База Данных

Лекция 9

Запросы

Общая структура запроса

При создании SQL-запроса необходимо учитывать следующие пять моментов.

1. С какой базой данных мы работаем?
2. Из какой таблицы в этой базе данных нам необходимо извлечь данные?
3. Какие поля в этой таблице нас интересуют?
4. Хотим ли мы исключить какие-либо данные, отфильтровать или исключить какой-либо диапазон или период времени?
5. Как сформулировать наш запрос одним простым предложением на человеческом языке

Оператор SELECT

- Операторы запроса облегчают извлечение данных из одной или нескольких таблиц.

- Результатом любого запроса является таблица. Результатом можно дополнительно манипулировать с помощью других операций запроса.

- Синтаксис:
  SELECT attribute(s) FROM table(s)
  [WHERE selection condition(s)];

Оператор SELECT

- **Вывод всех данных из таблицы**
- Для вывода всех полей из определённой таблицы используется символ *.
- Синтаксис:
  SELECT * FROM Students;

- **Вывод данных из определённых колонок таблицы**
- Если необходимо вывести информацию только по определённым столбцам таблицы, а не всю сразу, то это можно сделать перечисляя названия столбцов через запятую:

Синтаксис:
SELECT First_name, Last_name FROM Students;

Псевдонимы (Aliasing) в SQL

- Псевдоним в SQL присваивает таблице или столбцу временное имя в запросе. Псевдонимы существуют только во время выполнения запроса. При добавлении
- псевдонима данные в базе данных не изменяются. Псевдонимы изменяют только
- способ отображения полей на панели результатов

- Ниже показан синтаксис псевдонима таблицы:

```
SELECT column_list
 FROM table_name AS alias_name;
```

- Ключевое слово AS в синтаксисе псевдонима таблицы является необязательным.
- **ПРИМЕЧАНИЕ** Не используйте никакие ключевые слова SQL в качестве псевдонимов. Это вызовет путаницу или синтаксические ошибки. РСУБД может интерпретировать данный псевдоним как команду.

Псевдонимы (Aliasing) в SQL

Пример:

/* Данный запрос осуществляет выборку полей имени, фамилии, электронной почты и номера телефона из таблицы customers (клиенты) и демонстрирует четыре различных способа использования псевдонима. */

```
SELECT
    FirstName AS 'First Name’,
    LastName AS [Last Name],
    Email AS EMAIL
    Phone CELL
FROM customers
```

Если созданный вами псевдоним содержит несколько слов (например, имя и фамилию), его необходимо разграничить, в данном случае либо одинарными кавычками '', либо квадратными скобками [], как показано в примере.

Псевдонимы (Aliasing) в SQL

Результат примера:

БЕЗ ПСЕВДОНИМА

	FirstName	LastName	Email	Phone
1	Luís	Gonçalves	luisg@embraer.com.br	+55 (12) 3923-5555
2	Leonie	Köhler	leonekohler@surfeu.de	+49 0711 2842222
3	François	Tremblay	ftremblay@gmail.com	+1 (514) 721-4711
4	Bjørn	Hansen	bjorn.hansen@yahoo.no	+47 22 44 22 22
5	František	Wichterlová	frantisekw@jetbrains.com	+420 2 4172 5555

С ПСЕВДОНИМОМ

	First Name	Last Name	EMAIL	CELL
1	Luís	Gonçalves	luisg@embraer.com.br	+55 (12) 3923-5555
2	Leonie	Köhler	leonekohler@surfeu.de	+49 0711 2842222
3	François	Tremblay	ftremblay@gmail.com	+1 (514) 721-4711
4	Bjørn	Hansen	bjorn.hansen@yahoo.no	+47 22 44 22 22
5	František	Wichterlová	frantisekw@jetbrains.com	+420 2 4172 5555

Псевдонимы (Aliasing) в SQL

Псевдоним таблицы имеет несколько применений:

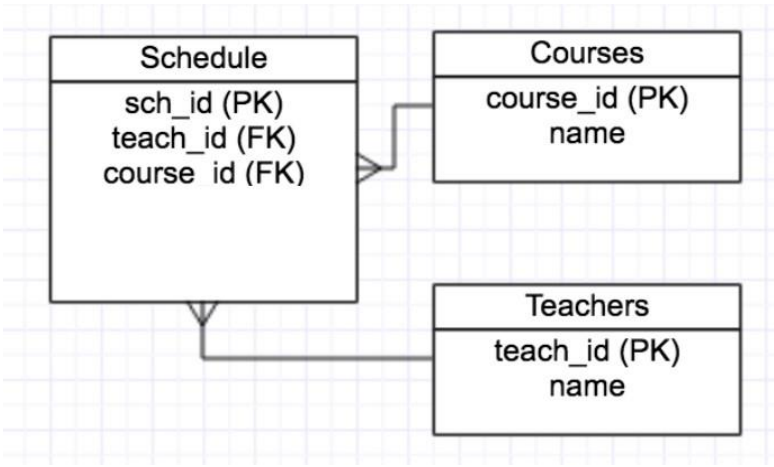
- Во-первых, если вам необходимо указать имя столбца с длинным именем таблицы, вы можете использовать псевдоним таблицы, чтобы сделать ваш запрос более читабельным.

- Практическое применение - когда вы запрашиваете данные из нескольких таблиц с одинаковыми именами столбцов. В этом случае вы должны указать столбцы, используя имена таблиц.

Псевдонимы (Aliasing) в SQL

- Использование псевдонимов в названиях таблиц при соединении таблиц (join) делает их намного более компактными и читабельными.

```
SELECT c.name, t.name
    FROM Courses c, Teachers t, Schedule s
    WHERE c.course_id = s.course_id AND
    t.teach_id = s.teach_id;
```



Псевдонимы (Aliasing) в SQL

- Ниже показан синтаксис псевдонима столбца:

```
SELECT column_name AS alias_name
FROM table_name;
```

  - В этом синтаксисе column_name присваивается как alias_name. Ключевое слово AS необязательное.

  - Пример переименования столбца fname в First_Name:

```
SELECT fname AS First_name
FROM Students;
```

Конкатенация строк

- **Конкатенация** — объединение двух или более строк. Для слияния двух полей вместе используется символ ||.
- Например в таблице Students имя и фамилия хранятся как два отдельных атрибута. Для объединения их в один столбец используйте оператор ||

```
SELECT fname || lname
FROM Students;
```

- Оператор конкатенации || просто соединил оба поля вместе без пробелов. Для удобства чтения мы можем последовательно использовать две конкатенации и заключить пробел в одинарные кавычки:

```
SELECT fname || ‘ ‘ || lname
FROM Students;
```

Конкатенация строк

- Рассмотрим следующий пример, где для создания в одной строке имени и адреса клиента используется множественная конкатенация.
- SELECT FirstName || ' ' || LastName || ' ' || Address || ', ' || City || ', ' || State || ' ' || PostalCode AS [MailingAddress] FROM customers WHERE Country = "USA" При выполнении запроса получим следующий результат:

	MailingAddress
1	Frank Harris 1600 Amphitheatre Parkway, Mountain View, CA 94043-1351
2	Jack Smith 1 Microsoft Way, Redmond, WA 98052-8300
3	Michelle Brooks 627 Broadway, New York, NY 10012-2612
4	Tim Goyer 1 Infinite Loop, Cupertino, CA 95014
5	Dan Miller 541 Del Medio Avenue, Mountain View, CA 94040-111
	13 строк получено за 5 мс

(Источник рисунка: Шилдс Уолтер. SQL: быстрое погружение)

Условие Distinct

- Условие DISTINCT используется в конструкции SELECT для удаления повторяющихся строк из результирующего набора.
- Условие DISTINCT сохраняет по одной строке для каждой группы дубликатов.

