

## Лабораторная работа №2. Составление отчетов. Транспонирование результирующих множеств.

### 1. Разворачивание результирующего множества в одну строку

#### Задача

Требуется транспонировать (развернуть) группу строк, превращая их значения в столбцы.

Пусть имеется результирующее множество, информирующее о количестве сотрудников в каждом отделе:

DEPTNO	COUNT(*)
10	3
20	2
30	3
40	2
-	1

Требуется переформатировать результат так, чтобы множество имело вид:

DEPTNO_10	DEPTNO_20	DEPTNO_30	DEPTNO_40
3	2	3	2

#### Решение

##### Вариант 1

```
SELECT SUM(CASE WHEN deptno=10 THEN 1 ELSE 0 END) AS deptno_10,
       SUM(CASE WHEN deptno=20 THEN 1 ELSE 0 END) AS deptno_20,
       SUM(CASE WHEN deptno=30 THEN 1 ELSE 0 END) AS deptno_30,
       SUM(CASE WHEN deptno=40 THEN 1 ELSE 0 END) AS deptno_40
FROM career;
```

##### Вариант 2

```
SELECT MAX(CASE WHEN deptno=10 THEN empcount ELSE NULL END) AS deptno_10,
       MAX(CASE WHEN deptno=20 THEN empcount ELSE NULL END) AS deptno_20,
       MAX(CASE WHEN deptno=30 THEN empcount ELSE NULL END) AS deptno_30,
       MAX(CASE WHEN deptno=40 THEN empcount ELSE NULL END) AS deptno_40
FROM (
  SELECT deptno, COUNT(*) AS empcount
  FROM career
  GROUP BY deptno);
```

#### Обсуждение

##### Вариант 1

Выражение используется для расстановки «флагов», обозначая к какому DEPTNO относится строка:

```
SELECT deptno, CASE WHEN deptno=10 THEN 1 ELSE 0 END AS deptno_10,
       CASE WHEN deptno=20 THEN 1 ELSE 0 END AS deptno_20,
       CASE WHEN deptno=30 THEN 1 ELSE 0 END AS deptno_30,
       CASE WHEN deptno=40 THEN 1 ELSE 0 END AS deptno_40
FROM career
```

DEPTNO	DEPTNO_10	DEPTNO_20	DEPTNO_30	DEPTNO_40
10	1	0	0	0
10	1	0	0	0
20	0	1	0	0
30	0	0	1	0
20	0	1	0	0
30	0	0	1	0
-	0	0	0	0
30	0	0	1	0
10	1	0	0	0
40	0	0	0	1
40	0	0	0	1

Т. о. преобразование «строк в столбцы» уже выполнено. Следующий шаг – сложить значения, возвращенные в столбцах DEPTNO\_10, DEPTNO\_20, DEPTNO\_30, DEPTNO\_40.

## Вариант 2

В этом подходе для подсчета количества служащих в отделе используется вложенный запрос. Выражение CASE внешнего запроса преобразует строки в столбцы, предоставляя следующие результаты:

```
SELECT CASE WHEN deptno=10 THEN empcount ELSE null END AS deptno_10,
       CASE WHEN deptno=20 THEN empcount ELSE null END AS deptno_20,
       CASE WHEN deptno=30 THEN empcount ELSE null END AS deptno_30,
       CASE WHEN deptno=40 THEN empcount ELSE null END AS deptno_40
FROM (
  SELECT deptno, count(*) as empcount
  FROM career
  GROUP BY deptno
);
```

DEPTNO_10	DEPTNO_20	DEPTNO_30	DEPTNO_40
-	-	3	-
-	-	-	-
-	2	-	-
-	-	-	2
3	-	-	-

## 2. Разворачивание результирующего множества в несколько строк

### Задача

Требуется преобразовать строки в столбцы, создавая для каждого значения заданного столбца отдельный столбец.

