# Определения (Лекция 16)

**Создание сводных таблиц:**

Для визуализации информации удобно переходить от вида в 3НФ к сводной таблице:

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Месяц | Год | Колво |  | Год | 1 | 2 | …. |
| 1 | 2020 | 15 |  | 2020 | 15 | 16 |  |
| 2 | 2020 | 16 | => | 2021 | 20 |  |  |
| … |  |  |  |  |  |  |  |
| 1 | 2021 | 20 |  |  |  |  |  |

DECLARE @Данные TABLE (Месяц int, Год int, Колво int)

INSERT INTO @Данные VALUES (1, 2020, 15), (1, 2021, 20), (2, 2020, 16), (2, 2021, 21), (3, 2020, 17), (3, 2021, 22), (4, 2020, 13), (4, 2021, 2), (5, 2020, 16), (6, 2020, 18), (7, 2020, 13), (8, 2020, 10), (9, 2020, 15), (10, 2020, 9), (11, 2020, 17), (12, 2020, 11), (1, 2019, 2), (2, 2019, 6), (3, 2019, 8), (4, 2019, 7), (5, 2019, 3), (6, 2019, 8), (7, 2019, 3), (8, 2019, 1), (9, 2019, 5), (10, 2019, 8), (11, 2019, 7), (12, 2019, 2)

SELECT Год,

SUM(CASE WHEN Месяц = 1 THEN Колво ELSE 0 END) Колво1,

SUM(CASE WHEN Месяц = 2 THEN Колво ELSE 0 END) Колво2,

. . . - - по аналогии для остальных месяцев

SUM(CASE WHEN Месяц = 12 THEN Колво ELSE 0 END) Колво12

FROM @Данные

GROUP BY Год

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Год** | **Колво1** | **Колво2** | **Колво3** | **Колво4** | **Колво5** | **Колво6** | **Колво7** | **Колво8** | **Колво9** | **Колво10** | **Колво11** | **Колво12** |
| 2019 | 2 | 6 | 8 | 7 | 3 | 8 | 3 | 1 | 5 | 8 | 7 | 2 |
| 2020 | 15 | 16 | 17 | 13 | 16 | 18 | 13 | 10 | 15 | 9 | 17 | 11 |
| 2021 | 20 | 21 | 22 | 2 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**Применение PIVOT**

SELECT Год, [1], [2], [3], [4], [5], [6], [7], [8], [9], [10], [11], [12]

FROM **(SELECT Год, Месяц, Колво FROM @Данные)** AS D

PIVOT

(

SUM(Колво)

FOR Месяц IN([1],[2],[3],[4],[5],[6],[7],[8],[9],[10],[11],[12])

) AS P;

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Год** | **1** | **2** | **3** | **4** | **5** | **6** | **7** | **8** | **9** | **10** | **11** | **12** |
| 2019 | 2 | 6 | 8 | 7 | 3 | 8 | 3 | 1 | 5 | 8 | 7 | 2 |
| 2020 | 15 | 16 | 17 | 13 | 16 | 18 | 13 | 10 | 15 | 9 | 17 | 11 |
| 2021 | 20 | 21 | 22 | 2 | **NULL** | **NULL** | **NULL** | **NULL** | **NULL** | **NULL** | **NULL** | **NULL** |

**Как от сводной таблицы вернуться к удобному для анализа виду:**

DECLARE @Сводная TABLE (Год int, Колво1 int, Колво2 int, Колво3 int, Колво4 int, Колво5 int, Колво6 int)

INSERT INTO @Сводная VALUES (2019,2,6,8,7,3,8),(2020,15,16,17,13,16,18),(2021,20,21,22,2,0,NULL)

SELECT Год,

LEFT(REVERSE(Месяц), **PATINDEX('%[^0-9]%', REVERSE(Месяц))** - 1) Месяц, Колво

FROM

(

SELECT Год, Колво1, Колво2, Колво3, Колво4, Колво5, Колво6

FROM @Сводная

) AS T

UNPIVOT

(

Колво

FOR Месяц IN (Колво1, Колво2, Колво3, Колво4, Колво5, Колво6)

) AS unpvt

-- проверьте, что в таком виде в данных не будет данных для 6-ого месяца 2021 года

**Еще способ сделать Unpivot:**

SELECT Год, Месяц,

CASE Месяц

WHEN 1 THEN Колво1

WHEN 2 THEN Колво2

WHEN 3 THEN Колво3

WHEN 4 THEN Колво4

WHEN 5 THEN Колво5

WHEN 6 THEN Колво6

END AS Колво

FROM @Сводная

CROSS JOIN (VALUES(1),(2),(3),(4),(5),(6)) AS MNTH(Месяц);

**Комментарии:**

- понятно ли Вам, как от сводной таблицы вернуться к виду, удобному для анализа, при помощи UNION?