Синтаксис предложения DISTINCT:

```
SELECT DISTINCT column_name
FROM table_name;
```

- Чтобы выбрать разные фамилии из числа студентов:

```
SELECT DISTINCT lname
FROM Students;
```

Условие Distinct для нескольких столбцов

При использовании оператора DISTINCT для двух и более столбцов будут удаляться записи, которые имеют одинаковые значения по всем полям.

```
SELECT DISTINCT column_1, column_2
FROM table_name;
```

В этом случае для вычисления дубликата будет использоваться комбинация как column_1, так и column_2.

Условие Distinct для нескольких столбцов

То есть для такой таблицы:

first_name	last_name
John	Scott
William	Dawson
Raul	Hartman
William	Hartman
John	Scott
John	Hartman

запрос с оператором DISTINCT вернул бы все сочетания имён и фамилий кроме дублирующихся «John Scott».

```
SELECT DISTINCT first_name, last_name FROM User;
```

first_name	last_name
John	Scott
William	Dawson
Raul	Hartman
William	Hartman
John	Hartman

IS NULL / IS NOT NULL

- **IS NULL** в условии WHERE вернет только нулевые значения.
- **NOT NULL** вернет только значения, которые не были нулевыми.
- При работе со значениями **NULL** необходимо использовать операторы **IS** и **NOT**, а не оператор равенства =. Нулевые значения указывают на недостаток данных. Оператор = сравнивает значения двух элементов. Нулевые значения не содержат значений, поэтому их нельзя сравнивать, используя оператор =. Использование оператора = приведет к ошибке.

IS NULL / IS NOT NULL

- Рассмотрим следующий запрос:

```
SELECT stud_id, fname
FROM Students
WHERE group_id IS NULL;
```

- Этот запрос возвращает запись о каждом студенте, где group_id равен null (пусто).

IS NULL / IS NOT NULL

Таблица Students

stud_id	fname	group_id
1	student1	2
2	student2	2
3	student3	

… WHERE group_id IS NULL;

stud_id	fname
3	student3

… WHERE group_id IS NOT NULL;

stud_id	fname
1	student1
2	student2

Операторы сравнения

- Одним из наиболее распространенных условий выбора является условие диапазона. Условие диапазона фильтрует результаты, в которых значения в столбце находятся между одним или двумя значениями.

- Существует два способа выполнить операцию диапазона:

С помощью операторов <, <=, >, >=.

С помощью оператора BETWEEN.

Операторы сравнения

Operator	Description
<	меньше
>	больше
<=	меньше или равно
>=	больше или равно
=	равно
<> or !=	Не равно

Операторы сравнения

- Условие диапазона задается с помощью операторов <,<=,> и >= в качестве

```
SELECT … FROM …
WHERE column < value1 AND column > value2;
```

- Пример: Запросите имена и фамилии всех студентов со средним баллом от 3 до 4:

```
SELECT fname, lname
FROM Students
WHERE gpa >= 3 AND gpa <= 4;
```

Оператор BETWEEN

- Оператор **BETWEEN** задает диапазон для проверки условия. Для определения необходимого диапазона значений вместе с оператором **BETWEEN** используется оператор **AND**.

- Вы должны ввести ключевое слово **BETWEEN** с начальным значением, ключевое **AND** и конечное значение. В отличие от **IN, BETWEEN** чувствителен к порядку

- Запрос на предыдущем слайде можно переписать следующим образом

```
SELECT fname, lname
FROM Students
WHERE gpa BETWEEN 3 AND 4;
```

Операторы сравнения

- Оператор BETWEEN имеет отрицание: NOT BETWEEN

- Оператор BETWEEN определен для большинства типов данных, включая числовые и временные данные.

