

Управление данными

Лекция 4

Операторы и агрегатные функции

Свечников С.В.

Содержание занятия

01. Специальные операторы

02. Агрегатные функции

01. Специальные операторы

Специальные операторы SQL

В дополнение к логическим операторам и операторам отношения в SQL есть специальные операторы:

- ☐ IN
- ☐ BETWEEN
- ☐ LIKE
- ☐ IS NULL

Оператор IN

Оператор **IN** определяет множество, которому может принадлежать или не принадлежать значение.

Примеры

```
SELECT * FROM person  
WHERE city='Москва'  
OR city='Ростов';
```

```
SELECT * FROM person  
WHERE city IN ('Москва', 'Ростов');
```

Оператор BETWEEN

Оператор **BETWEEN** задает диапазон, в который должно попадать значение.

В отличие от IN, BETWEEN чувствителен к последовательности значений. Они должны располагаться в алфавитном или числовом порядке.

Пример

```
SELECT * FROM person  
WHERE year BETWEEN 22 AND 45;
```

BETWEEN является включающим оператором.

Оператор BETWEEN

В операторе BETWEEN важную роль играет последовательность указания границ.

Примеры

```
SELECT * FROM person  
WHERE year BETWEEN 22 AND 45;
```

Правильно

```
SELECT * FROM person  
WHERE year BETWEEN 45 AND 22;
```

Не правильно

BETWEEN может сравнивать символьные значения.

Пример

```
SELECT * FROM person  
WHERE city BETWEEN 'P' AND 'T';
```

Оператор LIKE

Оператор **LIKE** применяется для поиска подстрок в текстовых данных.

Для проверки текста с искомой строкой используются шаблоны – специальные символы, обозначающие любой текст.

Для оператора LIKE определены 2 шаблона:

- ❑ **_** (символ подчеркивания) – обозначает 1 любой символ
- ❑ **%** (символ процента) – обозначает любое количество символов

Примеры

```
SELECT * FROM person  
WHERE lname LIKE 'K%';
```

```
SELECT * FROM person  
WHERE lname LIKE '_o%В';
```


ESCAPE-символ

Если в строке необходимо найти знак «_» или «%», то необходимо определить **ESCAPE-символ**, который интерпретирует искомый знак как символ, а не шаблон.

Пример

```
SELECT * FROM person  
WHERE fname LIKE '%/_' ESCAPE '/';
```

Если нужно найти в столбце сам ESCAPE-символ, то его нужно ввести дважды.

Пример

```
SELECT * FROM person  
WHERE fname LIKE '%//_' ESCAPE '/';
```

Оператор IS NULL

Оператор **IS NULL** используется для поиска и обработки NULL-значений.

Пример

```
SELECT * FROM person  
WHERE phone IS NULL;
```

Специальным операторам может предшествовать логическое отрицание NOT.

Пример

```
SELECT * FROM person  
WHERE phone IS NOT NULL;
```

```
SELECT * FROM person  
WHERE NOT phone IS NULL;
```

Использование NOT в специальных операторах

NOT можно использовать с любым из специальных операторов.

Примеры

```
SELECT * FROM person  
WHERE city NOT IN ('Москва', 'Ростов');
```

```
SELECT * FROM person  
WHERE year NOT BETWEEN 22 AND 45;
```

```
SELECT * FROM person  
WHERE lname NOT LIKE 'K%';
```

02. Агрегатные функции

Вычисление итоговых значений

В запросе может выполняться обобщение как отдельных значений, так и групп. Для этого используются **агрегатные функции**, которые выводят единственное значение для группы.

Агрегатные функции:

- ☐ COUNT – определяет количество значений столбцов
- ☐ SUM – вычисляет арифметическую сумму значений
- ☐ AVG – выдает среднее значение для значений
- ☐ MAX – определяет максимальное значение
- ☐ MIN – определяет минимальное значение

SUM и AVG

Для функций **SUM** и **AVG** разрешается использовать только числовые столбцы.

Примеры

```
SELECT SUM(year) FROM person;
```

```
SELECT AVG(year) FROM person;
```

При использовании агрегатных функций возвращается единственное значение, независимо от количества строк в таблице.

COUNT

Функция **COUNT** подсчитывает количество значений в столбце.

Пример

```
SELECT COUNT(fname) FROM person;
```

- ❑ Для подсчета различных значений используется оператор DISTINCT.
- ❑ COUNT не учитывает NULL-значения.
- ❑ Для подсчета общего количества строк в таблице используется символ *.

Примеры

```
SELECT COUNT(DISTINCT city) FROM person;
```

```
SELECT COUNT(phone) FROM person;
```

```
SELECT COUNT(*) FROM person;
```

MAX и MIN

Функция **MAX** – определяет максимальное значение в столбце,
MIN – минимальное значение.

Пример

```
SELECT MIN(year), MAX(year) FROM person;
```


Использование выражений в агрегатах

Аргументы агрегатных функций могут состоять из скалярных выражений, включающих в себя несколько столбцов.

Примеры

```
SELECT AVG(year*12) FROM person;
```

```
SELECT AVG(year)*12 FROM person;
```

Группировка значений

Предложение **GROUP BY** позволяет выделить из отдельного столбца подмножество значений и применить к нему агрегатную функцию.

Примеры

```
SELECT city, COUNT(city) FROM person GROUP BY city;  
SELECT city, AVG(year) FROM person GROUP BY city;
```

Результатом применения агрегатной функции к группе также является единственное значение.

GROUP BY можно использовать одновременно с несколькими столбцами.

Пример

```
SELECT city, year, COUNT(year)  
FROM person  
GROUP BY city, year;
```

Предложение HAVING

Для фильтрации групп используется предложение **HAVING**. В нем задаются критерии, по которым из результата исключаются определенные группы.

Примеры

```
SELECT city, COUNT(city)
FROM person
GROUP BY city
HAVING COUNT(city)>1;
```

```
SELECT city, AVG(year)
FROM person
GROUP BY city
HAVING AVG(year) BETWEEN 30 AND 50;
```

Вложение агрегатных функций

Недопустимо использование агрегатных функций в качестве аргументов других функций.

Пример

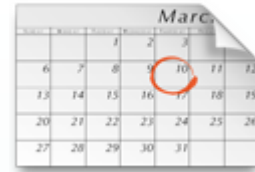
```
SELECT city, MAX(AVG(year))  
FROM person  
GROUP BY city;
```

Не правильно

Вопросы



Домашнее задание



- ☐ Изучить главы 7 и 8 книги М.Грабера Введение в SQL
- ☐ Подготовиться к контрольной работе по лекции
- ☐ Выполнить лабораторную работу №1
- ☐ Выполнить главу 1 курсового проекта «Анализ предметной области»
- ☐ Оставить отзывы о системе тестирования www.opentest.ru

Контроль



Для выполнения контрольного теста используем ссылку:

app.startexam.com/Center/Web/student

Короткая ссылка:

clck.ru/2Undm

Пароль на тест:

7625