- если колво столбцов заранее неизвестно, то можно комбинировать pivot и динамический SQL.

**Накопительная и скользящая статистики**

Способ: (с 2020 версии)

SELECT Год, Месяц, Колво,

SUM(Колво) OVER (

PARTITION BY Год ORDER BY Месяц

ROWS BETWEEN UNBOUNDED PRECEDING AND CURRENT ROW

/\***ROWS** BETWEEN 2 PRECEDING AND 1 FOLLOWING \*/

) СуммКолво

FROM @Данные;

|  |  |
| --- | --- |
| Накопительная статистика (Станд синтаксис) | Скользящая статистика (Станд синтаксис) |
| SELECT T1.Год, T1.Месяц, T1.Колво, SUM(T2.Колво) СуммКолво  FROM @Данные T1 INNER JOIN @Данные T2 ON  T1.Год = T2.Год AND  T1.Месяц >= T2.Месяц  GROUP BY T1.Год, T1.Месяц, T1.Колво  ORDER BY 1, 2 | SELECT T1.Год, T1.Месяц, T1.Колво, SUM(T2.Колво) СуммКолво  FROM @Данные T1 INNER JOIN @Данные T2 ON  T1.Год = T2.Год AND  T1.Месяц >= T2.Месяц **AND**  **T2.Месяц > T1.Месяц - 3**  GROUP BY T1.Год, T1.Месяц, T1.Колво  ORDER BY 1, 2 |
| |  |  |  |  | | --- | --- | --- | --- | | Год | Месяц | Колво | СуммКолво | | 2019 | 1 | 2 | 2 | | 2019 | 2 | 6 | 8 | | 2019 | 3 | 8 | 16 | | 2019 | 4 | 7 | 23 | | 2019 | 5 | 3 | 26 | | 2019 | 6 | 8 | 34 | | 2019 | 7 | 3 | 37 | | 2019 | 8 | 1 | 38 | | 2019 | 9 | 5 | 43 | | 2019 | 10 | 8 | 51 | | 2019 | 11 | 7 | 58 | | 2019 | 12 | 2 | 60 | | |  |  |  |  | | --- | --- | --- | --- | | Год | Месяц | Колво | СуммКолво | | 2019 | 1 | 2 | 2 | | 2019 | 2 | 6 | 8 | | 2019 | 3 | 8 | 16 | | 2019 | 4 | 7 | 21 | | 2019 | 5 | 3 | 18 | | 2019 | 6 | 8 | 18 | | 2019 | 7 | 3 | 14 | | 2019 | 8 | 1 | 12 | | 2019 | 9 | 5 | 9 | | 2019 | 10 | 8 | 14 | | 2019 | 11 | 7 | 20 | | 2019 | 12 | 2 | 17 | |

Не забывайте про подзапросы в SELECT (например, в случае иных версий СУБД), они тоже помогут.

SELECT Год, Месяц, Колво,

(

SELECT SUM(Данные.Колво)

FROM @Данные Данные

WHERE Данные.Год = T.Год AND **T.Месяц >= Данные.Месяц AND Данные.Месяц > T.Месяц-3**

) СуммКолво

FROM @Данные T

ORDER BY 1, 2

**Домашнее задание:** сделать аналогичные запросы при помощи ranking function (нумерация строк).

**Способы группировки данных:**

DECLARE @Документы TABLE (ндок int, Дата datetime, Пок\_ID int, Товар\_ID int, Колво int, primary key(ндок, Товар\_ID))

INSERT INTO @Документы VALUES

(1, '20210115', 1, 1, 15),

(1, '20210115', 1, 2, 10),

(2, '20210215', 2, 1, 5)

