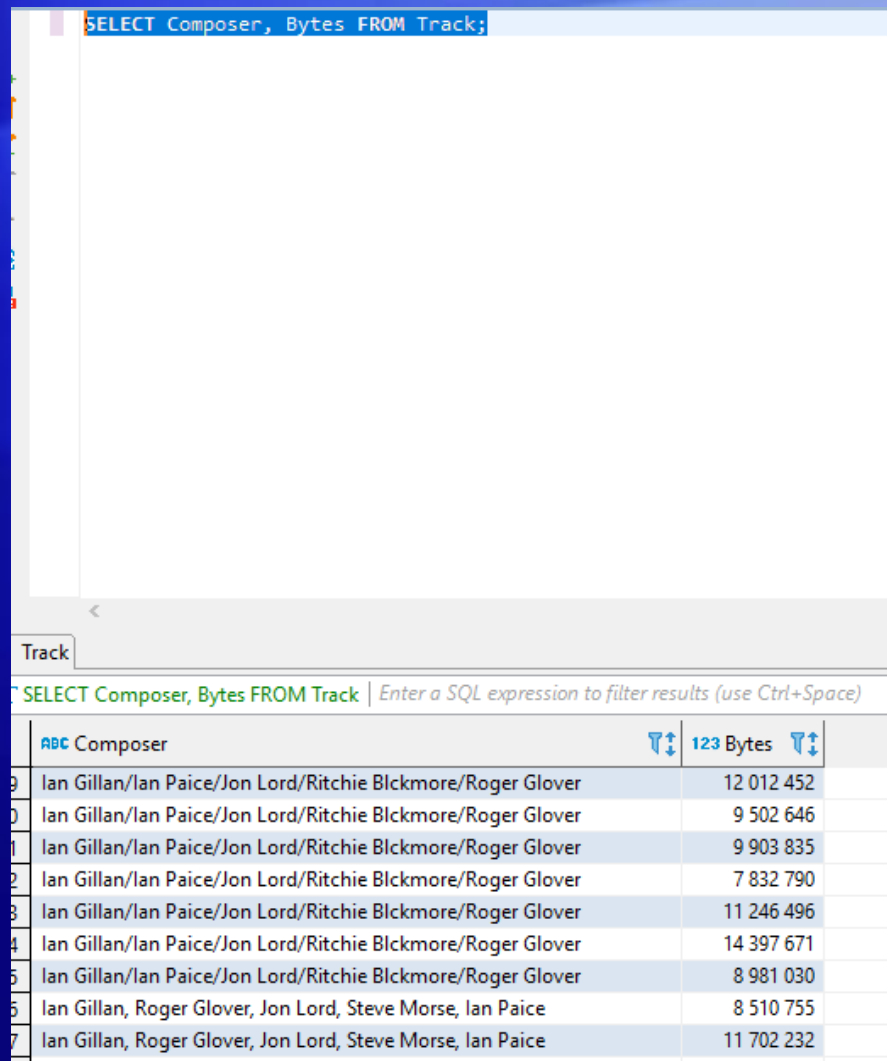
Below the query editor, the results of the query are displayed in a table. The table has six columns: ID, FirstName, LastName, Company, Address, and City. The results show two rows of data.

	ABC FirstName	ABC LastName	ABC Company	ABC Address	ABC City
5	František	Wichterlová	JetBrains s.r.o.	Klanova 9/506	Prague
6	Helena	Holý	[NULL]	Rilská 3174/6	Prague

Для фильтрации данных по строке в конструкцию SELECT, после указания таблицы выборки добавляется команда WHERE с условием фильтрации по строкам. Условие “=” выведет только те строки, которые точно соответствуют правой части условия.

Результатом выполнения инструкции SELECT * FROM Customer WHERE City = 'Prague' станет вся информация о покупателях, которые проживают в городе Прага.

Фильтрация таблицы по строкам



The screenshot shows a database query editor with a SQL query in the top pane and its results in the bottom pane. The query is `SELECT Composer, Bytes FROM Track;`. The results pane shows a table with two columns: **Composer** and **Bytes**. The **Composer** column contains names of composers, and the **Bytes** column contains file sizes. The results are filtered to show only rows where the **Composer** column contains the name "Ilan Gillan".

	Composer	Bytes
9	Ilan Gillan/Ilan Paice/Jon Lord/Ritchie Blckmore/Roger Glover	12 012 452
0	Ilan Gillan/Ilan Paice/Jon Lord/Ritchie Blckmore/Roger Glover	9 502 646
1	Ilan Gillan/Ilan Paice/Jon Lord/Ritchie Blckmore/Roger Glover	9 903 835
2	Ilan Gillan/Ilan Paice/Jon Lord/Ritchie Blckmore/Roger Glover	7 832 790
3	Ilan Gillan/Ilan Paice/Jon Lord/Ritchie Blckmore/Roger Glover	11 246 496
4	Ilan Gillan/Ilan Paice/Jon Lord/Ritchie Blckmore/Roger Glover	14 397 671
5	Ilan Gillan/Ilan Paice/Jon Lord/Ritchie Blckmore/Roger Glover	8 981 030
6	Ilan Gillan, Roger Glover, Jon Lord, Steve Morse, Ian Paice	8 510 755
7	Ilan Gillan, Roger Glover, Jon Lord, Steve Morse, Ian Paice	11 702 232

С помощью фильтрации по столбцам была создана выборка из таблицы “Композиции” (Track), которая содержит только информацию о композиторе и размер файла с музыкой (Composer, Bytes).

Следующая группа фильтров продемонстрирует работу с числовыми данными на полученном массиве.

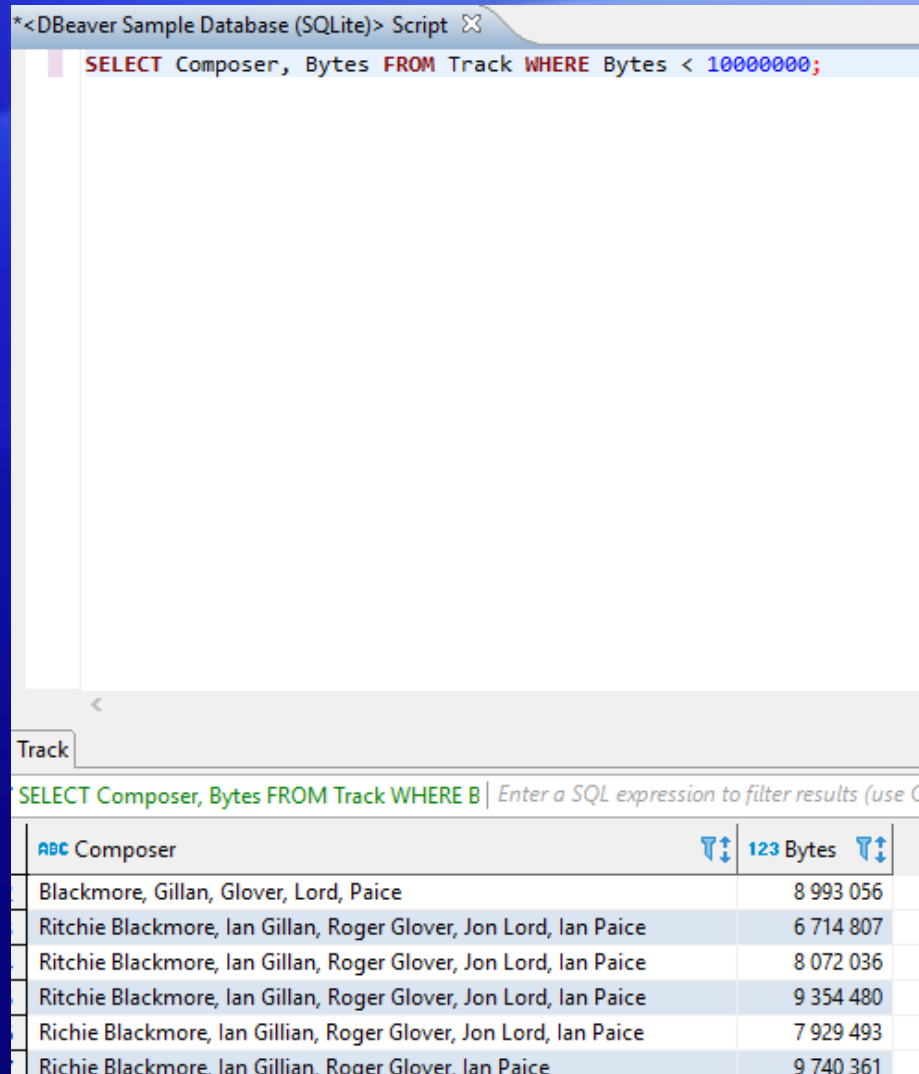
Общие сведения: условные операторы фильтрации строк по числовым значениям