Пусть имеется результирующее множество, информирующее о должностях, которые когда-либо занимали сотрудники:

JOBNAME	EMPNAME
MANAGER	JOHN KLINTON
MANAGER	JONES
FINANCIAL DIRECTOR	ALLEN
FINANCIAL DIRECTOR	ALLEN
EXECUTIVE DIRECTOR	ALLEN
SALESMAN	SMITH
CLERK	SMITH
DRIVER	ALEX BOUSH
PRESIDENT	SCOTT

Требуется переформатировать результат так, чтобы множество имело вид:

PRESIDENTS	FINANCIAL_DIRECTORS	MANAGERS	CLERKS	DRIVERS
SCOTT	ALLEN	JOHN KLINTON	SMITH	ALEX BOUSH
-	ALLEN	JONES	-	-

### Решение

```
SELECT MAX(CASE WHEN jobname='PRESIDENT' THEN empname ELSE NULL END) AS PRESIDENTS,
       MAX(CASE WHEN jobname='FINANCIAL DIRECTOR' THEN empname ELSE NULL END) AS FINANCIAL_DIRECTORS,
       MAX(CASE WHEN jobname='MANAGER' THEN empname ELSE NULL END) AS MANAGERS,
       MAX(CASE WHEN jobname='CLERK' THEN empname ELSE NULL END) AS CLERKS,
       MAX(CASE WHEN jobname='DRIVER' THEN empname ELSE NULL END) AS DRIVERS
FROM (
  SELECT jobname, empname, ROW_NUMBER()OVER(PARTITION BY jobname ORDER BY empname) rn
  FROM emp JOIN (career JOIN job USING (jobno)) USING (empno)
)
GROUP BY rn;
```

### Обсуждение

Чтобы решить данную задачу необходимо сделать каждое сочетание JOBNAME+EMPNAME уникальным. С этой целью используем ранжирующую функцию ROW\_NUMBER OVER:

```
SELECT jobname, empname, ROW_NUMBER()OVER(PARTITION BY jobname ORDER BY empname) rn
FROM emp JOIN (career JOIN job USING (jobno)) USING (empno);
```

Results Explain Describe Saved SQL History

JOBNAME	EMPNAME	RN
CLERK	SMITH	1
DRIVER	ALEX BOUSH	1
EXECUTIVE DIRECTOR	ALLEN	1
FINANCIAL DIRECTOR	ALLEN	1
FINANCIAL DIRECTOR	ALLEN	2
MANAGER	JOHN KLINTON	1
MANAGER	JONES	2
PRESIDENT	SCOTT	1
SALESMAN	SMITH	1

Следующий шаг – использовать выражение CASE для распределения значений EMPNAME по соответствующим столбцам JOBNAME:

```
SELECT CASE WHEN jobname='PRESIDENT' THEN empname ELSE NULL END AS PRESIDENTS,
CASE WHEN jobname='FINANCIAL DIRECTOR' THEN empname ELSE NULL END AS FINANTIAL_DIRECTORS,
CASE WHEN jobname='MANAGER' THEN empname ELSE NULL END AS MANAGERS,
CASE WHEN jobname='CLERK' THEN empname ELSE NULL END AS CLERKS,
CASE WHEN jobname='DRIVER' THEN empname ELSE NULL END AS DRIVERS
FROM (
SELECT jobname, empname, ROW_NUMBER()OVER(PARTITION BY jobname ORDER BY empname) rn
FROM emp JOIN (career JOIN job USING (jobno)) USING (empno)
);
```

Results Explain Describe Saved SQL History

PRESIDENTS	FINANTIAL_DIRECTORS	MANAGERS	CLERKS	DRIVERS
-	-	JOHN KLINTON	-	-
-	-	JONES	-	-
-	ALLEN	-	-	-
-	ALLEN	-	-	-
-	-	-	-	-
-	-	-	-	-
-	-	-	SMITH	-
-	-	-	-	ALEX BOUSH
SCOTT	-	-	-	-

В результате строки транспонированы в столбцы, осталось удалить значения NULL. Используем для этого агрегатную функцию MAX и группируем результаты по RN:

```
SELECT MAX(CASE WHEN jobname='PRESIDENT' THEN empname ELSE NULL END) AS PRESIDENTS,
MAX(CASE WHEN jobname='FINANCIAL DIRECTOR' THEN empname ELSE NULL END) AS FINANTIAL_DIRECTORS,
MAX(CASE WHEN jobname='MANAGER' THEN empname ELSE NULL END) AS MANAGERS,
MAX(CASE WHEN jobname='CLERK' THEN empname ELSE NULL END) AS CLERKS,
MAX(CASE WHEN jobname='DRIVER' THEN empname ELSE NULL END) AS DRIVERS
FROM (
SELECT jobname, empname, ROW_NUMBER()OVER(PARTITION BY jobname ORDER BY empname) rn
FROM emp JOIN (career JOIN job USING (jobno)) USING (empno)
)
GROUP BY rn;
```

