

Лабораторная работа 3

Агрегатные функции и группировка данных

Агрегатные функции обрабатывают группу строк и возвращают один результат для группы. Группа строк может включать как всю таблицу, так и часть таблицы. В таблице 1 содержится описание основных агрегатных функций

Таблица 1. Основные агрегатные функции

Функция	Возвращает	Тип аргумента
SUM(expr)	Сумму значений expr, игнорируя значения NULL	Число
COUNT(expr *})	Число строк, игнорируя значения NULL. При использовании в качестве аргумента * число строк.	Любой
MAX(expr)	Максимальное значение expr, игнорируя значения NULL	Число, строка, дата
MIN(expr)	Минимальное значение expr, игнорируя значения NULL	Число, строка, дата
AVG(expr)	Среднее значение expr , игнорируя значения NULL	Число

Синтаксис агрегатных функций:

{имя функции} ({expr})

Где:

expr – аргумент агрегатной функции, который может содержать следующие элементы :

[DISTINCT] {имя столбца} | {выражение} | {однострочная функция}

Запрос 1. Вывести обобщенные данные о зарплате сотрудников

```
SELECT MIN(salary) AS "MIN", MAX(salary) AS "MAX",
ROUND(AVG(salary),2) AS "AVG",
SUM(salary) As "SUM"
FROM Employees
```

MIN	MAX	AVG	SUM
2200.00	24000.00	6479.05	680300.00

Запрос 2. Пример использования функции COUNT()

```
SELECT COUNT(*) as "COUNT(*)", COUNT(salary) as "COUNT(salary)",
COUNT(DISTINCT salary) as "COUNT(DISTINCT salary)"
FROM Employees
```

COUNT(*)	COUNT(salary)	COUNT(DISTINCT salary)
107	105	56

Группировка

Чаще всего агрегатные функции используются в запросах с группировкой. В общем виде запрос с группировкой может быть представлен в следующем виде:

```
SELECT {список столбцов*}, {агрегатные функции}
FROM {таблица}
WHERE {условия}
GROUP BY {список столбцов*}
HAVING {условия на группу};
```

Списки столбцов в предложениях SELECT и GROUP BY должны совпадать.

Предложение GROUP BY разбивает данные на группы, и запрос выводит обобщенные данные о каждой группе.

Запрос 3. Для каждого отдела вывести количество и суммарную зарплату сотрудников.

```
SELECT department_id, COUNT(*), SUM(salary)
FROM Employees
GROUP BY department_id
ORDER BY department_id
```

department_id	count	sum
10	1	4400.00
20	2	19000.00
30	6	24900.00
40	1	6500.00
50	45	154300.00

Группировка по нескольким столбцам

В предложении GROUP BY можно указать несколько столбцов. В этом случае группу образуют строки с совпадающими значениями всех столбцов, по которым осуществляется группировка. Рассмотрим задачи, в которых требуется группировка по нескольким столбцам.

Запрос 4. Для сотрудников, работающих в отделах 30 и 50, рейтинг которых >2, вывести код должности (**job_id**), рейтинг (**rating_e**) и количество сотрудников, которые имеют одинаковые пары значений **job_id**, **rating_e**.

```
SELECT job_id, rating_e, count(*)
FROM Employees
WHERE department_id in (30,50)
and rating_e>2
GROUP BY department_id, job_id, rating_e
ORDER BY job_id, rating_e
```

job_id	rating_e	count
	4	1
PU_CLERK	3	4
SH_CLERK	3	1
SH_CLERK	4	6
SH_CLERK	5	5
ST_CLERK	3	6
ST_CLERK	4	2
ST_CLERK	5	4
ST_MAN	4	1

Использование условий на группу

В запросах с группировкой можно использовать предложение **HAVING**, которое содержит условия на группу. Результат запроса будет содержать данные только о тех группах, которые удовлетворяют этим условиям.

Запрос 5. Вывести суммарную зарплату для отделов, у которых суммарная зарплата превышает 50000.

```
SELECT department_id, SUM(salary)
FROM Employees
GROUP BY department_id
HAVING SUM(salary) > 50000
ORDER BY department_id
```

department_id	sum
50	154300.00
80	295500.00
90	58000.00
100	51600.00

Использование выражения LIMIT

Это выражение позволяет ограничить количество строк в результате выполнения запроса. Если запрос содержит выражение **LIMIT N**, то будет выводиться не более первых **N** строк результата.