SQL поддерживает следующие условные операторы для числовых значений:

Знак операции	Значение
=	Равно
<>	Не равно
<	Меньше
<=	Меньше или равно
>	Больше
>=	Больше или равно
BETWEEN	Между двумя значениями
IS NULL	Отсутствует запись

Фильтрация таблицы по строкам



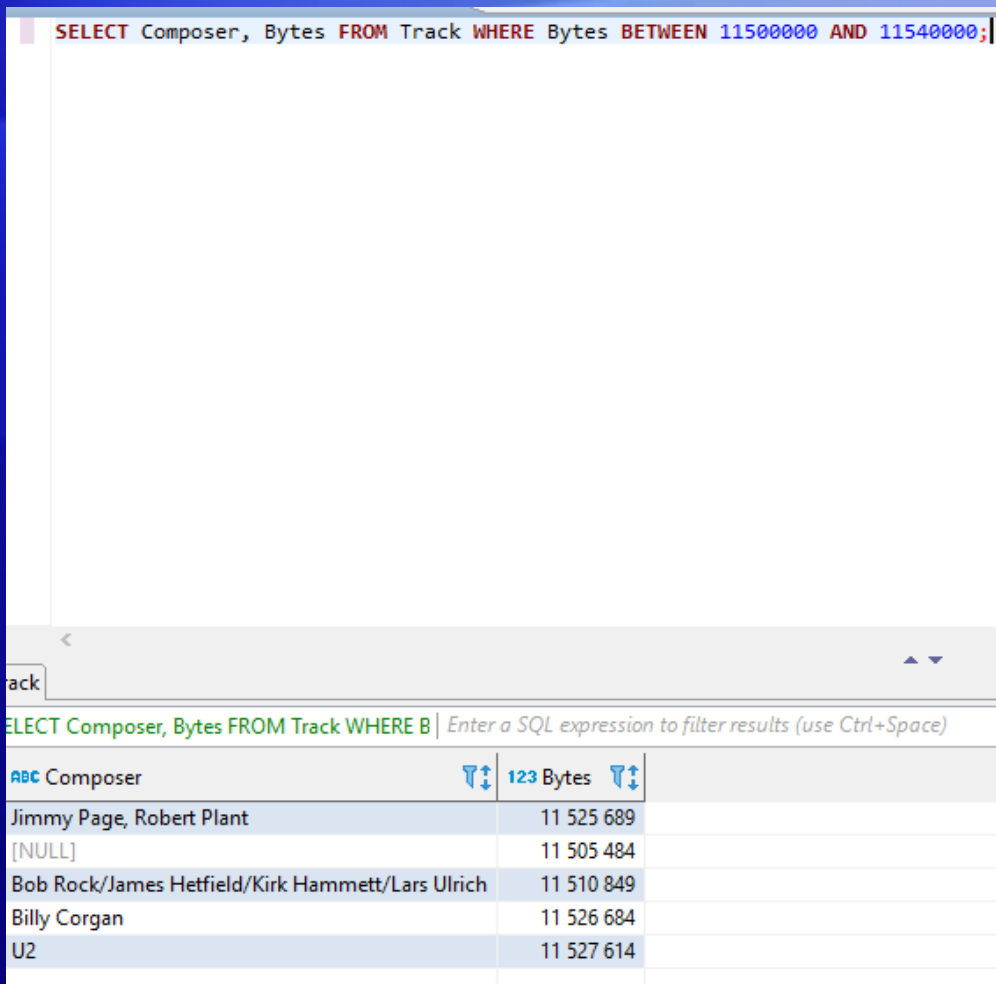
The screenshot shows the DBBeaver interface with a script editor at the top containing the SQL query: `SELECT Composer, Bytes FROM Track WHERE Bytes < 10000000;`. Below the editor, the results of the query are displayed in a table. The table has two columns: 'Composer' and 'Bytes'. The 'Composer' column contains the names of the composers, and the 'Bytes' column contains the size of the compositions in bytes. The results are filtered to show only those compositions where the size is less than 10,000,000 bytes.

Composer	Bytes
Blackmore, Gillan, Glover, Lord, Paice	8 993 056
Ritchie Blackmore, Ian Gillan, Roger Glover, Jon Lord, Ian Paice	6 714 807
Ritchie Blackmore, Ian Gillan, Roger Glover, Jon Lord, Ian Paice	8 072 036
Ritchie Blackmore, Ian Gillan, Roger Glover, Jon Lord, Ian Paice	9 354 480
Richie Blackmore, Ian Gillan, Roger Glover, Jon Lord, Ian Paice	7 929 493
Richie Blackmore, Ian Gillan, Roger Glover, Ian Paice	9 740 361

На рисунке показан случай применения условия “<” на подготовленную выборку из столбцов Composer и Bytes.

Результатом выполнения инструкции `SELECT Composer, Bytes FROM Track WHERE Bytes < 10000000` станет информация о композиторах и их композициях, размер которых строго не более 10000000 байт.

Фильтрация таблицы по строкам с диапазоном



```
SELECT Composer, Bytes FROM Track WHERE Bytes BETWEEN 11500000 AND 11540000;
```

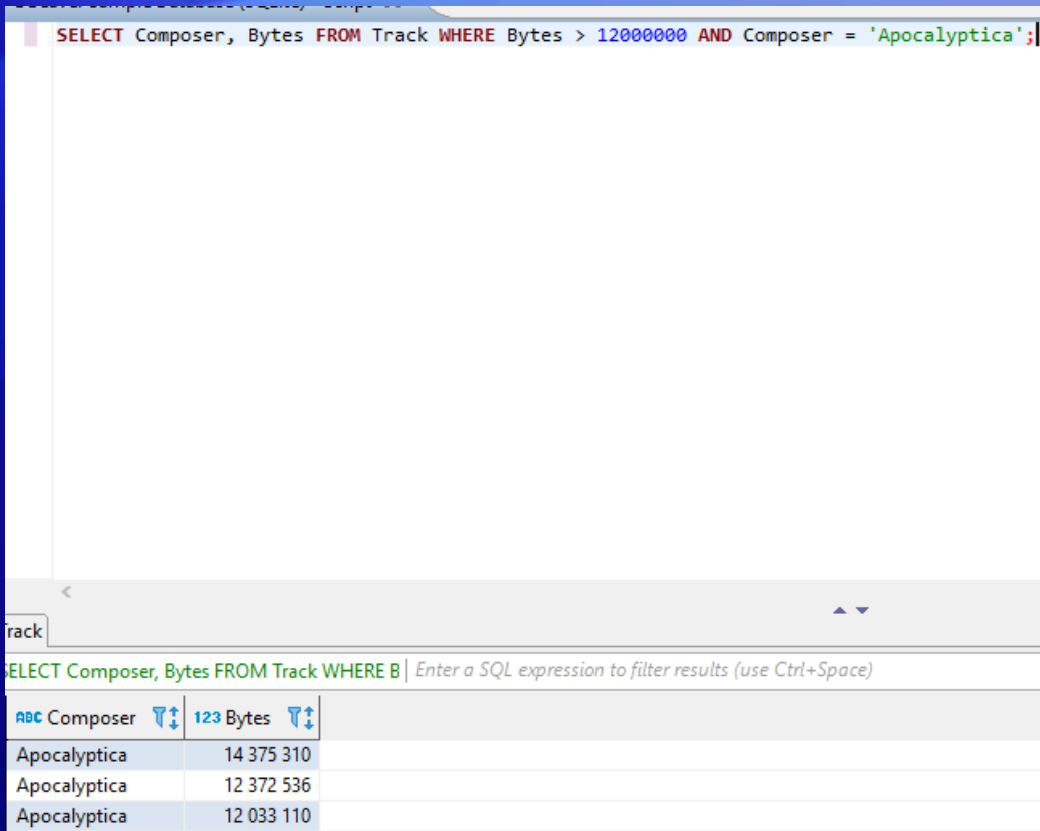
Composer	Bytes
Jimmy Page, Robert Plant	11 525 689
[NULL]	11 505 484
Bob Rock/James Hetfield/Kirk Hammett/Lars Ulrich	11 510 849
Billy Corgan	11 526 684
U2	11 527 614

