


ФУНКЦИИ SQL

Групповые функции

Групповые функции

- Групповые функции работают с множествами строк и возвращают один результат на группу.
- Групповые функции могут быть заданы в списках `SELECT` и предложениях `HAVING`.
- Предложение `GROUP BY` в команде `SELECT` разбивает множество строк на группы.
- Предложение `HAVING` исключает из результата некоторые группы.

Предложения GROUP BY и HAVING

```
SELECT column, group_function
FROM   table
[WHERE condition]
[GROUP BY group_by_expression]
[HAVING group_condition]
[ORDER BY column];
```

- Предложение **GROUP BY** делит строки на группы.
- Предложение **HAVING** исключает из рассмотрения некоторые группы.

Групповые функции

- **AVG (DISTINCT|ALL|*n*)**
- **COUNT (DISTINCT|ALL|*expr**)**
- **MAX (DISTINCT|ALL|*expr*)**
- **MIN (DISTINCT|ALL|*expr*)**
- **STDDEV (DISTINCT|ALL|*n*)**
- **SUM (DISTINCT|ALL|*n*)**
- **VARIANCE (DISTINCT|ALL|*n*)**

Групповые функции: пример

Функции AVG и SUM применяются к столбцам с числовыми данными.

```
SQL> SELECT  AVG(salary), MAX(salary),  
2  MIN(salary), SUM(salary)  
3  FROM s_emp  
4  WHERE  UPPER(title) LIKE 'SALES%';
```

Функции MAX и MIN применяются к данным любого типа.

```
SQL> SELECT  MIN(last_name), MAX(last_name)  
2  FROM s_emp;
```

Функция COUNT: примеры

COUNT(*) возвращает количество строк в таблице.

```
SQL> SELECT  COUNT (*)
2  FROM s_emp
3  WHERE    dept_id = 31;
```

COUNT(*expr*) возвращает количество строк с определенными значениями (не NULL).

```
SQL> SELECT  COUNT (commission_pct)
2  FROM s_emp
3  WHERE    dept_id = 31;
```

Предложение GROUP BY

```
SELECT column, group_function
FROM   table
[WHERE condition]
[GROUP BY group_by_expression]
[ORDER BY column];
```

- Предложение **GROUP BY** разбивает строки таблицы на группы.
- Если в предложении **SELECT** заданы столбцы, их список должен использоваться и в предложении **GROUP BY**.
- С помощью предложения **ORDER BY** можно изменить порядок сортировки, используемый по умолчанию.

Без предложения GROUP BY

```
SQL> SELECT  id, last_name, dept_id DEPARTMENT
2  FROM      s_emp
3  WHERE     dept_id = 41;
```

```
ID  LAST_NAME DEPARTMENT
--  -
2   Ngao      41
6   Urguhart  41
16  Maduro    41
17  Smith     41
```

← Номер 41
← повторяется четыре
← раза, т.к. является
← номером отдела для
← четырех служащих.

С предложением GROUP BY

```
SQL> SELECT dept_id, COUNT(*) "Number"  
2 FROM s_emp  
3 WHERE dept_id = 41  
4 GROUP BY dept_id;
```

DEPT_ID	Number
41	4

Благодаря предложению **GROUP BY** на каждый отдел, заданный в предложении **WHERE**, выводится одна строка, а функция **COUNT(*)** возвращает количество служащих в каждом выбранном отделе (группе).

Предложение GROUP BY: примеры

Количество клиентов в каждой категории по кредитному рейтингу.

```
SQL> SELECT  credit_rating, COUNT(*) "# Cust"
2  FROM s_customer
3  GROUP BY credit_rating;
```

Должности и месячная заработная плата для каждой должности.

```
SQL> SELECT  title, SUM(salary) PAYROLL
2  FROM s_emp
3  WHERE     title NOT LIKE 'VP%'
4  GROUP BY title
5  ORDER BY SUM(salary);
```

Предложение GROUP BY

- Все столбцы из списка **SELECT**, не входящие в групповые функции, должны быть включены в предложение **GROUP BY**.
- Столбец, заданный в предложении **GROUP BY**, не обязательно должен быть задан в предложении **SELECT**.
- Если столбец из предложения **GROUP BY** входит в список **SELECT**, результат имеет больше смысла.

```
SQL> SELECT  title, MAX(salary)
2  FROM s_emp
3  GROUP BY title;
```

Недействительные запросы

- Если предложение **GROUP BY** отсутствует или неправильно, выдается сообщение об ошибке.
- Все столбцы или выражения из списка **SELECT**, не являющиеся групповой функцией, должны быть включены в предложение **GROUP BY**.

```
SQL> SELECT  region_id, COUNT(name)
          2  FROM s_dept;
SELECT region_id, COUNT(name)
          *
ERROR at line 1:
ORA-00937: not a single-group group function
```