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| SELECT  **GROUPING\_ID(Пок\_ID, Товар\_ID, YEAR(Дата), MONTH(Дата)) Грп**,  YEAR(Дата) Год, MONTH(Дата) Месяц, Пок\_ID, Товар\_ID, SUM(Колво) СуммКолво  FROM @Документы  GROUP BY **GROUPING SETS**  (  ( Пок\_ID, Товар\_ID, YEAR(Дата), MONTH(Дата) ),  ( Пок\_ID, YEAR(Дата), MONTH(Дата) ),  ( Товар\_ID, YEAR(Дата), MONTH(Дата) ),  ( Товар\_ID, YEAR(Дата) ),  ()  )  ORDER BY 1 | |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | | Грп | Год | Месяц | Пок\_ID | Товар\_ID | СуммКолво | | 0 | 2021 | 1 | 1 | 1 | 15 | | 0 | 2021 | 2 | 2 | 1 | 5 | | 0 | 2021 | 1 | 1 | 2 | 10 | | 4 | 2021 | 1 | 1 | NULL | 25 | | 4 | 2021 | 2 | 2 | NULL | 5 | | 8 | 2021 | 1 | NULL | 2 | 10 | | 8 | 2021 | 2 | NULL | 1 | 5 | | 8 | 2021 | 1 | NULL | 1 | 15 | | 9 | 2021 | NULL | NULL | 1 | 20 | | 9 | 2021 | NULL | NULL | 2 | 10 | | 15 | NULL | NULL | NULL | NULL | 30 | |

SELECT 0 Грп, YEAR(Дата) Год, MONTH(Дата) Месяц, Пок\_ID, Товар\_ID, SUM(Колво) СуммКолво

FROM @Документы

GROUP BY YEAR(Дата), MONTH(Дата), Пок\_ID, Товар\_ID

UNION

SELECT 4, YEAR(Дата) Год, MONTH(Дата) Месяц, Пок\_ID, NULL, SUM(Колво) СуммКолво

FROM @Документы

GROUP BY YEAR(Дата), MONTH(Дата), Пок\_ID

UNION

SELECT 8, YEAR(Дата) Год, MONTH(Дата) Месяц, NULL, Товар\_ID, SUM(Колво) СуммКолво

FROM @Документы

GROUP BY YEAR(Дата), MONTH(Дата), Товар\_ID

UNION

SELECT 9, YEAR(Дата) Год, NULL, NULL, Товар\_ID, SUM(Колво) СуммКолво

FROM @Документы

GROUP BY YEAR(Дата), Товар\_ID

UNION

SELECT 15, NULL, NULL, NULL, NULL, SUM(Колво) СуммКолво FROM @Документы

**Про Grouping\_ID():**

GROUPING\_ID (a, b) – что выдает, если группировка происходит по указанному множеству:

|  |  |  |
| --- | --- | --- |
|  | 11 | 3 |
| a | 01 | 1 |
| b | 10 | 2 |
| ab | 00 | 0 |

То есть если атрибут не входит в группировку, то его значение для функции – 1, иначе 0.

Соответственно, на примере выше: GROUPING\_ID(Пок\_ID, Товар\_ID, YEAR(Дата), MONTH(Дата)) и в GROUP BY ( Товар\_ID, YEAR(Дата) ), то это равносильно 1001, что равно 1\*2^3 + 1\*2^0 = 9.

**Еще способ понять, входит ли поле в группировку:**

SELECT GROUPING\_ID(Пок\_ID, Товар\_ID, YEAR(Дата), MONTH(Дата)) Грп,

YEAR(Дата) Год, MONTH(Дата) Месяц, Пок\_ID, Товар\_ID, SUM(Колво) СуммКолво,

**GROUPING(Пок\_ID)**

FROM @Документы

GROUP BY GROUPING SETS

(

( Пок\_ID, Товар\_ID, YEAR(Дата), MONTH(Дата) ),

( Пок\_ID, YEAR(Дата), MONTH(Дата) ),

( Товар\_ID, YEAR(Дата), MONTH(Дата) ),

( Товар\_ID, YEAR(Дата) ),

()

)

ORDER BY 1

Функция GROUPING, которая принимает на вход название столбца, и выдает 1 – если это поле не участвует в группировке в этой строке и 0 – если это поле участвует в группировке в этой строке.