На рисунке показан случай применения условия “BETWEEN...AND” на подготовленную выборку из столбцов Composer и Bytes.

Результатом выполнения инструкции `SELECT Composer, Bytes FROM Track WHERE Bytes BETWEEN 11500000 AND 11540000` станет информация о композиторах и их композициях, размер которых превышает 11500000, но не превышает 11540000 байт.

Фильтрация таблицы по строкам с логикой

На рисунке показан случай применения условия AND на подготовленную выборку из столбцов Composer и Bytes.



The screenshot shows a SQL query editor with a query window and a results window. The query window contains the following SQL statement:

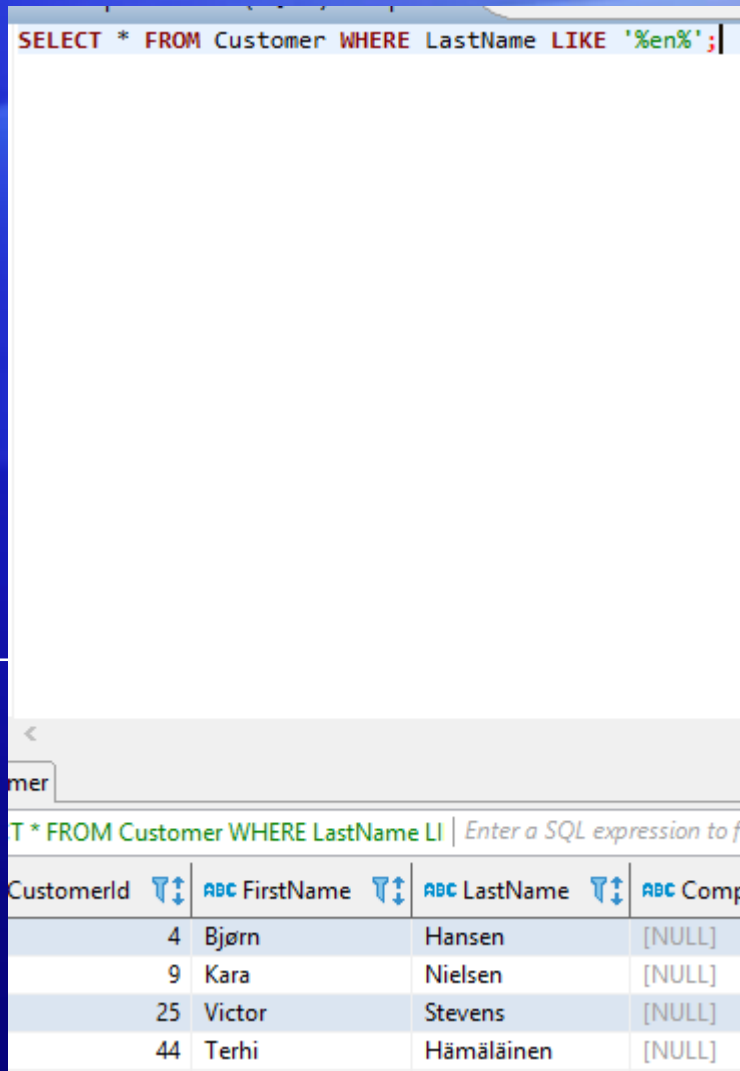
```
SELECT Composer, Bytes FROM Track WHERE Bytes > 12000000 AND Composer = 'Apocalyptica';
```

The results window displays the following table:

Composer	Bytes
Apocalyptica	14 375 310
Apocalyptica	12 372 536
Apocalyptica	12 033 110

Результатом выполнения инструкции `SELECT Composer, Bytes FROM Track WHERE Bytes > 12000000 AND Composer = 'Apocalyptica'` станет информация о композициях Apocalyptica размером более 12000000 байт.

Регулярные выражения и символы подстановки



The screenshot shows a SQL query editor with the following query entered:

```
SELECT * FROM Customer WHERE LastName LIKE '%en%';
```

Below the query editor, the results of the query are displayed in a table. The table has four columns: CustomerId, FirstName, LastName, and CompanyName. The results show four rows of data:

CustomerId	FirstName	LastName	CompanyName
4	Bjørn	Hansen	[NULL]
9	Kara	Nielsen	[NULL]
25	Victor	Stevens	[NULL]
44	Terhi	Hämäläinen	[NULL]

На рисунке показан случай применения регулярного выражения с символом % и оператора LIKE. Символ “%” в регулярном выражении означает, что в выборку можно будет добавить все значения, у которых в столбце LastName будет словосочетание “en”.

Результатом выполнения инструкции `SELECT * FROM Customer WHERE LastName LIKE '%en%'` станет информация о фамилиях покупателей, в которых есть сочетание “en” в любой части слова.

3-ый учебный вопрос

Использование функций

Общие сведения: функции обработки текстовых значений



Инструкции языка, которые используются для обработки текстовых значений. Важно отметить, что синтаксис функций SQL зависит от диалекта SQL.

ASCII	CHAR	CHARINDEX
CONCAT	CONCAT_WS	DIFFERENCE
FORMAT	LEFT	LEN
LOWER	LTRIM	NCHAR
PATINDEX	QUOTENAME	REPLACE
REPLICATE	REVERSE	RIGHT
RTRIM	SOUNDEX	SPACE
STR	STRING_AGG	STRING_ESCAPE
STRING_SPLIT	STUFF	SUBSTRING
TRANSLATE	TRIM	UNICODE
UPPER		

