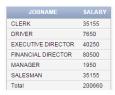
# Лабораторная работа №3. Составление отчетов. Вычисления.

## 1. Вычисление простых подсумм

Под «простой суммой» подразумевается результирующее множество строк, содержащее значения, полученные в результате агрегации одного столбца, и общий итог таблицы.

#### Задача

Требуется получить результирующее множество, содержащее суммы заработных плат таблицы EMP по должностям, а также сумму всех заработных плат таблицы EMP:



#### Решение

```
SELECT CASE GROUPING(jobname)

WHEN 0 THEN jobname

ELSE 'Total'

END jobname,

SUM(salvalue) salary

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

GROUP BY ROLLUP(jobname):
```

# Обсуждение

Используем агрегатную функцию SUM, чтобы просуммировать заработные платы; с помощью расширения ROLLUP оператора GROUP BY организовываем результаты в подсуммы по JOBNAME и находим общую сумму.

# 2. Вычисление подсумм для всех возможных сочетаний

#### Задача

Требуется найти суммы всех заработных плат по отделам, по должностям и для каждого сочетания DEPTNAME/JOBNAME. Должна быть также вычислена общая сумма всех заработных плат:



# Решение

```
SELECT deptname, jobname,

CASE GROUPING(deptname) || GROUPING(jobname)

WHEN '00' THEN 'TOTAL BY DEPT and JOB'

WHEN '10' THEN 'TOTAL BY JOB'

WHEN '01' THEN 'TOTAL BY DEPT'

WHEN '11' THEN 'GRAND TOTAL for TABLE'

END category,

SUM(salvalue) salary

FROM emp JOIN salary USING(empno) JOIN career USING(empno) JOIN job USING(jobno) JOIN dept USING(deptno)

GROUP BY CUBE(deptname, jobname)

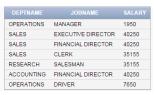
ORDER BY GROUPING(deptname,), GROUPING(jobname);
```

## Обсуждение

Используется расширение CUBE оператора GROUP BY в сочетании с оператором конкатенации ||.

Первый шаг — найти суммарные заработные платы для каждого сочетания DEPTNAME и JOBNAME, применяя агрегатную функцию SUM и группируя значения по DEPTNAME и JOBNAME:

SELECT deptname, jobname, SUM(salvalue) salary
FROM emp JOIN salary USING(empno) JOIN career USING(empno) JOIN job USING(jobno) JOIN dept USING(deptno)
GROUP BY deptname, jobname;



Следующий шаг — вычислить подсуммы по DEPTNAME и JOBNAME и общую сумму для всей таблицы. С помощью расширения CUBE оператора GROUP BY выполняем агрегацию значений SALARY по DEPTNAME, JOBNAME и затем для всей таблицы в следующем запросе (\*):

SELECT deptname, jobname, SUM(salvalue) salary
FROM emp JOIN salary USING(empno) JOIN career USING(empno) JOIN job USING(jobno) JOIN dept USING(deptno)
GROUP BY CUBE(deptname, jobname);



Далее используем функцию GROUPING в сочетании с выражением CASE, чтобы представить результаты в более выразительном формате. GROUPING(JOBNAME) возвращает значения 1 или 0 в зависимости от того, получены ли значения SALARY оператором GROUP BY или его расширением CUBE. Если значение возвращено CUBE, получаем 1, В противном случае 0. Аналогично GROUPING(DEPTNAME). Из первого шага решения видим, что группировка выполняется по DEPTNAME и JOBNAME. Таким образом, в результате вызова GROUPING для строки, представляющей сочетание DEPTNAME и JOBNAME, должен быть возвращен 0. Следующий запрос подтверждает это:

```
SELECT deptname, jobname,

GROUPING(deptname) dept_subtotal,

GROUPING(jobname) job_subtotal,

SUM(salvalue) salary

FROM emp JOIN salary USING(empno) JOIN career USING(empno) JOIN job USING(jobno) JOIN dept USING(deptno)

GROUP BY CUBE(deptname, jobname)

ORDER BY 3, 4;
```



Заключительный шаг — использовать выражение CASE для определения категории строки на основании значений, возвращаемых в результате конкатенации . GROUPING(JOBNAME) и GROUPING(DEPTNAME).