Results Explain Describe Saved SQL History

PRESIDENTS	FINANTIAL_DIRECTORS	MANAGERS	CLERKS	DRIVERS
SCOTT	ALLEN	JOHN KLINTON	SMITH	ALEX BOUSH
-	ALLEN	JONES	-	-

### 3. Обратное разворачивание результирующего множества

#### Задача

Требуется преобразовать столбцы в строки. Имеется следующее результирующее множество:

DEPTNO_10	DEPTNO_20	DEPTNO_30	DEPTNO_40
1	2	3	2

Требуется преобразовать его к виду:

DEPTNO	COUNTS_BY_DEPT
10	1
20	2
30	3
40	2

## Решение

```
SELECT d.deptno,
       CASE d.deptno
         WHEN 10 THEN emp_cnts.deptno_10
         WHEN 20 THEN emp_cnts.deptno_20
         WHEN 30 THEN emp_cnts.deptno_30
         WHEN 40 THEN emp_cnts.deptno_40
       END AS counts_by_dept
FROM (
  SELECT SUM(CASE WHEN deptno=10 THEN 1 ELSE 0 END) AS deptno_10,
         SUM(CASE WHEN deptno=20 THEN 1 ELSE 0 END) AS deptno_20,
         SUM(CASE WHEN deptno=30 THEN 1 ELSE 0 END) AS deptno_30,
         SUM(CASE WHEN deptno=40 THEN 1 ELSE 0 END) AS deptno_40
  FROM career
) emp_cnts,
(SELECT deptno FROM dept) d
```

## Обсуждение

Вложенный запрос EMP\_CNTS является исходным результирующим множеством, которое требуется преобразовать в строки:

```
SELECT SUM(CASE WHEN deptno=10 THEN 1 ELSE 0 END) AS deptno_10,
       SUM(CASE WHEN deptno=20 THEN 1 ELSE 0 END) AS deptno_20,
       SUM(CASE WHEN deptno=30 THEN 1 ELSE 0 END) AS deptno_30,
       SUM(CASE WHEN deptno=40 THEN 1 ELSE 0 END) AS deptno_40
FROM career
) emp_cnts,
```

DEPTNO_10	DEPTNO_20	DEPTNO_30	DEPTNO_40
1	2	3	2

Поскольку в исходном результирующем множестве четыре столбца, то будет создано четыре строки.

Выполним декартово произведение между вложенным запросом EMP\_CNTS и некоторой таблицей, имеющей не менее четырех строк. В качестве такой таблицы используем таблицу DEPT (Упражнение: убедитесь, что в таблице DEPT четыре строки).

```
SELECT d.deptno,
       emp_cnts.deptno_10,
       emp_cnts.deptno_20,
       emp_cnts.deptno_30,
       emp_cnts.deptno_40
FROM (
  SELECT SUM(CASE WHEN deptno=10 THEN 1 ELSE 0 END) AS deptno_10,
         SUM(CASE WHEN deptno=20 THEN 1 ELSE 0 END) AS deptno_20,
         SUM(CASE WHEN deptno=30 THEN 1 ELSE 0 END) AS deptno_30,
         SUM(CASE WHEN deptno=40 THEN 1 ELSE 0 END) AS deptno_40
  FROM career
) emp_cnts,
(SELECT deptno FROM dept) d
```