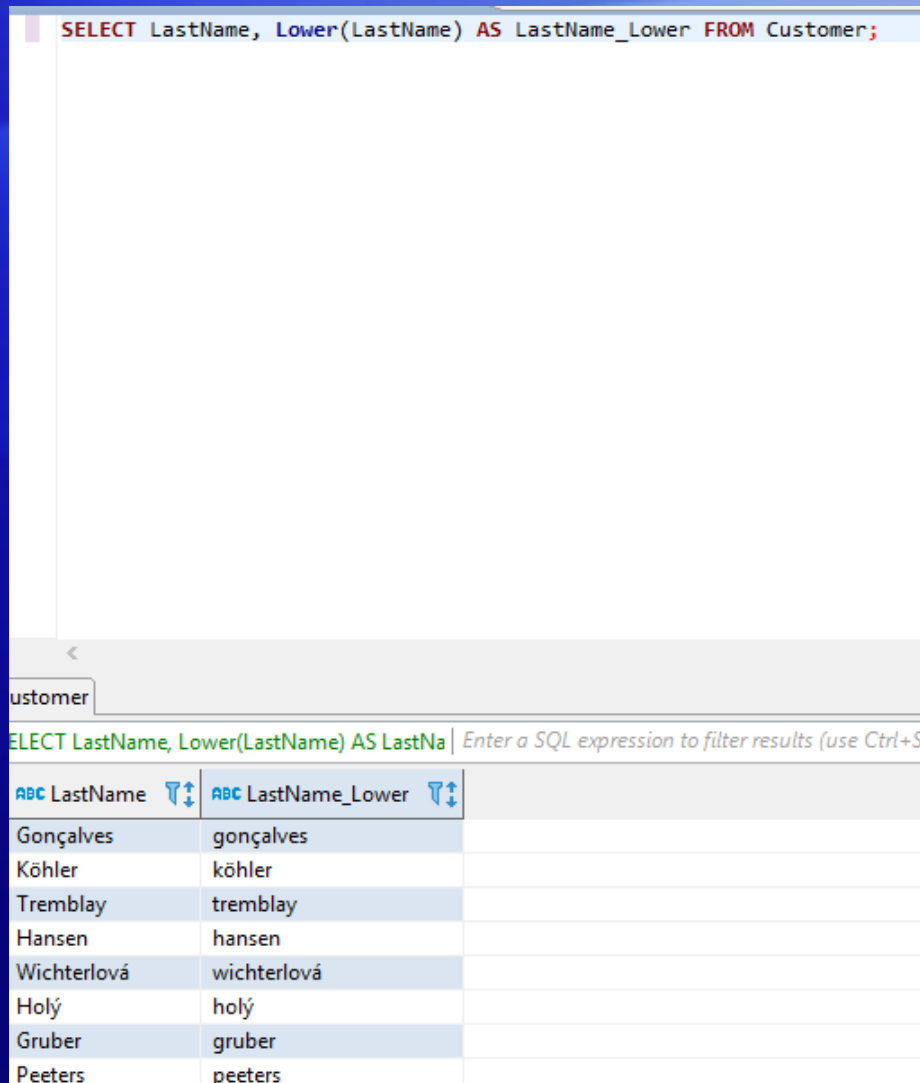
Общие сведения: функции обработки числовых значений



Инструкции языка, которые используются для обработки числовых значений. Важно отметить, что синтаксис функций SQL зависит от диалекта SQL.

ABS	DEGREES	RAND
ACOS	EXP	ROUND
ASIN	FLOOR	SIGN
ATAN	LOG	SIN
ATN2	LOG10	SQRT
CEILING	PI	SQUARE
COS	POWER	TAN
COT	RADIANS	

Функция обработки текста. Lower().



The screenshot shows a SQL query editor with the following query:

```
SELECT LastName, Lower(LastName) AS LastName_Lower FROM Customer;
```

Below the query editor, a table of results is displayed. The table has two columns: 'LastName' and 'LastName_Lower'. The results show the following data:

LastName	LastName_Lower
Gonçalves	gonçalves
Köhler	köhler
Tremblay	tremblay
Hansen	hansen
Wichterlová	wichterlová
Holý	holý
Gruber	gruber
Peeters	peeters

На рисунке показан случай применения функции обработки текста Lower(). Данная функция для указанного столбца заменит все символы верхнего регистра на символы нижнего регистра.

Обратите внимание, что для демонстрации, результат выполнения функции выведен в отдельный, физически не существующий столбец.

4-ый учебный вопрос

Типы данных

Общие сведения: типы данных SQL



В SQL Server у каждого столбца, локальной переменной, выражения и параметра есть определенный тип данных.

Тип данных - атрибут, определяющий, какого рода данные могут храниться в объекте: целые числа, символы, данные денежного типа, метки времени и даты, двоичные строки и так далее.

Важно отметить, что набор типов данных, которым может оперировать проектировщик базы данных или аналитик очень сильно зависит от СУБД в которой осуществляется работа. Так, например, типы данных MS SQL Server отличаются от типов данных Oracle.

Также, типы данных SQL слабо коррелируют с типами данных языков программирования.

Типы данных MS SQL Server

Тип Данных	Описание
Binary	Массив двоичных данных, длина от 0 до 8000 байт
Char	Массив символьных данных, длин от 0 до 8000 байт
Datetime	Дата и время длиной 8 байт
Decimal	Десятичное число, можно задавать точность и масштаб
Float	Число с плавающей точкой длиной 8 байт
Image	Массив двоичных данных переменной длины
Int	4-хбайтовое целое число
Money	Денежная сумма, длин 8 байт
Numeric	Аналогично Decimal

Типы данных MS SQL Server

Тип Данных	Описание
Real	4-ехбайтовое число с плавающей точкой
Smalldatetime	Дата и время длиной 4 байт
Smallint	2-хбайтовое целое число
Smallmoney	Денежная сумма, длина 4 байт
Text	Текст переменной длины
Tinyint	1-нобайтовое целое от 0 до 255
Varchar	Массив символьных данных переменной длины

Типы данных Oracle

Тип Данных	Описание
BLOB	Большой двоичный объект
CHAR(n)	Текстовое поле фиксированной длины n
DATE	Поле длиной 7 байт
INT	Целое число длиной 7 знаков
NUMBER (n,d)	Число длиной n с d знаками после запятой
VARCHAR(n)	Текстовое поле длины до n символов

5-ый учебный вопрос

Обработка неопределенных
значений

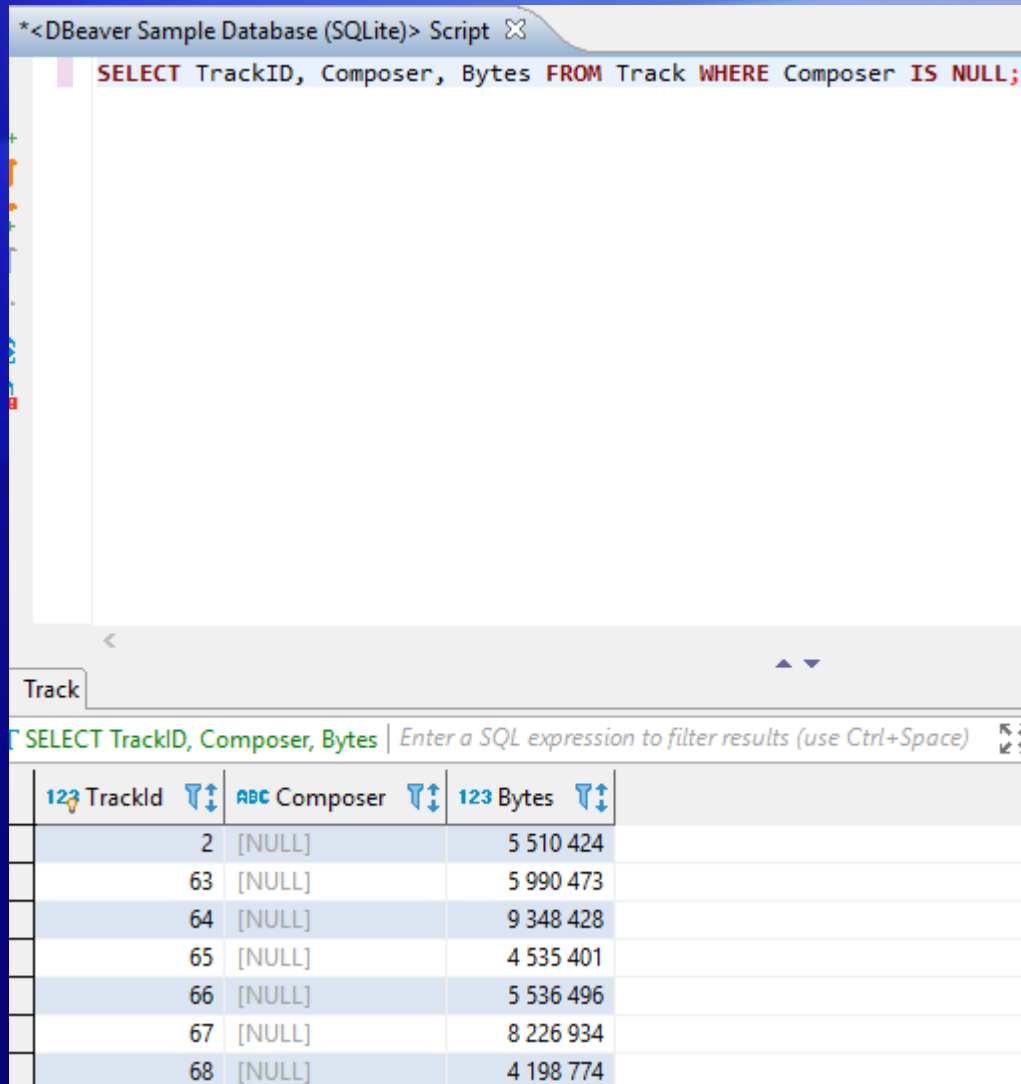
Общие сведения: неопределенные значения NULL



NULL указывает на то, что значение неизвестно. NULL отличается от пустого или нулевого значения. Два значения NULL считаются эквивалентными. Сравнения между двумя значениями NULL или между значением NULL и каким-либо иным значением возвращают неизвестную величину, так как значение каждого из значений NULL неизвестно.

