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

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:

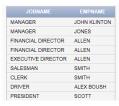


2. Разворачивание результирующего множества в несколько строк

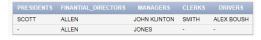
Задача

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:



Требуется переформатировать результат так, чтобы множество имело вид:



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 FINANTIAL_DIRECTORS,

MAX(CASE WHEN jobname='MANAGER' THEN empname ELSE NULL END) AS CLERKS,

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

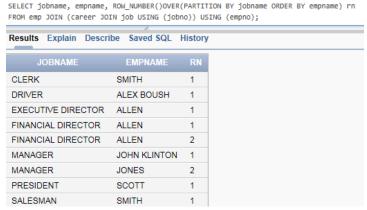
FROM emp JOIN (career JOIN job USING (jobno)) USING (empno)

)

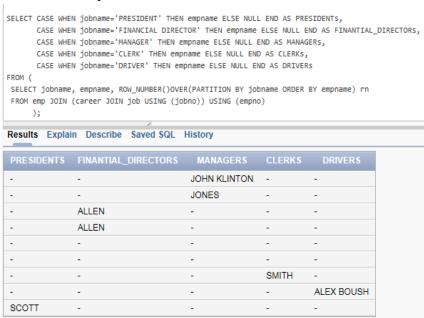
GROUP BY rn;
```

Обсуждение

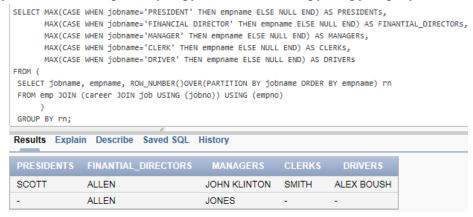
Чтобы решить данную задачу необходимо сделать каждое сочетание JOBNAME+EMPNAME уникальным. С этой целью используем ранжирующую функцию ROW_NUMBER OVER:



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



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



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,
           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
                    10
                    20
                    30
                             3
                    40
```

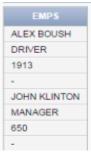
4. Обратное разворачивание результирующего множества в один столбец

Задача

Имеется представление EMPS_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:



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 emps_info e,
(SELECT *
FROM emps_info) four_rows
WHERE e.deptno = 40
)
```

Обсуждение

Используя ранжирующую функцию ROW_NUMBER OVER присваиваем ранг каждому сотруднику 40-го отдела:

| EMPNAME | JOBNAME | AVG_SALARY | |
|--------------|---------|------------|---|
| 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 результирующем множеством и некоторой таблицей, имеющей четыре строки (в данном примере в качестве такой таблицы выступает представление EMPS INFO:

CREATE OR REPLACE VIEW emps_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 emps_info e,

(SELECT *

FROM emps_info) four_rows

WHERE e.deptno = 40
```

| EMPNAME | JOBNAME | AVG_SALARY | |
|--------------|---------|------------|---|
| 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.

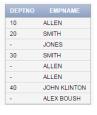
Теперь с помощью выражения CASE разместить значения EMPNAME, JOBNAME, AVG_SALARY всех сотрудников в один столбец (чтобы значения AVG_SALARY могли использоваться в CASE, их необходимо привести к строковому типу):

```
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 emps info e,
     (SELECT *
       FROM emps_info) four_rows
WHERE e.deptno = 40
                    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:



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)
```

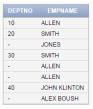
Обсуждение

С помощью оконной функции LAG OVER возвратить 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)
```

| | | 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):



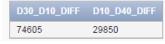
6. Разворачивание результирующего множества для упрощения вычислений

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



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)
```



. . .

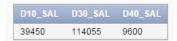
Суммируем все заработные платы для каждого DEPTNO, применяя агрегатную функцию SUM к каждому выражению CASE:

```
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)
```



Далее оформляем приведенный выше запрос как вложенный и находим разности.

<u>Задания</u>

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

Пусть имеется результирующее множество:

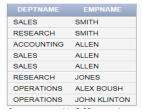
| JOBNAME | EMPCOUNT |
|--------------------|----------|
| CLERK | 1 |
| FINANCIAL DIRECTOR | 1 |
| PRESIDENT | 1 |
| MANAGER | 2 |
| DRIVER | 1 |

Требуется переформатировать результат так, чтобы множество имело вид:

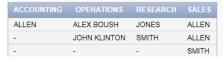


2. Требуется преобразовать строки в столбцы, создавая для каждого значения заданного столбца отдельный столбец.

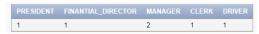
Пусть имеется результирующее множество:



Требуется переформатировать результат так, чтобы множество имело вид:



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