**CUBE() и ROLLUP()**

SELECT **GROUPING\_ID(Пок\_ID, Товар\_ID, YEAR(Дата), MONTH(Дата)) Грп**,

YEAR(Дата) Год, MONTH(Дата) Месяц, Пок\_ID, Товар\_ID, SUM(Колво) СуммКолво

FROM @Документы

GROUP BY **CUBE (Пок\_ID, Товар\_ID, YEAR(Дата)**, **MONTH(Дата))**

ORDER BY 1

DECLARE @Сводная TABLE (Год int, Месяц int, Колво int)

INSERT INTO @Сводная

VALUES (2021, 1, 50), (2021, 2, 100), (2019, 1, 25), (2019, 2, 30), (2020, 1, 35), (2020, 2, 80)

|  |  |
| --- | --- |
| SELECT GROUPING\_ID(Год, Месяц) Грп,  Год, Месяц, SUM(Колво) СуммКолво  FROM @Сводная  GROUP BY **CUBE(Год, Месяц)**  ORDER BY 1 | SELECT GROUPING\_ID(Год, Месяц) Грп,  Год, Месяц, SUM(Колво) СуммКолво  FROM @Сводная  GROUP BY **ROLLUP(Год, Месяц)**  ORDER BY 1 |
| |  |  |  |  | | --- | --- | --- | --- | | **Грп** | **Год** | **Месяц** | **СуммКолво** | | 0 | 2021 | 1 | 50 | | 0 | 2019 | 1 | 25 | | 0 | 2020 | 1 | 35 | | 0 | 2021 | 2 | 100 | | 0 | 2019 | 2 | 30 | | 0 | 2020 | 2 | 80 | | 1 | 2021 | NULL | 150 | | 1 | 2019 | NULL | 55 | | 1 | 2020 | NULL | 115 | | 2 | NULL | 2 | 210 | | 2 | NULL | 1 | 110 | | 3 | NULL | NULL | 320 | | |  |  |  |  | | --- | --- | --- | --- | | **Грп** | **Год** | **Месяц** | **СуммКолво** | | 0 | 2021 | 1 | 50 | | 0 | 2021 | 2 | 100 | | 0 | 2019 | 1 | 25 | | 0 | 2019 | 2 | 30 | | 0 | 2020 | 1 | 35 | | 0 | 2020 | 2 | 80 | | 1 | 2020 | NULL | 115 | | 1 | 2019 | NULL | 55 | | 1 | 2021 | NULL | 150 | | 3 | NULL | NULL | 320 | |

SELECT GROUPING\_ID(Год, Месяц) Грп,

Год, Месяц, SUM(Колво) СуммКолво

FROM @Сводная

GROUP BY **GROUPING SETS**

(

(Год),

(Год, Месяц),

()

)

ORDER BY 1

**Dynamic SQL:**

**Exec:**

Простой вариант:

DECLARE @sql1 AS NVARCHAR(100);

SET @sql1 = 'SELECT \* FROM Товары WHERE Товар\_ID = 1;';

EXEC (@sql1)

Вариант с подстановкой параметра:

DECLARE @sql2 AS NVARCHAR(100), @n as nvarchar(2);

SET @n = 1

SET @sql2 = 'SELECT \* FROM Товары WHERE Товар\_ID = CAST(' + @n + ' as int)';

EXEC (@sql2)

Вариант с SQL инъекцией:

DECLARE @sql3 AS NVARCHAR(100), @k as nvarchar(100);

SET @k = '1; DROP TABLE Товары'

SET @sql3 = 'SELECT \* FROM Товары WHERE Товар\_ID = ' + @k + '';

EXEC (@sql3)

**sp\_executeSQL:**

DECLARE @sql4 AS NVARCHAR(100);

SET @sql4 = 'SELECT TOP 1 \* FROM Товары WHERE Товар\_ID = @good;';

EXEC sp\_executesql

@stmt = @sql4,

@params = N'@good AS INT',

@good = 1;

DECLARE @sql5 AS NVARCHAR(200);

SET @sql5 = 'SELECT \* FROM Товары WHERE Товар = @good;';

EXEC sp\_executesql

@stmt = @sql5,

@params = N'@good AS NVARCHAR(200)',

@good = '1; DROP TABLE Товары';