Значение NULL обычно указывает на то, что данные неизвестны, неприменимы или будут добавлены позже. Например, на момент размещения клиентом заказа может быть неизвестно его отчество.

Проверка на наличие неопределенных значений



The screenshot shows the DBeaver SQL editor with a query window titled '*<DBeaver Sample Database (SQLite)> Script'. The query is: `SELECT TrackID, Composer, Bytes FROM Track WHERE Composer IS NULL;`. Below the query editor, a table named 'Track' is displayed, showing the results of the query. The table has three columns: 'TrackId', 'Composer', and 'Bytes'. The 'Composer' column contains '[NULL]' for all rows, indicating that the composer names are undefined.

TrackId	Composer	Bytes
2	[NULL]	5 510 424
63	[NULL]	5 990 473
64	[NULL]	9 348 428
65	[NULL]	4 535 401
66	[NULL]	5 536 496
67	[NULL]	8 226 934
68	[NULL]	4 198 774

На рисунке показан случай применения проверки на наличие неопределенных значений(NULL). Данная функция вернет все строки таблицы Track, для которых название Композитора – неопределенное значение.

Очевидно, что предикат IS NOT NULL наоборот, исключит все строки с неопределенными значениями из результата выборки.

6-ой учебный вопрос

Сортировка

Общие сведения: сортировка, предложение ORDER BY

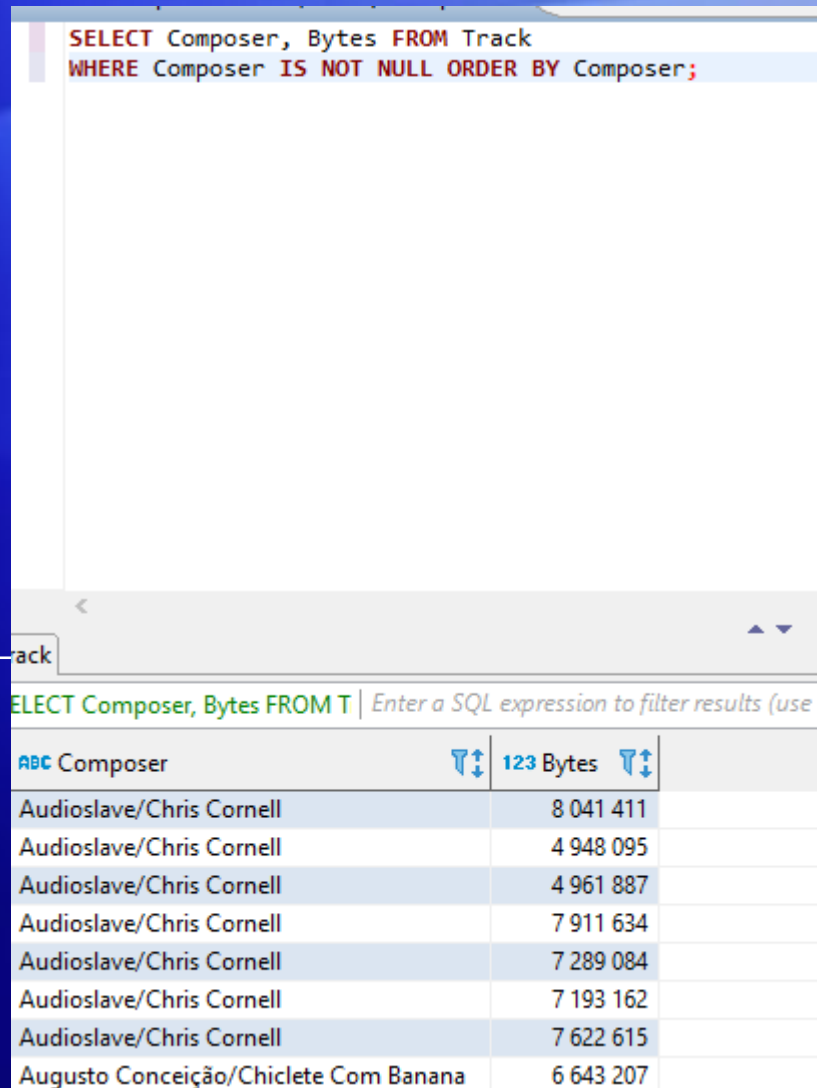


Сортирует данные, возвращенные запросом в SQL Server. Это предложение используется для следующих целей:

Упорядочение результирующего набора запроса по заданному списку столбцов и (дополнительно) ограничение числа возвращаемых строк указанным диапазоном. Порядок, в котором строки возвращаются в результирующем наборе, не гарантируется, если не указано предложение ORDER BY.

Определение порядка, в котором значения ранжирующей функции применяются к результирующему набору.

Простая сортировка на единственном столбце



The screenshot shows a SQL query editor with the following query:

```
SELECT Composer, Bytes FROM Track
WHERE Composer IS NOT NULL ORDER BY Composer;
```

Below the query editor, the results are displayed in a table. The table has two columns: 'Composer' and 'Bytes'. The results are sorted alphabetically by the 'Composer' column.

Composer	Bytes
Audioslave/Chris Cornell	8 041 411
Audioslave/Chris Cornell	4 948 095
Audioslave/Chris Cornell	4 961 887
Audioslave/Chris Cornell	7 911 634
Audioslave/Chris Cornell	7 289 084
Audioslave/Chris Cornell	7 193 162
Audioslave/Chris Cornell	7 622 615
Augusto Conceição/Chiclete Com Banana	6 643 207

На рисунке показана сортировка командой ORDER BY без дополнительных условий. Из таблицы Track возвращены данные из столбцов Composer и Bytes без неопределенных значений в столбце Composer и отсортированные по столбцу Composer.

По умолчанию, текстовые столбцы сортируются по алфавиту от А до Z.

Сортировка на столбце, не выводящемся в результатный набор

Несмотря на то, что в конечном наборе отсутствует столбец TrackID, сортировка по нему осуществлена.

