

УСР 1.1 Расширения SQL Server для группировки (ROLLUP, CUBE, GROUPING SETS, OVER).

I. Теоретическая часть

Operator ROLLUP создаёт группу для каждого сочетания выражений столбцов. Кроме того, выполняет сведение результатов в промежуточные и общие итоги.

Для этого запрос перемещается справа налево, уменьшая количество выражений столбцов, по которым он создает группы и агрегаты.

Синтаксис команды имеет следующий вид:

GROUP BY ROLLUP(<список столбцов>)

или

GROUP BY <список столбцов> WITH ROLLUP

Порядок столбцов влияет на выходные данные ROLLUP и может отразиться на количестве строк в результирующем наборе.

GROUP BY ROLLUP(col1, col2) создает группы для каждой комбинации выражений столбцов в следующих списках:

col1, col2

col1, NULL

NULL, NULL – это общий итог.

Оператор CUBE создает группы для всех возможных сочетаний столбцов. Синтаксис команды имеет следующий вид:

GROUP BY CUBE(<список столбцов>)

или

GROUP BY <список столбцов> WITH CUBE

GROUP BY CUBE(col1, col2) создает группы для каждой комбинации выражений столбцов в следующих списках:

col1, col2

col1,

NULL

NULL,

col2

NULL, NULL – это общий итог.

Оператор GROUPING SETS позволяет объединять несколько предложений GROUP BY в одно предложение GROUP BY.

Синтаксис команды имеет следующий вид:

GROUP BY GROUPING SETS(<список столбцов>)

Если параметр GROUPING SETS имеет два или более элементов, результатом будет объединение элементов.

SQL не консолидирует повторяющиеся группы, созданные для списка GROUPING SETS.

GROUP BY GROUPING SETS(col1, col2) создает группы для каждой комбинации выражений столбцов в следующих списках:

col1, NULL

NULL, col2

Для предложения GROUP BY, использующего ROLLUP, CUBE или GROUPING SETS, допускается максимум 32 выражения.

Функция GROUPING указывает, является ли указанное выражение столбца в списке GROUP BY статистическим или нет.

В результирующем наборе возврат будет 1 (статистическое выражение) или ноль (нестатистическое выражение).

Функция GROUPING может использоваться только в предложениях SELECT, HAVING и ORDER BY, если указано предложение GROUP BY.

Функция GROUPING_ID вычисляет уровень группирования.

Функция GROUPING_ID может использоваться только в предложениях SELECT, HAVING и ORDER BY, если указано предложение GROUP BY.

Оператор PIVOT поворачивает возвращающее табличное значение выражение, преобразуя уникальные значения одного столбца выражения в несколько выходных столбцов. В случае необходимости PIVOT также объединяет оставшиеся повторяющиеся значения столбца и отображает их в выходных данных.

Синтаксис имеет следующий вид:

```
SELECT <столбцы для группировки>, <пивируемые столбцы >  
FROM  
(<запрос возвращающий данных>)  
AS <псевдоним>  
PIVOT  
(<агрегирующая функция>(<столбец>)  
FOR [<столбец, значения которого будут заголовками>]  
IN (<список пивируемых столбцов>)  
) AS <псевдоним для пивот-таблицы>
```

Оператор UNPIVOT поворачивает возвращающее табличное значение выражения, преобразуя нескольких выходных столбцов в один столбец.

Синтаксис имеет следующий вид:

```
SELECT <список столбцов>  
FROM  
<таблица>  
UNPIVOT  
(<столбец значения строк> FOR [<столбец, значения заголовков>]  
IN (<список анпивируемых столбцов>)  
) AS <псевдоним для анпивот-таблицы>
```

II. Практическая часть

Создайте в своей рабочей БД следующую таблицу с данными.

id	fio	predmet	ush	ball
1	Иванова	математика	лицей	98,5
2	Петров	физика	лицей	99
3	Сидоров	математика	лицей	88
4	Полухина	физика	гимназия	78
5	Матвеева	химия	лицей	92
6	Касимов	химия	гимназия	68
7	Нестеров	математика	гимназия	81
8	Авдеев	физика	лицей	87
9	Никитина	химия	лицей	94
10	Барышев	химия	лицей	88

