

---



# ВЫБОРКА ДАННЫХ

Выборка строк

# Команды SQL

---

- Команда может занимать одну или несколько строк.
- Для удобства чтения команды можно использовать табуляцию и отступы.
- Сокращение и перенос слов запрещены.
- Символы верхнего и нижнего регистров не различаются.
- Команды вводятся в буфер SQL.

# Основной блок запроса

```
SELECT [DISTINCT]      {*,column  
[alias],...}  
FROM table;
```

- **SELECT** задает *столбцы*, подлежащие выборке.
- **FROM** указывает, *из какой таблицы*

Простейший оператор **SELECT** содержит два предложения:

- Предложение **SELECT**
  - Звездочка (\*) обозначает все столбцы
- Предложение **FROM**

```
SQL> SELECT  *  
2 FROM      s_dept;
```

# Выборка всех столбцов и всех строк

```
SQL> SELECT *  
2 FROM s_dept;
```

ID	NAME	REGION_ID
10	Finance	1
31	Sales	1
32	Sales	2
33	Sales	3
34	Sales	4
35	Sales	5
41	Operations	1
42	Operations	2
43	Operations	3
44	Operations	4
45	Operations	5
50	Administration	1

12 rows selected.

# Выборка заданных столбцов

```
SQL> SELECT dept_id, last_name, manager_id  
2 FROM s_emp;
```

- Перечислить столбцы в предложении **SELECT**.
- Разделить столбцы в списке запятыми.
- Указать столбцы в порядке, в котором они должны появиться на выводе.

## МЕТКИ СТОЛБЦОВ ПО УМОЛЧАНИЮ

- Выравнивание метки по умолчанию:
  - Слева: даты и символьные данные
  - Справа: числовые данные
- По умолчанию вывод меток производится в символах верхнего регистра.

# Арифметические выражения

Создание выражений для типов данных **NUMBER** и **DATE** с помощью арифметических операторов.

Сложение +

Вычитание -

Умножение \*

Деление /

```
SQL> SELECT last_name, salary * 12, commission_pct  
2 FROM s_emp;
```

```
LAST_NAME SALARY*12 COMMISSION_PCT  
-----
```

```
...
```

```
Havel 15684
```

```
Magee 16800 10
```

```
Giljum 17880 12.5
```

```
Sedeghi 18180 10
```

```
Nguyen 18300 15
```

```
Dumas 17400 17.5
```

```
Maduro 16800
```

```
...
```

# Порядок выполнения операторов

- Умножение и деление выполняются до сложения и вычитания.
- Операторы, имеющие один и тот же приоритет, выполняются по очереди слева направо.
- Для изменения порядка вычислений и удобства чтения выражений можно использовать скобки.

Скобки используются для изменения порядка выполнения действий при вычислении выражения.

```
SQL> SELECT last_name, salary, 12 * salary + 100
2 FROM s_emp;
... Velasquez 2500 30100
```

```
SQL> SELECT last_name, salary, 12 * (salary + 100)
2 FROM s_emp;
... Velasquez 2500 31200
```

# Псевдонимы столбцов

---

**Псевдоним столбца заменяет его заголовок.**

- **Особенно полезен при расчетах.**
- **Следует сразу за заголовком столбца.**
  - **Между заголовком и псевдонимом столбца может находиться необязательное ключевое слово AS.**
- **Если псевдоним содержит пробелы или специальные символы или если в нем различаются символы верхнего и нижнего регистров, двойные кавычки обязательны.**



# Оператор конкатенации

## Оператор конкатенации:

- Обозначается двойной вертикальной чертой (||).
- Соединяет столбцы или текстовые строки с другими столбцами.
- Создает столбец, являющийся символьным выражением.

```
SQL> SELECT  first_name||last_name  
2  FROM      s_emp;
```

```
Employees  
-----
```

```
CarmenVelasquez
```

```
LaDorisNgao
```

```
MidoriNagayama
```

```
MarkQuick-To-See
```

```
AudryRopeburn
```

```
MollyUrguhart
```

```
...
```

# Строка символов - литерал

- **Литерал — это строка символов, выражение или число, включенные в список SELECT.**
- **Символьные литералы и литералы-даты должны быть заключены в апострофы.**
- **Каждая строка символов выводится по одному разу для каждой возвращаемой строки таблицы.**

```
SQL> SELECT  first_name || ' ' || last_name  
2 || ', ' || title "Employees"  
3 FROM      s_emp;
```

# Обработка неопределенных значений

- Неопределенным значением (NULL) называется недоступное, неприсвоенное, неизвестное или неприменимое значение.
- Неопределенное значение отличается от нуля и пробела.
- Результатом арифметического выражения, содержащего неопределенное значение, также является неопределенное значение.

```
SQL> SELECT  last_name, title,  
2 salary*commission_pct/100 COMM  
3 FROM      s_emp;
```

# Функция NVL

Преобразование **NULL** в фактическое значение с помощью функции **NVL**.

- **Используемые типы данных:** дата, символьные и числовые.
- **Типы данных должны совпадать:**
  - **NVL (start\_date, '01-JAN-95')**
  - **NVL (title, 'No Title Yet')**
  - **NVL (salary, 1000)**

```
SQL> SELECT  last_name, title,  
2 salary*NVL(commission_pct,0)/100 COMM  
3 FROM      s_emp;
```

# Дубликаты строк

- По умолчанию результат запроса включает все строки - в том числе и дубликаты.

```
SQL> SELECT name  
2 FROM s_dept;
```

Предотвратить вывод дубликатов можно с помощью ключевого слова **DISTINCT** в предложении **SELECT**..

```
SQL> SELECT DISTINCT name  
2 FROM s_dept;
```

- **DISTINCT** относится ко всем столбцам в списке **SELECT**.

```
SQL> SELECT DISTINCT dept_id, title  
2 FROM s_emp;
```

Результат применения **DISTINCT** к нескольким столбцам - вывод строк с неповторяющимися сочетаниями значений этих столбцов.