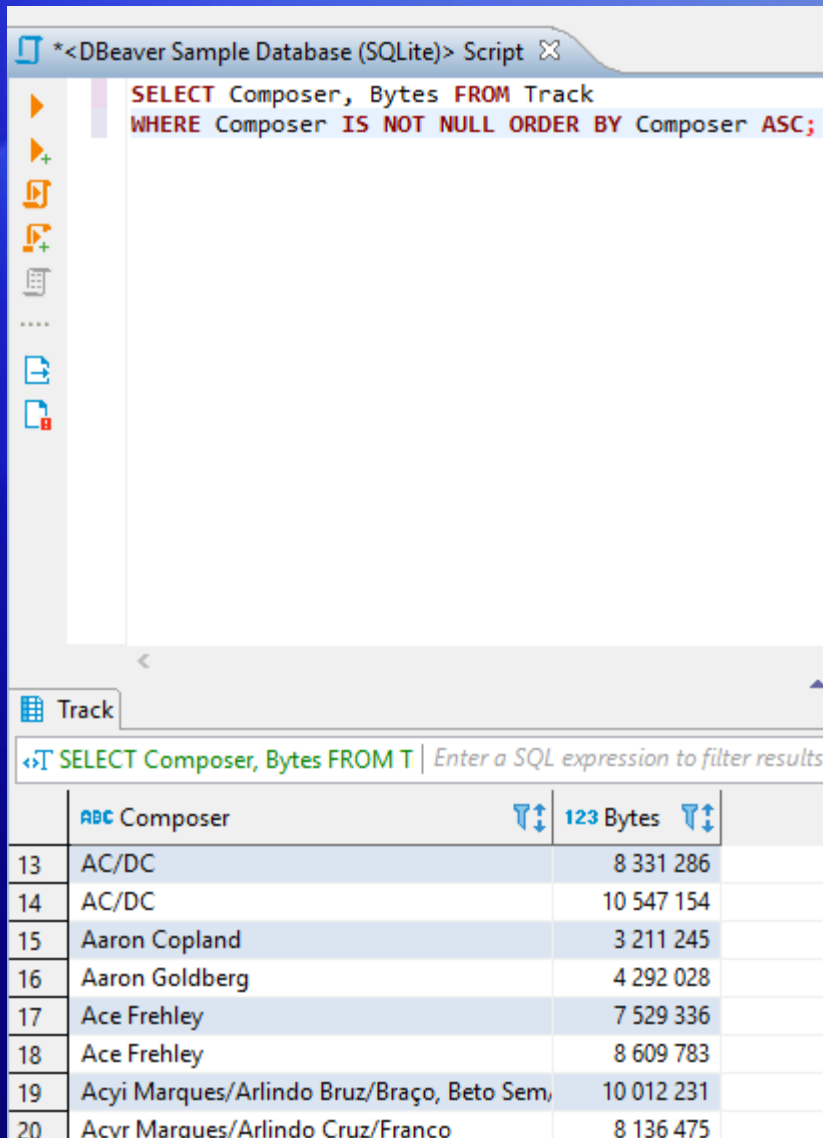
```
SELECT Composer, Bytes FROM Track
WHERE Composer IS NOT NULL ORDER BY TrackID;
```

ack

SELECT Composer, Bytes FROM T | Enter a SQL expression to filter results (use Ctrl+Space)

ABC Composer	123 Bytes
Jorge Ben	5 842 196
Tradicional	3 225 364
Tom Jobim - Newton Mendonça	4 865 597
Caetano Veloso	15 889 054
Caetano Veloso	7 382 703
Caetano Veloso	2 752 161
Gilberto Gil	4 333 651

Сортировка с явным указанием по возрастанию



The screenshot shows the DBeaver SQL editor interface. The top pane displays the following SQL query:

```
SELECT Composer, Bytes FROM Track
WHERE Composer IS NOT NULL ORDER BY Composer ASC;
```

The bottom pane shows the results of the query in a table format. The table has two columns: 'Composer' and 'Bytes'. The results are sorted by 'Composer' in ascending order. The first two rows (13 and 14) show 'AC/DC' with 'Bytes' values of 8 331 286 and 10 547 154 respectively. The subsequent rows list various composers and their corresponding byte values.

	Composer	Bytes
13	AC/DC	8 331 286
14	AC/DC	10 547 154
15	Aaron Copland	3 211 245
16	Aaron Goldberg	4 292 028
17	Ace Frehley	7 529 336
18	Ace Frehley	8 609 783
19	Acyi Marques/Arlindo Bruz/Braço, Beto Sem,	10 012 231
20	Acyr Marques/Arlindo Cruz/Franco	8 136 475

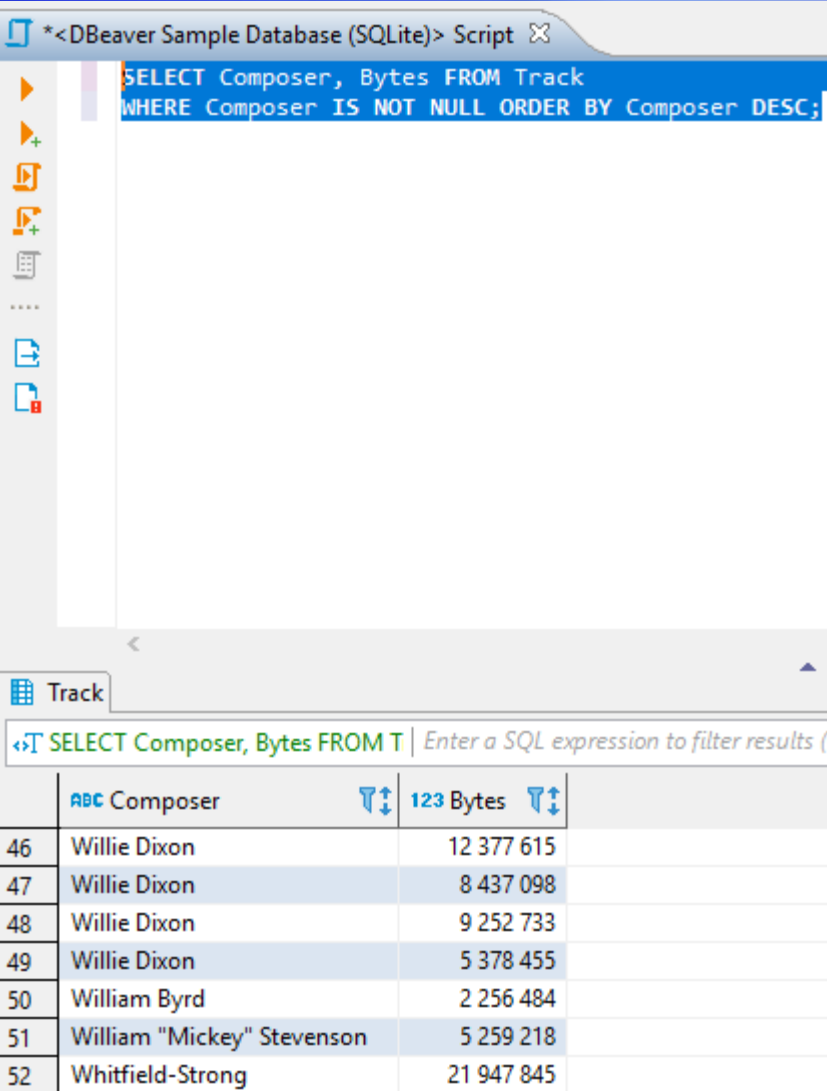
К команде ORDER BY была добавлена условие ASC (по возрастанию). В случае, когда сортировок в одной выборке несколько (пример чуть ниже), явное указание направления сортировки обязательно.

Условие ASC позволяет сортировать не только символьные, но и числовые значения.

Сортировка с явным указанием по убыванию

К команде ORDER BY была добавлена условие DESC (по убыванию). В случае, когда сортировок в одной выборке несколько (пример чуть ниже), явное указание направления сортировки обязательно.

