

Лекция 4

Основы SQL (Structured Query Language), Базовые команды SQL, Ограничение и сортировка выходных данных. Функции

Цель

Освоить базовые конструкции языка SQL, научиться создавать запросы для выборки, фильтрации и сортировки данных, изучить основные встроенные функции СУБД.

Основные вопросы

1. Архитектура и компоненты SQL
2. Операторы выборки данных (SELECT)
3. Фильтрация и сортировка результатов
4. Встроенные функции SQL
5. Группировка данных и агрегатные функции

Лекция

Введение в SQL

SQL (Structured Query Language) - декларативный язык программирования, предназначенный для управления реляционными базами данных и работы с данными.

Историческая справка:

- 1970: Разработка языка SEQUEL в IBM
- 1974: Первая публикация Дональда Чемберлина
- 1986: Первый стандарт SQL-86 (ANSI)
- 1992: SQL-92 (major revision)
- Современные версии: SQL:1999, SQL:2003, SQL:2008, SQL:2011, SQL:2016

Особенности SQL:

- Декларативный характер (описывается ЧТО, а не КАК)
- Стандартизированный синтаксис
- Переносимость между СУБД
- Мощные возможности манипуляции данными

Компоненты SQL

Язык определения данных (DDL - Data Definition Language)

- CREATE - создание объектов
- ALTER - изменение объектов
- DROP - удаление объектов

Язык манипуляции данными (DML - Data Manipulation Language)

- SELECT - выборка данных
- INSERT - вставка данных
- UPDATE - обновление данных
- DELETE - удаление данных

Язык управления данными (DCL - Data Control Language)

- GRANT - предоставление прав
- REVOKE - отзыв прав

Язык управления транзакциями (TCL - Transaction Control Language)

- COMMIT - подтверждение транзакции
- ROLLBACK - откат транзакции
- SAVEPOINT - точка сохранения

Базовый синтаксис SELECT

Структура запроса SELECT:

SELECT список_столбцов

FROM список_таблиц

[**WHERE** условие_фильтрации]

[**GROUP BY** столбцы_группировки]

[**HAVING** условие_фильтрации_групп]

[**ORDER BY** столбцы_сортировки];

Простая выборка данных

Выбор всех столбцов:

```
SELECT * FROM Employees;
```

Выбор конкретных столбцов:

```
SELECT first_name, last_name, salary  
FROM Employees;
```

Использование псевдонимов (aliases):

```
SELECT  
    first_name AS Имя,  
    last_name AS Фамилия,  
    salary * 12 AS "Годовая зарплата"  
FROM Employees;
```

Фильтрация данных (WHERE)

Операторы сравнения:

- =, <>, != (не равно)
- <, >, <=, >=
- BETWEEN - диапазон значений
- IN - принадлежность множеству
- LIKE - сравнение с шаблоном
- IS NULL - проверка на NULL

Примеры фильтрации:

Простое условие:

```
SELECT * FROM Employees  
WHERE salary > 50000;
```

Диапазон значений:

```
SELECT * FROM Products
```

```
WHERE price BETWEEN 100 AND 500;
```

Принадлежность множеству:

```
SELECT * FROM Customers
```

```
WHERE country IN ('USA', 'Canada', 'UK');
```

Поиск по шаблону:

```
SELECT * FROM Employees
```

```
WHERE last_name LIKE 'Sm%'; -- начинается с Sm
```

```
SELECT * FROM Products
```

```
WHERE product_name LIKE '%computer%'; -- содержит computer
```

Проверка на NULL:

```
SELECT * FROM Orders
```

```
WHERE ship_date IS NULL;
```

Логические операторы:

```
SELECT * FROM Employees
```

```
WHERE salary > 50000
```

```
AND department = 'IT'
```

```
AND hire_date > '2020-01-01';
```

Сортировка результатов (ORDER BY)

Синтаксис:

```
ORDER BY столбец1 [ASC|DESC], столбец2 [ASC|DESC], ...
```

Примеры сортировки:

Простая сортировка:

```
SELECT * FROM Employees  
ORDER BY last_name ASC;
```

Сортировка по нескольким столбцам:

```
SELECT * FROM Employees  
ORDER BY department ASC, salary DESC;
```

Сортировка по выражению:

```
SELECT *, salary * 12 AS annual_salary  
FROM Employees  
ORDER BY annual_salary DESC;
```

Сортировка по номеру столбца:

```
SELECT first_name, last_name, salary  
FROM Employees  
ORDER BY 3 DESC; -- сортировка по salary (3-й столбец)
```

Встроенные функции SQL

Строковые функции

CONCAT - объединение строк:

```
SELECT CONCAT(first_name, ' ', last_name) AS full_name  
FROM Employees;
```

SUBSTRING - извлечение подстроки:

```
SELECT SUBSTRING(phone_number, 1, 3) AS area_code  
FROM Customers;
```