Запрос 6. Вывести номера (product_id) 5-ти товаров с наибольшим количеством во всех продажах.

```
SELECT product_id, SUM(quantity) AS sum_quantity
FROM Order_items
GROUP BY product_id
ORDER BY sum_quantity desc
LIMIT 5;
```

product_id	sum_quantity
78	458
52	455
19	444
28	411
18	392

Использование специальных операторов группировки

Рассмотрим специальные операторы группировки и функции, которые позволяют существенно расширить возможности запросов, в которых используется группировка данных.

Оператор GROUP BY ROLLUP

Расширяет возможности GROUP BY, возвращая для каждой группы строку содержащую итоги по группе, а также строку содержащей общий итог для всех групп, и имеет следующий вид:

GROUP BY ROLLUP {список столбцов}

Запрос 7. Для сотрудников, работающих в отделах 30 и 50, рейтинг которых >2, вывести код должности и количество занимающих каждую должность

```
SELECT department_id, job_id, count(*)
FROM Employees
WHERE department_id IN (30,50)
and rating_e>2
GROUP BY ROLLUP (department_id, job_id)
ORDER BY department_id, job_id
```

department_id	job_id	count
30	PU_CLERK	4
30		4
50		1
50	SH_CLERK	12
50	ST_CLERK	12
50	ST_MAN	1
50		26
		30

Оператор GROUP BY CUBE

Возвращает предварительные итоги для всех комбинаций столбцов, и строку с общим итогом, и имеет следующий вид:

GROUP BY CUBE {список столбцов}

Запрос 8. Используя **GROUP BY CUBE**, вывести для сотрудников, работающих в отделах 30 и 50, рейтинг которых >2, код должности и количество занимающих каждую должность

```
SELECT department_id, job_id, count(*)
FROM Employees
WHERE department_id IN (30,50)
AND rating_e>2
GROUP BY CUBE (department_id, job_id)
ORDER BY department_id, job_id
```

department_id	job_id	count
30	PU_CLERK	4
30		4
50		1
50	SH_CLERK	12
50	ST_CLERK	12
50	ST_MAN	1
50		26
		1
	PU_CLERK	4
	SH_CLERK	12
	ST_CLERK	12
	ST_MAN	1
		30

Оператор GROUP BY CUBE выводит очень много строк, поэтому в ряде случаев удобнее использовать оператор GROUP BY GROUPING SETS.

Оператор GROUP BY GROUPING SETS

Используется вместо оператора GROUP BY CUBE, в тех случаях, когда нужно вывести только строки с промежуточными итогами.

Запрос 9. Для сотрудников, работающих в отделах 30 и 50, рейтинг которых >2, вывести количество сотрудников в отделе и количество сотрудников занимающих каждую должность

```
SELECT department_id, job_id, count(*)
FROM Employees
WHERE department_id IN (30,50)
AND rating_e>2
GROUP BY GROUPING SETS(department_id, job_id)
ORDER BY department_id,job_id
```

department_id	job_id	count
30		4
50		26
		1
	PU_CLERK	4
	SH_CLERK	12
	ST_CLERK	12
	ST_MAN	1

Запрос 10. Для сотрудников, работающих в отделах 30,50 и имеющих rating_e >2, вывести общее количество таких сотрудников в каждом отделе, и количество сотрудников, которые имеют одинаковые пары значений (job_id, rating_e).

```
SELECT department_id, job_id,rating_e, count(*)
FROM Employees
WHERE department_id IN (30,50)
and rating_e >2
GROUP BY GROUPING SETS(department_id, (job_id, rating_e))
ORDER BY department_id
```

department_id	job_id	rating_e	count
30			4
50			26
	SH_CLERK	3	1
	ST_CLERK	3	6
	ST_CLERK	5	4
	PU_CLERK	3	4
		4	1
	SH_CLERK	5	5
	SH_CLERK	4	6
	ST_CLERK	4	2
	ST_MAN	4	1

Задание

Задача 1. Найти количество товаров, в названии которых есть слово CORE.

Задача 2. Вывести количество заказов, которые клиент 46 оформил в течение каждого года

Задача 3. Вывести количество заказов, оформленных за каждый месяц 2019 года.

Задача 4. Определить номера товаров, по которым было совершено меньше 10 продаж. Продажа это строка в таблице Order_items.

Задача 5. Для заказов вывести номера товаров в заказе, их количество, общую стоимость каждого товара и всего заказа. Вывести эти данные только для заказов, у которых order_id <30.

Задача 6. Вывести количество заказов, оформленных в течение каждого года и количество заказов, которые оформил каждый клиент. Вывести только те строки, в которых количество заказов >4.

Задача 7. Для каждого отдела вывести суммарную зарплату сотрудников, за весь период их работы.