1.1 Запрос, который выводит количество учеников по предметам по каждой школе

```
1 use testbasa
2 --количество учеников по предметам по каждой школе,
3
4 SELECT
5     predmet,ush ,COUNT(fio) AS kol
6 FROM
7     Table_uch1
8 GROUP BY
9     predmet,ush
10
11
12
```

Результаты

predmet	ush	kol
математика	гимназия	1
физика	гимназия	1
химия	гимназия	1
математика	лицей	2
физика	лицей	2
химия	лицей	3

1.2 Запрос, который выводит количество учеников по предметам по каждой школе, и промежуточные итоги

```
1 use testbasa
2 --количество учеников по предметам по каждой школе,
3 --и промежуточные итоги
4 SELECT
5     predmet,ush ,COUNT(fio) AS kol
6 FROM
7     Table_uch1
8 GROUP BY
9     predmet,ush WITH ROLLUP
```

Результаты

predmet	ush	kol
математика	гимназия	1
математика	лицей	2
математика	NULL	3
физика	гимназия	1
физика	лицей	2
физика	NULL	3
химия	гимназия	1
химия	лицей	3
химия	NULL	4
NULL	NULL	10

2.1 Напишите запрос, который выводит количество учеников по предметам и по школам

```
12
13 --количество учеников по предметам и по школам,
14
15 SELECT
16     predmet,ush ,COUNT(fio) AS kol
17 FROM
18     Table_uch1
19
20 GROUP BY
21     predmet,ush
```

52 %

Результаты

	predmet	ush	kol
1	математика	гимназия	1
2	физика	гимназия	1
3	химия	гимназия	1
4	математика	лицей	2
5	физика	лицей	2
6	химия	лицей	3

2.2 Напишите запрос, который выводит количество учеников по предметам и по школам, и промежуточные итоги

```
24 --количество учеников по предметам и по школам,
25 --и промежуточные итоги
26 SELECT
27     predmet,ush ,COUNT(fio) AS kol
28 FROM
29     Table_uch1
30
31 GROUP BY
32     predmet,ush WITH CUBE
```

Результаты

	predmet	ush	kol
	математика	гимназия	1
	физика	гимназия	1
	химия	гимназия	1
	NULL	гимназия	3
	математика	лицей	2
	физика	лицей	2
	химия	лицей	3
	NULL	лицей	7
	NULL	NULL	10
0	математика	NULL	3
1	физика	NULL	3
2	химия	NULL	4

3. Напишите запрос, который выводит количество учеников по предметам и по школам:

```
34 --запрос, который выводит количество учеников по предметам и по школам
35 SELECT
36     predmet,ush ,COUNT(fio) AS kol
37 FROM
38     Table_uch1
39
40 GROUP BY
41     GROUPING SETS(predmet,ush)
```

Результаты

	predmet	ush	kol
	NULL	гимназия	3
	NULL	лицей	7
	математика	NULL	3
	физика	NULL	3
	химия	NULL	4

4. Напишите запрос, который выводит количество учеников по предметам по каждой школе и промежуточные итоги. NULL значения заменить на соответствующий текст

Функция **COALESCE**(выражение[,...n]) вычисляет аргументы по порядку и возвращает текущее значение первого выражения, изначально не вычисленного как NULL.

```

43
44 -- запрос, который выводит количество учеников по предметам по каждой школе
45 --и промежуточные итоги.
46 --NULL значения заменить на соответствующий текст
47 SELECT
48     COALESCE(predmet, 'ИТОГО') AS predmet
49     , COALESCE(ush, 'Итого') AS ush
50     , COUNT(fio) AS kol
51 FROM
52     Table_uch1
53 GROUP BY
54     ROLLUP(predmet,ush)
55
56