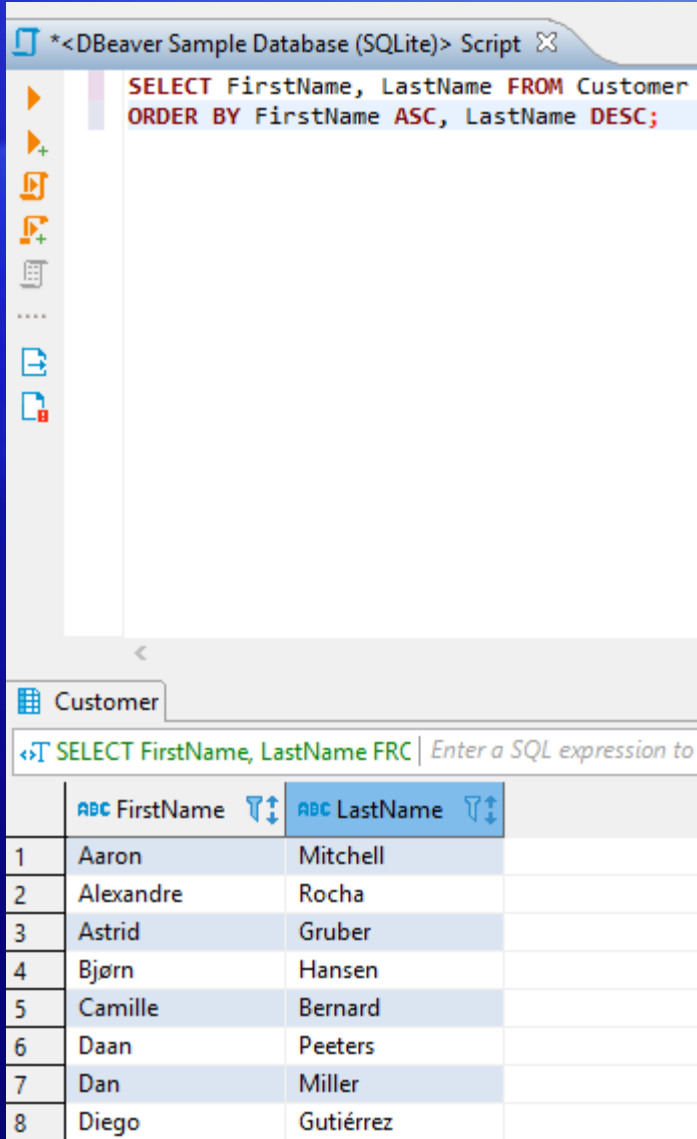
Условие DESC также позволяет сортировать не только символьные, но и числовые значения.



The screenshot shows the DBBeaver interface. The top pane displays a SQL script: `SELECT Composer, Bytes FROM Track WHERE Composer IS NOT NULL ORDER BY Composer DESC;`. The bottom pane shows the results of this query in a table format. The table has two columns: 'Composer' and 'Bytes'. The results are sorted by 'Composer' in descending order. The first four rows all have 'Willie Dixon' as the composer, followed by 'William Byrd', 'William "Mickey" Stevenson', and 'Whitfield-Strong'.

	ABC Composer	123 Bytes
46	Willie Dixon	12 377 615
47	Willie Dixon	8 437 098
48	Willie Dixon	9 252 733
49	Willie Dixon	5 378 455
50	William Byrd	2 256 484
51	William "Mickey" Stevenson	5 259 218
52	Whitfield-Strong	21 947 845

Сортировка по нескольким столбцам



The screenshot shows the DBeaver SQL editor interface. The top pane displays a SQL query: `SELECT FirstName, LastName FROM Customer ORDER BY FirstName ASC, LastName DESC;`. The bottom pane shows the results of the query, sorted by first name (ASC) and then by last name (DESC). The results are displayed in a table with two columns: 'FirstName' and 'LastName'.

	ABC FirstName	ABC LastName
1	Aaron	Mitchell
2	Alexandre	Rocha
3	Astrid	Gruber
4	Bjørn	Hansen
5	Camille	Bernard
6	Daan	Peeters
7	Dan	Miller
8	Diego	Gutiérrez

На рисунке показано, что сортировка возможна и по нескольким полям. В таблице Customer Имя будет сортироваться по возрастанию (ASC), и Фамилия – по убыванию (DESC).

Результаты сортировки (как, впрочем и все результаты выборки) в таблице не сохраняются.

7-ой учебный вопрос

Операция TOP

Общие сведения: операция TOP



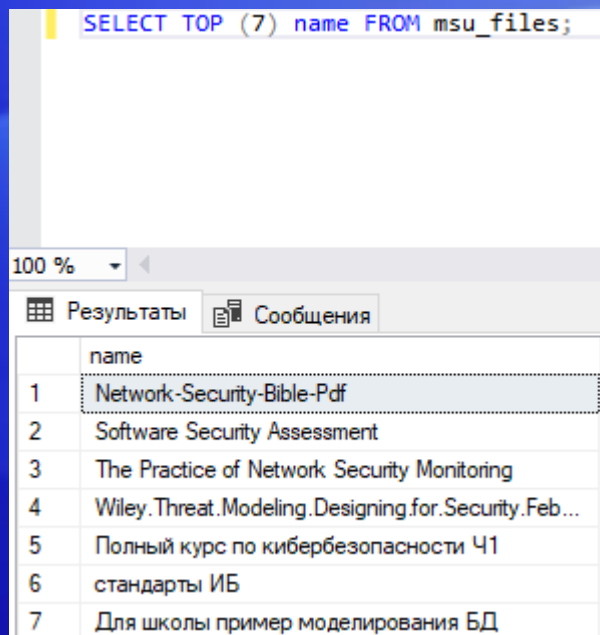
Ограничивает число строк, возвращаемых в результирующем наборе запроса до заданного числа или процентного значения в SQL Server.

Если TOP используется с предложением ORDER BY, в результирующий набор включаются только первые N строк отсортированного результата. В противном случае TOP возвращает N строк в неопределенном порядке.

Это предложение позволяет ограничить число строк, возвращаемых инструкцией SELECT.

Также TOP можно использовать для указания числа строк, затрагиваемых инструкциями INSERT, UPDATE или DELETE.

Применение TOP без сортировки



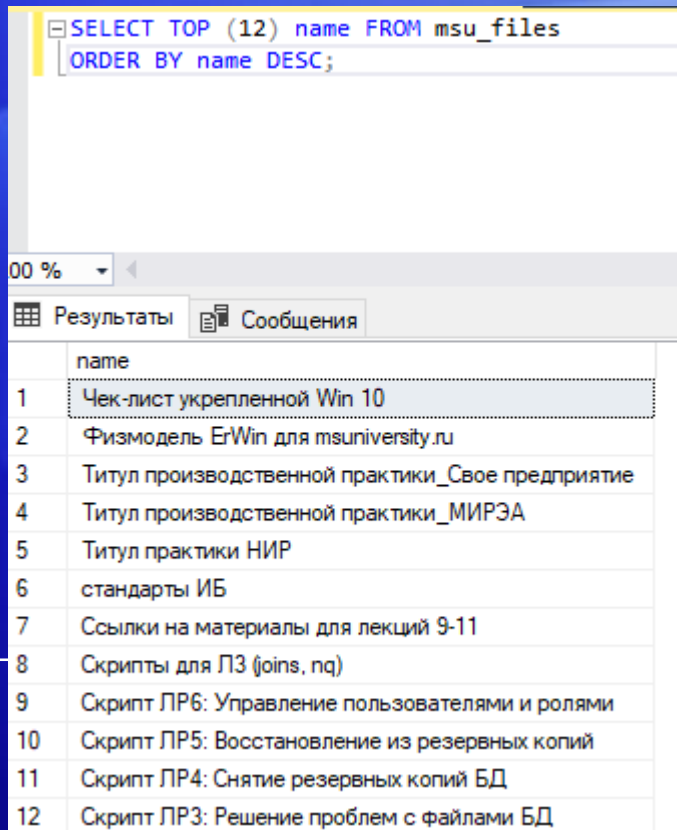
```
SELECT TOP (7) name FROM msu_files;
```

	name
1	Network-Security-Bible-Pdf
2	Software Security Assessment
3	The Practice of Network Security Monitoring
4	Wiley.Threat.Modeling.Designing.for.Security.Feb...
5	Полный курс по кибербезопасности Ч1
6	стандарты ИБ
7	Для школы пример моделирования БД

На рисунке функция TOP применяется для того, чтобы из таблицы msu_files произвольно выбрать и вывести в результат 7 произвольных значений из столбца name.