Недействительные запросы

- Предложение **WHERE** для исключения групп не используется.
- Для исключения некоторых групп следует пользоваться предложением **HAVING**.

```
SQL> SELECT  dept_id, AVG(salary)
 2  FROM s_emp
 3  WHERE    AVG(salary) > 2000
 4  GROUP BY dept_id;
WHERE AVG(salary) > 2000
      *
ERROR at line 3:
ORA-00934: group function is not allowed here
```

Группы внутри групп

- Для получения сводных результатов по нескольким группам и подгруппам можно указать в предложении **GROUP BY** более одного столбца.
- Порядок сортировки, используемый по умолчанию, определяется порядком столбцов в предложении **GROUP BY**.

```
SQL> SELECT  dept_id, title, COUNT(*)  
2  FROM s_emp  
3  GROUP BY dept_id, title;
```

Вывод конкретных строк с помощью предложения WHERE

```
SQL> SELECT last_name, title
2 FROM s_emp
3 WHERE last_name LIKE 'V%';
```

ĩđăăëîæáíèà
 WHERE
 (îãđàíè÷èâààò
 ÷èñëî âûáèđàáîûõ
 ñòđîê)

```
LAST_NAME TITLE
-----
Velasquez President
```

Âûâîä äàííûõ î
 êíîđàòĩĩ
 ñëóæàùàî â
 ñĩĩòââòñòâèè ñ
 êđèòâđèÿìè â
 ĩđăăëîæáíèè
 WHERE.

Вывод конкретных групп с помощью предложения HAVING

```
SQL> COLUMN "ANNUAL SALARY" FORMAT $99,999.99
SQL> SELECT title, 12 * AVG(salary) "ANNUAL SALARY",
2 COUNT(*) "NUMBER OF EMPLOYEES"
3 FROM s_emp
4 GROUP BY title
5 HAVING COUNT(*) > 2;
```

Предложение HAVING (исключение групп)

TITLE	ANNUAL SALARY	NUMBER OF EMPLOYEES
Sales Representative	\$17,712.00	5
Stock Clerk	\$11,388.00	10
Warehouse Manager	\$14,776.80	5

Вывод групп по должностям в соответствии с ограничениями в предложении HAVING.

Предложение HAVING

```
SELECT column, group_function
FROM   table
[WHERE condition]
[GROUP BY group_by_expression]
[HAVING group_condition]
[ORDER BY column];
```

Предложение **HAVING** используется для дальнейшего ограничения количества групп.

Шаг 1: Группирование строк.

Шаг 2: Применение групповых функций к группам.

Шаг 3: Вывод групп, удовлетворяющих условию предложения **HAVING**.

Предложение HAVING: пример

Группа "President" в выходных данных отсутствует, т.к. не удовлетворяет заданному критерию.

```
SQL> SELECT  title, SUM(salary) PAYROLL
2  FROM s_emp
3  WHERE    title NOT LIKE 'VP%'
4  GROUP BY title
5  HAVING   SUM(salary) > 5000
6  ORDER BY SUM(salary);
```

Предложение HAVING: пример

- Предложение **GROUP BY** можно использовать без указания групповой функции в списке **SELECT**.
- Если отбор строк производится по результатам групповой функции, то использование как предложения **GROUP BY**, так и предложения **HAVING** обязательно.

```
SQL> SELECT dept_id
2 FROM s_emp
3 GROUP BY dept_id
4 HAVING SUM(salary) > 4000;
```

Заключение

- Имеется семь групповых функций: **AVG**, **COUNT**, **MAX**, **MIN**, **STDDEV**, **SUM**, **VARIANCE**.
- С помощью предложения **GROUP BY** создаются группы.
- Некоторые группы исключаются с помощью предложения **HAVING**.

```
SELECT column, group_function
FROM   table
[WHERE condition]
[GROUP BY group_by_expression]
[HAVING group_condition]
[ORDER BY column];
```

Обзор практического занятия

- Демонстрация запросов с использованием всех групповых функций, кроме `STDDEV` и `VARIANCE`.
- Разбиение строк на группы для получения более, чем одного результата.
- Исключение групп с помощью предложения `HAVING`.

Задания для практического занятия

1. Групповые функции обрабатывают большое количество строк для получения одного результата? (Да/Нет)
2. Во время выполнения групповых функций учитываются неопределенные значения? (Да/Нет)
3. Предложение HAVING используется для исключения строк из выборки для группы? (Да/Нет)
4. Вывести наибольшую и наименьшую сумму заказа из таблицы S_ORD.
5. Вывести минимальную и максимальную заработную плату по всем должностям в алфавитном порядке.
6. Вывести номер каждого менеджера и заработную плату самого низкооплачиваемого из его подчиненных. Исключить группы с заработной платой менее 1000. Отсортировать по размеру заработной платы.
7. Вывести наименование каждого клиента и количество сделанных им заказов.
8. Получить список номеров и названий всех регионов с указанием количества отделов в каждом.