DEPTNO	DEPTNO_10	DEPTNO_20	DEPTNO_30	DEPTNO_40
10	1	2	3	2
20	1	2	3	2
30	1	2	3	2
40	1	2	3	2

Декартово произведение позволяет получить по строке для каждого столбца вложенного запроса EMP\_CNTS. Поскольку в окончательное результирующее множество должны войти только значения DEPTNO и количество служащих в соответствующем DEPTNO, используем выражение CASE для преобразования четырех столбцов в один:

```
SELECT d.deptno,
       CASE d.deptno
         WHEN 10 THEN emp_cnts.deptno_10
         WHEN 20 THEN emp_cnts.deptno_20
         WHEN 30 THEN emp_cnts.deptno_30
         WHEN 40 THEN emp_cnts.deptno_40
       END AS counts_by_dept
FROM (
  SELECT SUM(CASE WHEN deptno=10 THEN 1 ELSE 0 END) AS deptno_10,
         SUM(CASE WHEN deptno=20 THEN 1 ELSE 0 END) AS deptno_20,
         SUM(CASE WHEN deptno=30 THEN 1 ELSE 0 END) AS deptno_30,
         SUM(CASE WHEN deptno=40 THEN 1 ELSE 0 END) AS deptno_40
  FROM career
) emp_cnts,
   (SELECT deptno FROM dept) d
```

DEPTNO	COUNTS_BY_DEPT
10	1
20	2
30	3
40	2

#### 4. Обратное разворачивание результирующего множества в один столбец

##### Задача

Имеется представление EMP\_S\_INFO:

EMPNO	EMPNAME	JOBNAME	DEPTNO	AVG_SALARY
7369	SMITH	CLERK	30	2704
7499	ALLEN	FINANCIAL DIRECTOR	10	4931
7790	JOHN KLINTON	MANAGER	40	650
7789	ALEX BOUSH	DRIVER	40	1913

Требуется вывести все возвращаемые запросом столбцы в одном столбце. Например, стоит задача получить EMPNAME, JOBNAME, AVG\_SALARY всех сотрудников 40-го отдела. Все три значения должны быть выведены в одном столбце в трех строках для каждого сотрудника, и значения для разных сотрудников должны быть разделены пустой строкой:

EMP_S
ALEX BOUSH
DRIVER
1913
-
JOHN KLINTON
MANAGER
650
-

##### Решение

```

SELECT CASE rn
        WHEN 1 THEN empname
        WHEN 2 THEN jobname
        WHEN 3 THEN cast(avg_salary as char(4))
      END emps
FROM (
  SELECT e.empname, e.jobname, e.avg_salary,
         ROW_NUMBER()OVER(PARTITION BY e.empno
                           ORDER BY e.empno) rn
  FROM emp_info e,
       (SELECT *
        FROM emp_info) four_rows
  WHERE e.deptno = 40
)

```

### Обсуждение

Используя ранжирующую функцию ROW\_NUMBER OVER присваиваем ранг каждому сотруднику 40-го отдела:

```

SELECT e.empname, e.jobname, e.avg_salary,
       ROW_NUMBER()OVER(PARTITION BY e.empno
                         ORDER BY e.empno) rn
FROM emp_info e
WHERE e.deptno = 40

```

EMPNAME	JOBNAME	AVG_SALARY	RN
ALEX BOUSH	DRIVER	1913	1
JOHN KLINTON	MANAGER	650	1

Сегментирование проводится по EMPNO, поэтому всем сотрудникам 40-го отдела присвоен ранг 1.

Выполним декартово произведение между полученным результирующим множеством и некоторой таблицей, имеющей четыре строки (в данном примере в качестве такой таблицы выступает представление EMPS\_INFO:

```

CREATE OR REPLACE VIEW emp_info AS (
  SELECT empno, empname, jobname, deptno, ROUND(AVG(salvalue)) AS avg_salary
  FROM emp JOIN salary USING(empno) JOIN career USING(empno) JOIN job USING(jobno) JOIN dept
  USING(deptno)
  WHERE enddate IS NULL
  GROUP BY empno, empname, jobname, deptno)

```

```

SELECT e.empname, e.jobname, e.avg_salary,
       ROW_NUMBER()OVER(PARTITION BY e.empno
                         ORDER BY e.empno) rn
FROM emp_info e,
     (SELECT *
      FROM emp_info) four_rows
WHERE e.deptno = 40

```

EMPNAME	JOBNAME	AVG_SALARY	RN
ALEX BOUSH	DRIVER	1913	1
ALEX BOUSH	DRIVER	1913	2
ALEX BOUSH	DRIVER	1913	3
ALEX BOUSH	DRIVER	1913	4
JOHN KLINTON	MANAGER	650	1
JOHN KLINTON	MANAGER	650	2
JOHN KLINTON	MANAGER	650	3
JOHN KLINTON	MANAGER	650	4

RN равен 1 теперь не для всех сотрудников; теперь это повторяющаяся последовательность значений от 1 до 4, потому что ранжирующие функции применяются после выполнения операторов FROM и WHERE. Таким образом,

сегментирование по EMPNO обуславливает сброс RN в начальное значение (1) для записи нового сотрудника.