BETWEEN / NOT BETWEEN

BETWEEN обрабатывает значения как включенные в заданный диапазон. NOT BETWEEN выполняет противоположное сравнение.

a BETWEEN x AND y
Эквивалентно
a >= x AND a <= y

a NOT BETWEEN x AND y
Эквивалентно
a < x OR a > y

ОПЕРАТОР LIKE

- Оператор LIKE используется при условных запросах, когда мы хотим узнать соответствует ли строка определённому шаблону.
- Синтаксис
- ... WHERE поле_таблицы [NOT] LIKE шаблон_строки
- Шаблон может включать следующие специальные символы:

Символ	Описание
%	Последовательность любых символов (число символов в последовательности может быть от 0 и более)
_	Любой единичный символ

ОПЕРАТОР LIKE

- Символ % соответствует произвольному количеству символов, включая пробелы.
- Таким образом, vince% будет соответствовать каждому из следующих: vince, vincent, vincenzo

- Символ _ соответствует одному произвольному символу.
- Итак, v_nce будет соответствовать каждому из следующих значений: vince, vance, vbnce, vnnce, v1nce, и так далее.

ОПЕРАТОР LIKE

- Пример с %: Вывести номер телефона, если он начинается с 412.

```
SELECT phone
FROM Students
WHERE phone LIKE ‘412%’;
```

ОПЕРАТОР LIKE

- Пример с _: Вывести номер телефона, если он начинается с "20" и заканчивается на "-555-4335’.

```
SELECT phone
FROM Students
WHERE phone LIKE ‘20_-555-4335’;
```

ОПЕРАТОР LIKE

- Оператор LIKE также используется для исключения результатов, соответствующих указанным параметрам. Для этого необходимо поставить ключевое слово NOT перед оператором LIKE, и вы сможете исключить записи из результата запроса.

```
SELECT phone
FROM Students
WHERE phone NOT LIKE ‘20_-555-4335’;
```

ОПЕРАТОР LIKE

Использование подстановочных знаков (где Т — буква или фрагмент искомой строки)	**Результат** (не чувствительны к регистру)
'Т%'	Находит все записи, начинающиеся с буквы Т
'%Т'	Находит все записи, заканчивающиеся буквой Т
'%Т%'	Находит все записи, содержащие в строке букву Т
'Т%Т'	Находит все записи, начинающиеся и заканчивающиеся буквой Т

Преобразование типов данных

- CAST в PostgreSQL используется для преобразования из одного типа данных в другой.
- Сначала вы указываете выражение, которое может быть константой или столбцом таблицы, которое вы хотите преобразовать. Затем вы указываете целевой тип, в который вы хотите преобразовать.

- Синтаксис:
  CAST (выражение AS тип)

- Пример:
  SELECT CAST ('100' AS INTEGER);
  SELECT CAST (phone AS varchar (20))
  FROM Students;

Преобразование типов данных

- Помимо синтаксиса CAST, следующий синтаксис может использоваться для преобразования типа в другой:
- expression::type

- Обратите внимание, что синтаксис приведения с помощью :: специфичен для PostgreSQL и не соответствует SQL.

- Пример:
  SELECT '100'::INTEGER;

Книги

Коннолли, Томас М. Системы баз данных: практический подход к проектированию, внедрению и управлению / Томас М. Коннолли, Кэролин Э. Бегг.- Соединенные Штаты Америки: Pearson Education

Гарсия-Молина, Х. Система баз данных: Полная книга / Эктор Гарсия-Молина.- Соединенные Штаты Америки: Пирсон Прентис Холл

Шарма, Н. Основы баз данных: Книга для сообщества от сообщества / Нирадж Шарма, Ливиу Перниу.- Канада

www.postgresql.org/docs/manuals/

www.postgresql.org/docs/books/

Онлайн тренажеры по SQL

- sqlzoo.net
- sql-ex.ru
- sqlbolt.com
- pgexercises.com
-