



ГРУППОВЫЕ ФУНКЦИИ

Цели занятия

- Изучение групповых функций.
- Изучение предложения GROUP BY.
- Изучение предложения HAVING.

Синтаксис

***SELECT** групповая_функция ([DISTINCT] | * | столбец),
столбец
FROM таблица
[WHERE условие]
[GROUP BY выражение_группировки_данных]
[HAVING условие_группы]
[ORDER BY столбец];*

Где: выражение_группировки_данных – задает столбцы, на основе значений которых группируются строки.

Условие_группы – включает в выходной результат только те группы, для которых истинно заданное условие.

Групповые функции

В отличие от однострочных функций, групповые функции работают над множеством строк и выдают один результат на группу. Групповые функции используются и в списке SELECT и в предложении HAVING.

Функция	Назначение
AVG (DISTINCT ALL n)	Среднее значение <i>n</i> без учета неопределенных значений.
COUNT (DISTINCT ALL выражение *)	Количество строк без учета неопределенных значений <i>выражения</i> . По “*” подсчитываются все строки, включая повторяющиеся и строки с неопределенными значениями.
MAX (DISTINCT ALL выражение)	Максимальное значения <i>выражения</i> .
MIN (DISTINCT ALL выражение)	Минимальное значения <i>выражения</i> .
STDDEV (DISTINCT ALL n)	Стандартное отклонение <i>n</i> без учета неопределенных значений.
SUM (DISTINCT ALL n)	Сумма значений <i>n</i> без учета неопределенных значений.
VARIANCE (DISTINCT ALL n)	Дисперсия <i>n</i> без учета неопределенных значений.

Особенности групповых функций

1. По умолчанию все строки таблицы рассматриваются как одна группа. Для разбиения таблицы на меньшие группы используется предложение GROUP BY команды SELECT. Для отбора возвращаемых групп используется предложение HAVING.
2. Каждая функция допускает один аргумент. Если задан оператор DISTINCT, функция учитывает лишь неповторяющиеся значения, а при наличии оператора ALL рассматриваются все значения, включая повторяющиеся. Вариант ALL установлен по умолчанию, поэтому задавать его нет необходимости.
3. Если задано *выражение*, допустимыми типами данных для аргументов функций COUNT, MAX, MIN являются символьные (CHAR, VARCHAR2), числовые (NUMBER) и даты (DATE). Функции AVG, SUM, STDDEV, VARIANCE обрабатывают только числовые аргументы.
4. Все групповые функции, кроме COUNT(*), игнорируют неопределенные значения. Для подстановки нового значения вместо неопределенного используется функция NVL.

Групповые функции

Пример №1: вывод средней, самой высокой и самой низкой заработной платы, а также суммы заработной платы всех торговых представителей за месяц.

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

Пример №2: вывод первой и последней фамилий из алфавитного списка всех служащих.

```
SELECT MIN(last_name), MAX(last_name)  
FROM   s_emp;
```

Функция COUNT

Функция COUNT имеет два формата: COUNT(*) и COUNT(*выражение*). Функция COUNT(*) возвращает количество всех строк в таблице, включая повторяющиеся строки и строки с неопределенными значениями.

Пример №1: вывод количества всех сотрудников из 31 отдела.

```
SELECT COUNT(*)  
FROM s_emp WHERE dept_id = 31;
```

Пример №2: вывод количества сотрудников из 31 отдела, получающих комиссионные.

```
SELECT COUNT(commission_pct)  
FROM s_emp WHERE dept_id = 31;
```

Предложение GROUP BY

1. Если в предложение SELECT включена групповая функция, получить одновременно отдельные результаты по разным значениям можно только в том случае, если в предложении GROUP BY задан список столбцов группировки. Если список столбцов отсутствует, выдается сообщение об ошибке.
2. Предложение WHERE позволяет исключить некоторые строки до начала разбиения на группы.
3. Предложение GROUP BY используется для разбиения строк таблицы на группы и для получения сводной информации по каждой группе с использованием групповых функций.
4. В предложении GROUP BY нельзя использовать позиционные обозначения или псевдонимы столбцов.
5. По умолчанию строки сортируются в порядке возрастания в соответствии со списком GROUP BY. Изменить порядок сортировки можно с помощью предложения ORDER BY.
6. Столбец, заданный в предложении GROUP BY, не обязательно должен присутствовать в предложении SELECT, но его присутствие делает результаты более осмысленными.

Предложение GROUP BY

Пример №1: вывод всех возможных кредитных рейтингов и количества клиентов в каждой категории под заголовком “Customers”.

```
SELECT    credit_rating, COUNT(*) AS “Customers”  
FROM      s_customer  
GROUP BY credit_rating;
```

Пример №2: вывод максимальной заработной платы для каждой должности с указанием должности.

```
SELECT    title, MAX(salary)  
FROM      s_emp  
GROUP BY title;
```

Группировка внутри групп

Сводные результаты по подгруппам в группе можно получить путем указания более чем одного столбца в предложении GROUP BY.

Пример: вывод количества служащих по должностям внутри отделов.

```
SELECT dept_id, title, COUNT (*)  
FROM s_emp  
GROUP BY dept_id, title;
```

Предложение HAVING

1. Предложение HAVING задает условие отбора групп для вывода данных, т.е. на группы накладывается дальнейшее ограничение, основанное уже на обобщенной информации.
2. Образование групп и вычисление групповых функций происходят до того, как к группам из списка SELECT применяется ограничение, заданное в предложении HAVING.

Предложение HAVING

Пример №1: вывод номеров отделов и средней заработной платы для отделов, в которых средняя заработная плата более 2000.

```
SELECT dept_id, AVG(salary)
FROM   s_emp
GROUP BY dept_id
HAVING AVG(salary) > 2000;
```

Пример №2: вывод всех должностей, кроме вице-президентов, и соответствующей общей заработной платы за месяц. Список сортируется по общей заработной плате, сумма которой за месяц по должности превышает 5000.

```
SELECT title, SUM(salary) AS PAYROLL
FROM   s_emp
WHERE  title NOT LIKE 'VP%'
GROUP BY title
HAVING SUM(salary) > 5000
ORDER BY SUM(salary) ;
```