Вложенный запрос four\_rows – это SQL-выражение, возвращающее четыре строки. Должно быть получено по строке для каждого столбца (EMPNAME, JOBNAME, AVG\_SALARY) плюс строка столбец.

Теперь с помощью выражения CASE разместить значения EMPNAME, JOBNAME, AVG\_SALARY всех сотрудников в один столбец (чтобы значения AVG\_SALARY могли использоваться в CASE, их необходимо привести к строковому типу):

```
SELECT CASE rn
        WHEN 1 THEN empname
        WHEN 2 THEN jobname
        WHEN 3 THEN cast(avg_salary as char(4))
      END emps
FROM (
  SELECT e.empname, e.jobname, e.avg_salary,
         ROW_NUMBER()OVER(PARTITION BY e.empno
                           ORDER BY e.empno) rn
  FROM emp_info e,
       (SELECT *
        FROM emp_info) four_rows
 WHERE e.deptno = 40
)
```

EMPS
ALEX BOUSH
DRIVER
1913
-
JOHN KLINTON
MANAGER
650
-

## 5. Исключение повторяющихся значений из результирующего множества

### Задача

При формировании отчета выдвинуто требование о том, что дублирующиеся значения в столбце должны отображаться только один раз. Например, требуется извлечь значения DEPTNO и EMPNAME, при этом необходимо сгруппировать вместе все строки для каждого значения DEPTNO и выводить каждое значение DEPTNO только один раз:

DEPTNO	EMPNAME
10	ALLEN
20	SMITH
-	JONES
30	SMITH
-	ALLEN
-	ALLEN
40	JOHN KLINTON
-	ALEX BOUSH

### Решение

```
SELECT TO_NUMBER(
        DECODE(LAG(deptno) OVER(ORDER BY deptno),
              deptno, NULL, deptno)
      ) deptno, empname
  FROM emp JOIN career USING(empno) JOIN dept USING(deptno)
```

### Обсуждение

С помощью оконной функции LAG OVER вернуть предыдущие значение DEPTNO для каждой строки:

```
SELECT LAG(deptno) OVER(ORDER BY deptno) lag_deptno,
       deptno, empname
FROM emp JOIN career USING(empno) JOIN dept USING(deptno)
```

LAG_DEPTNO	DEPTNO	EMPNAME
-	10	ALLEN
10	20	SMITH
20	20	JONES
20	30	SMITH
30	30	ALLEN
30	30	ALLEN
30	40	JOHN KLINTON
40	40	ALEX BOUSH

Для строк, в которых значения DEPTNO и LAG\_DEPTNO совпадают, должно быть присвоено значение NULL. Для этого используется функция DECODE (функция TO\_NUMBER включена, чтобы привести значение DEPTNO к числовому типу):

```
SELECT TO_NUMBER(
       DECODE(LAG(deptno) OVER(ORDER BY deptno),
              deptno, NULL, deptno)
       ) deptno, empname
FROM emp JOIN career USING(empno) JOIN dept USING(deptno)
```

DEPTNO	EMPNAME
10	ALLEN
20	SMITH
-	JONES
30	SMITH
-	ALLEN
-	ALLEN
40	JOHN KLINTON
-	ALEX BOUSH

## 6. Разворачивание результирующего множества для упрощения вычислений

### Задача

Требуется вычислить разность между заработными платами DEPTNO 30 и DEPTNO 10 и заработными платами DEPTNO 10 и DEPTNO 40.

Таким образом, требуется выполнить вычисления, в которых участвуют данные нескольких строк. Чтобы упростить задачу, эти строки надо развернуть и превратить в столбцы, так чтобы все необходимые значения располагались в одной строке.