```

12 %

Результаты Сообщения

	predmet	ush	kol
1	математика	гимназия	1
2	математика	лицей	2
3	математика	Итого	3
4	физика	гимназия	1
5	физика	лицей	2
6	физика	Итого	3
7	химия	гимназия	1
8	химия	лицей	3
9	химия	Итого	4
10	ИТОГО	Итого	10

5. Напишите запрос, который выводит количество учеников по предметам и по школам, и промежуточные итоги. В итоговых строках NULL значения заменить на соответствующий текст в зависимости от группировки:

Функция **IF**(условие, выражение_если_истина, выражение_если_ложь) — возвращает одно из двух значений в зависимости от того, принимает логическое выражение значение true или false.

GROUPING (Transact-SQL) Указывает, является ли указанное выражение столбца в списке GROUP BY статистическим или нет. В результирующем наборе функция GROUPING возвращает 1 (статистическое выражение) или ноль (нестатистическое выражение). Функция GROUPING может использоваться только в предложениях SELECT <select>, HAVING и ORDER BY, если указано предложение GROUP BY.

GROUPING (<column_expression>)

```

57 --и промежуточные итоги.
58 --В итоговых строках NULL значения заменить на соответствующий текст
59 --в зависимости от группировки
60 SELECT
61     IIF(GROUPING(predmet)=1, 'ИТОГО', predmet) AS predmet
62     , IIF(GROUPING(ush)=1, 'Итого', ush) AS ush
63     , COUNT(fio) AS kol
64
65 FROM
66     Table_uch1
67
68 GROUP BY
69     CUBE (predmet, ush)
70

```

predmet	ush	kol
математика	гимназия	1
физика	гимназия	1
химия	гимназия	1
ИТОГО	гимназия	3
математика	лицей	2
физика	лицей	2
химия	лицей	3
ИТОГО	лицей	7
ИТОГО	Итого	10

6. Напишите запрос, который выводит количество учеников по предметам и по школам. В итоговых строках NULL значения заменить на соответствующий текст в зависимости от уровней группировки

Функция *ISNULL(выражение, замена)* заменяет значение *NULL* указанным замещающим значением

```

71 --запрос, который выводит количество учеников по предметам и по школам
72 --в итоговых строках NULL значения заменить на соответствующий текст
73 --в зависимости от уровней группировки
74
75 SELECT
76     CASE GROUPING_ID(predmet, ush)
77     WHEN 1 THEN 'Итого по предметам'
78     WHEN 3 THEN 'Итого'
79     ELSE ''
80     END AS Итого
81     , ISNULL(predmet, '') AS predmet
82     , ISNULL(ush, '') AS ush
83     , COUNT(fio) AS kol
84
85 FROM
86     Table_uch1
87
88 GROUP BY
89     ROLLUP(predmet, ush)

```

	Итого	predmet	ush	kol
1		математика	гимназия	1
2		математика	лицей	2
3	Итого по предметам	математика		3
4		физика	гимназия	1
5		физика	лицей	2
6	Итого по предметам	физика		3
7		химия	гимназия	1
8		химия	лицей	3
9	Итого по предметам	химия		4
10	Итого			10

III. Самостоятельная работа

1. Напишите запрос, который выводит максимальный балл учеников по школам, по каждому предмету по каждой школе и промежуточные итоги.
2. Напишите запрос, который выводит минимальный балл учеников по школам и по предметам, и промежуточные итоги.
3. Напишите запрос, который выводит средний балл учеников по школам и по предметам.
4. Напишите запрос, который выводит количество учеников по каждой школе по предметам и промежуточные итоги. NULL значения заменить на соответствующий текст.
5. Напишите запрос, который выводит суммарный балл учеников по школам и по предметам, и промежуточные итоги. В итоговых строках NULL значения заменить на соответствующий текст в зависимости от группировки.
6. Напишите запрос, который выводит максимальный балл учеников по школам и по предметам. В итоговых строках NULL значения заменить на соответствующий текст в зависимости от уровней группировки.