Для получения, например, первых n значений или последних n значений, к операции TOP необходимо добавить соответствующую сортировку.

Применение TOP с сортировкой



The screenshot shows a SQL query window with the following query:

```
SELECT TOP (12) name FROM msu_files  
ORDER BY name DESC;
```

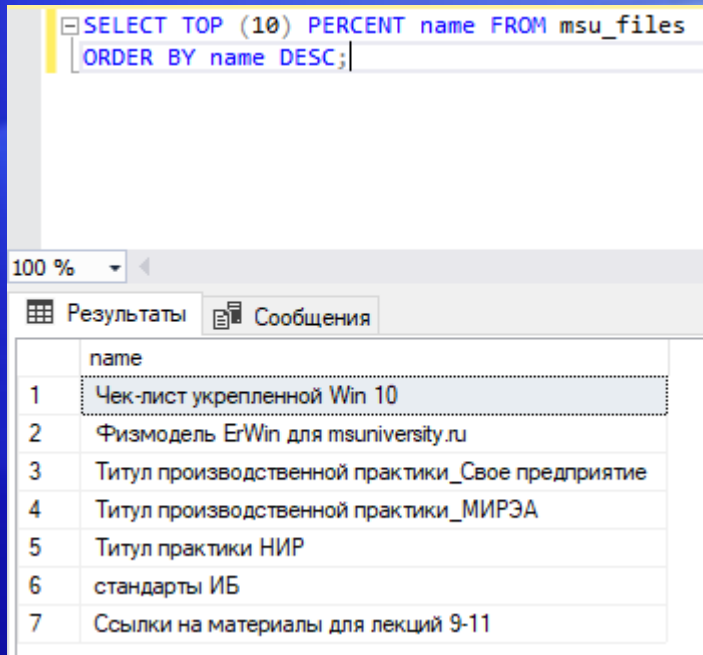
Below the query, the results are displayed in a table with two columns: a row number and the file name. The results are sorted in descending order of the file names.

	name
1	Чек-лист укрепленной Win 10
2	Физмодель ErWin для msuniversity.ru
3	Титул производственной практики_Свое предприятие
4	Титул производственной практики_МИРЭА
5	Титул практики НИР
6	стандарты ИБ
7	Ссылки на материалы для лекций 9-11
8	Скрипты для ЛЗ (joins, nq)
9	Скрипт ЛР6: Управление пользователями и ролями
10	Скрипт ЛР5: Восстановление из резервных копий
11	Скрипт ЛР4: Снятие резервных копий БД
12	Скрипт ЛР3: Решение проблем с файлами БД

На рисунке функция TOP применяется для того, чтобы из таблицы msu_files выбрать и вывести в результат 7 нижних значений из столбца name при сортировке по этому же столбцу.

Для получения, первых n значений добавляется сортировка ORDER BY...ASC.

Применение TOP с процентами



The screenshot shows a SQL query window with the following text:

```
SELECT TOP (10) PERCENT name FROM msu_files  
ORDER BY name DESC;
```

Below the query window, the 'Results' tab is active, displaying a table with 7 rows. The first row is highlighted. The table has a single column named 'name'.

	name
1	Чек-лист укрепленной Win 10
2	Физмодель ErWin для msuniversity.ru
3	Титул производственной практики_Свое предприятие
4	Титул производственной практики_МИРЭА
5	Титул практики НИР
6	стандарты ИБ
7	Ссылки на материалы для лекций 9-11

На рисунке функция TOP применяется с дополнительным указанием PERCENT, выводя в результате последние 10 процентов значений столбца name из таблицы msu_files.

8-ой учебный вопрос

Устранение дубликатов

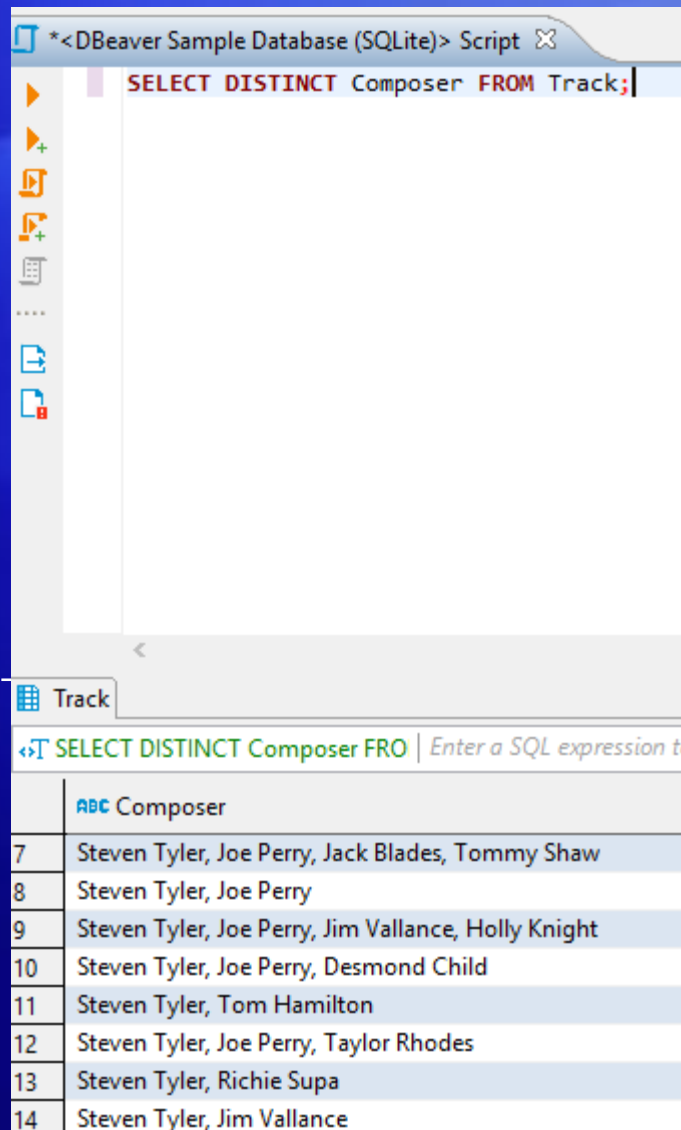
Общие сведения: функция DISTINCT



Вычисляет заданный набор, удаляя из него повторяющиеся кортежи, и возвращает результирующий набор.

Необходимо отметить, что этот набор не исключает дубликаты из таблицы полностью, а просто выводится с их исключением. Для полного удаления строк-дубликатов, необходимо или виртуальной таблицей или оконными функциями MS SQL Server.

Исключение из запроса дублирующихся значений



The screenshot shows the DBeaver SQL editor interface. The top pane displays the SQL query: `SELECT DISTINCT Composer FROM Track;`. The bottom pane shows the results of the query in a table format. The table has a single column labeled 'Composer' with a list of unique values. The results are as follows:

	Composer
7	Steven Tyler, Joe Perry, Jack Blades, Tommy Shaw
8	Steven Tyler, Joe Perry
9	Steven Tyler, Joe Perry, Jim Vallance, Holly Knight
10	Steven Tyler, Joe Perry, Desmond Child
11	Steven Tyler, Tom Hamilton
12	Steven Tyler, Joe Perry, Taylor Rhodes
13	Steven Tyler, Richie Supa
14	Steven Tyler, Jim Vallance

На рисунке функция DISTINCT исключает из выборки по столбцу Composer из таблицы Track все повторяющиеся значения.

9-ой учебный вопрос

Порядок операций в запросе

Общие сведения: порядок обработки инструкции SELECT



Следующие действия демонстрируют логический порядок обработки или порядок привязки инструкции SELECT. Этот порядок определяет, когда объекты, определенные в одном шаге, становятся доступными для предложений в последующих шагах.

1. FROM
2. ON
3. JOIN
4. WHERE
5. GROUP BY
6. WITH CUBE или WITH ROLLUP
7. HAVING
8. SELECT
9. DISTINCT
10. ORDER BY
11. В начало