### Решение

```
SELECT d30_sal - d10_sal AS d30_d10_diff,
       d10_sal - d40_sal AS d10_d40_diff
FROM (
  SELECT SUM(CASE WHEN deptno = 10 THEN salvalue END) AS d10_sal,
         SUM(CASE WHEN deptno = 30 THEN salvalue END) AS d30_sal,
         SUM(CASE WHEN deptno = 40 THEN salvalue END) AS d40_sal
  FROM emp JOIN salary USING(empno) JOIN career USING(empno) JOIN job USING(jobno) JOIN dept USING(deptno)
  ) totals_by_dept
```

D30_D10_DIFF	D10_D40_DIFF
74605	29850

### Обсуждение

С помощью выражения CASE развернуть таблицу и перенести заработные платы для каждого DEPTNO из строк в столбцы:



```
SELECT CASE WHEN deptno = 10 THEN salvalue END AS d10_sal,
       CASE WHEN deptno = 30 THEN salvalue END AS d30_sal,
       CASE WHEN deptno = 40 THEN salvalue END AS d40_sal
FROM emp JOIN salary USING(empno) JOIN career USING(empno) JOIN job USING(jobno) JOIN dept USING(deptno)
```

D10_SAL	D30_SAL	D40_SAL
-	3000	-
-	-	-
-	2580	-
-	-	-
-	2650	-
-	-	-
-	2510	-
-	-	-
-	2495	-
-	-	-
-	1750	-
-	-	-
-	3540	-
-	-	-
-	2580	-
-	-	-
-	2050	-
-	-	-
-	-	1850
-	-	1900
-	-	1950
-	-	1950
-	-	600
-	-	650
-	-	700
8050	-	-
-	8050	-
-	8050	-
8050	-	-
-	8050	-
-	8050	-
8150	-	-
-	8150	-
-	8150	-
-	3000	-
-	-	-
-	3000	-
-	-	-
-	3000	-
-	-	-
-	3000	-
-	-	-
3000	-	-

...

Суммируем все заработные платы для каждого DEPTNO, применяя агрегатную функцию SUM к каждому выражению CASE:

```
SELECT SUM(CASE WHEN deptno = 10 THEN salvalue END) AS d10_sal,
       SUM(CASE WHEN deptno = 30 THEN salvalue END) AS d30_sal,
       SUM(CASE WHEN deptno = 40 THEN salvalue END) AS d40_sal
FROM emp JOIN salary USING(empno) JOIN career USING(empno) JOIN job USING(jobno) JOIN dept USING(deptno)
```

D10_SAL	D30_SAL	D40_SAL
39450	114055	9600

Далее оформляем приведенный выше запрос как вложенный и находим разности.

### Задания

1. Развернуть группу строк информирующую о количестве сотрудников на каждой должности в настоящий момент времени, превращая их значения в столбцы.

Пусть имеется результирующее множество:

JOBNAME	EMPCOUNT
CLERK	1
FINANCIAL DIRECTOR	1
PRESIDENT	1
MANAGER	2
DRIVER	1

Требуется переформатировать результат так, чтобы множество имело вид:

PRESIDENT	FINANCIAL_DIRECTOR	MANAGER	CLERK	DRIVER
1	1	2	1	1

2. Требуется преобразовать строки в столбцы, создавая для каждого значения заданного столбца отдельный столбец.

Пусть имеется результирующее множество:

DEPTNAME	EMPNAME
SALES	SMITH
RESEARCH	SMITH
ACCOUNTING	ALLEN
SALES	ALLEN
SALES	ALLEN
RESEARCH	JONES
OPERATIONS	ALEX BOUSH
OPERATIONS	JOHN KLINTON

Требуется переформатировать результат так, чтобы множество имело вид:

ACCOUNTING	OPERATIONS	RESEARCH	SALES
ALLEN	ALEX BOUSH	JONES	ALLEN
-	JOHN KLINTON	SMITH	ALLEN
-	-	-	SMITH

3. Выполните обратное разворачивание для результирующего множества, полученного в задании 1:

PRESIDENT	FINANTIAL_DIRECTOR	MANAGER	CLERK	DRIVER
1	1	2	1	1

4. Составьте запрос, который будет выполнять обратное разворачивание результирующего множества в один столбец.
5. Составьте запрос, который будет исключать повторяющиеся значения из результирующего множества.