UPPER/LOWER - регистр символов:

```
SELECT UPPER(last_name) AS last_name_upper
```

```
FROM Employees;
```

LENGTH - длина строки:

```
SELECT product_name, LENGTH(product_name) AS name_length
```

```
FROM Products;
```

TRIM - удаление пробелов:

```
SELECT TRIM(' Hello World ') AS trimmed_string;
```

Числовые функции

ABS - абсолютное значение:

```
SELECT ABS(-10) AS absolute_value;
```

ROUND - округление:

```
SELECT ROUND(price, 2) AS rounded_price
```

```
FROM Products;
```

CEILING/FLOOR - округление вверх/вниз:

```
SELECT
```

```
    CEILING(4.3) AS ceil_value, -- 5
```

```
    FLOOR(4.7) AS floor_value; -- 4
```

POWER - возведение в степень:

```
SELECT POWER(2, 3) AS power_result; -- 8
```

Функции даты и времени

CURRENT_DATE/TIME - текущая дата/время:

```
SELECT
  CURRENT_DATE AS today,
  CURRENT_TIME AS now;
```

EXTRACT - извлечение компонентов даты:

```
SELECT
  EXTRACT(YEAR FROM hire_date) AS hire_year,
  EXTRACT(MONTH FROM hire_date) AS hire_month
FROM Employees;
```

DATE_ADD/DATE_SUB - добавление/вычитание интервалов:

```
SELECT
  order_date,
  DATE_ADD(order_date, INTERVAL 7 DAY) AS delivery_date
FROM Orders;
```

DATEDIFF - разница между датами:

```
SELECT
  DATEDIFF(CURRENT_DATE, hire_date) AS days_employed
FROM Employees;
```

Агрегатные функции и группировка

Основные агрегатные функции

COUNT - подсчет строк:

```
SELECT COUNT(*) AS total_employees FROM Employees;
```

SUM - сумма значений:

```
SELECT SUM(salary) AS total_salary FROM Employees;
```

AVG - среднее значение:

```
SELECT AVG(salary) AS average_salary FROM Employees;
```

MAX/MIN - максимальное/минимальное значение:

SELECT

```
MAX(salary) AS max_salary,  
MIN(salary) AS min_salary
```

FROM Employees;

Группировка данных (GROUP BY)

Группировка по одному столбцу:

SELECT

```
department,  
COUNT(*) AS employee_count,  
AVG(salary) AS avg_salary  
FROM Employees
```

GROUP BY department;

Группировка по нескольким столбцам:

SELECT

```
department,  
job_title,  
COUNT(*) AS employee_count  
FROM Employees
```

GROUP BY department, job_title;

Фильтрация групп (HAVING)

HAVING vs WHERE:

- WHERE фильтрует строки до группировки
- HAVING фильтрует группы после группировки

Пример использования HAVING:

SELECT

```
department,  
AVG(salary) AS avg_salary
```



```
FROM Employees
GROUP BY department

HAVING AVG(salary) > 50000;
```

Практические примеры комплексных запросов

Пример 1: Отчет по отделам

```
SELECT
    department AS Отдел,
    COUNT(*) AS "Количество сотрудников",
    ROUND(AVG(salary), 2) AS "Средняя зарплата",
    MAX(salary) AS "Максимальная зарплата",
    MIN(salary) AS "Минимальная зарплата"
FROM Employees
WHERE hire_date > '2020-01-01'
GROUP BY department
HAVING COUNT(*) > 5

ORDER BY AVG(salary) DESC;
```

Пример 2: Анализ продаж

```
SELECT
    EXTRACT(YEAR FROM order_date) AS year,
    EXTRACT(MONTH FROM order_date) AS month,
    COUNT(*) AS order_count,
    SUM(total_amount) AS total_sales,
    AVG(total_amount) AS average_order_value
FROM Orders
WHERE order_date BETWEEN '2023-01-01' AND '2023-12-31'
GROUP BY EXTRACT(YEAR FROM order_date), EXTRACT(MONTH FROM
order_date)

ORDER BY year, month;
```

Контрольные вопросы

1. Какие основные компоненты включает язык SQL?

2. Как работает оператор SELECT? Опишите его базовый синтаксис.
3. Какие операторы фильтрации доступны в конструкции WHERE?
4. В чем разница между операторами WHERE и HAVING?
5. Перечислите основные агрегатные функции и приведите примеры их использования.
6. Как выполняется сортировка результатов запроса? Какие варианты сортировки доступны?

Литература

1. Дейт К. Дж. Введение в системы баз данных. - Глава 4.
2. Коннолли Т., Бегг К. Базы данных: проектирование, реализация и сопровождение. - Глава 5-6.
3. Кузнецов С.Д. Основы баз данных. - Глава 5.
4. Форта Б. SQL в примерах и задачах. - Глава 1-4.