### 3. Выявление строк, в которых представлены не подсуммы

#### Задача

При создании отчета использовалось расширение CUBE оператора GROUP BY. Требуется отличить строки, сформированные оператором GROUP BY, и строки, являющиеся результатом выполнения CUBE или ROLLUP.

Имеется множество – результат работы запроса (\*):



#### Решение

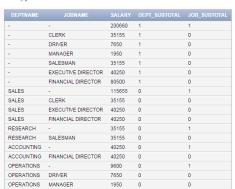
SELECT deptname, jobname, SUM(salvalue) salary,

GROUPING(deptname) dept\_subtotal,

GROUPING(jobname) job\_subtotal

FROM emp JOIN salary USING(empno) JOIN career USING(empno) JOIN job USING(jobno) JOIN dept USING(deptno)

GROUP BY CUBE(deptname, jobname);



Чтобы обозначить, какие из значений получены благодаря выполнению CUBE или ROLLUP, а какие из них являются *суперагрегатными*, используем функцию GROUPING.

## Обсуждение

Если DEPT\_SUBTOTAL равно 1, то значение поля SALARY представляет подсумму по DEPTNAME, созданную CUBE. Если JOB\_SUBTOTAL равно 1, то значение поля

SALARY представляет подсумму по JOBNAME, созданную СUBE. Если в обоих столбцах, DEPT\_SUBTOTAL и JOB\_SUBTOTAL, располагается 1, то значение поля SALARY представляет общую сумму всех заработных плат, сформированных СUBE. Строки, в которых оба значения, DEPT\_SUBTOTAL и JOB\_SUBTOTAL, равны 0, созданы в результате обычной агрегации (значение SALARY представляет сумму заработных плат для каждого сочетания DEPTNAME и JOBNAME).

# 4. Агрегация разных групп одновременно

#### Задача

Требуется выполнить агрегацию «в разных измерениях» одновременно. Например, необходимо получить результирующее множество, в котором для каждого сотрудника указаны имя, отдел, количество сотрудников в отделе (включая его самого), количество сотрудников, занимающих ту же должность, что и этот сотрудник (включая его самого), и общее число сотрудников в таблице:

EMPNAME	DEPTNAME	DEPTNAME_EMP_CNT	JOBNAME	JOBNAME_EMP_CNT	TOTAL
RICHARD MARTIN	ACCOUNTING	3	CLERK	2	11
ALLEN	ACCOUNTING	3	FINANCIAL DIRECTOR	2	11
RICHARD MARTIN	ACCOUNTING	3	SALESMAN	2	11
JOHN KLINTON	OPERATIONS	2	MANAGER	3	11
ALEX BOUSH	OPERATIONS	2	DRIVER	1	11
SMITH	RESEARCH	3	SALESMAN	2	11
JOHN MARTIN	RESEARCH	3	MANAGER	3	11
JONES	RESEARCH	3	MANAGER	3	11
ALLEN	SALES	3	EXECUTIVE DIRECTOR	1	11
ALLEN	SALES	3	FINANCIAL DIRECTOR	2	11
SMITH	SALES	3	CLERK	2	11

### Рекомендации

Используйте оконную функцию COUNT OVER, задавая разные группы данных, для которых проводится агрегация.

#### 5. Агрегация скользящего множества значений

# Задача

Требуется выполнить скользящую агрегацию, например, найти скользящую сумму заработных плат. Вычислять сумму для каждого интервала в 90 день, начиная с даты приема на работу (поле STARTDATE таблицы CAREER) первого сотрудника, чтобы увидеть динамику изменения расходов для каждого 90-дневного периода между датами приема на работу первого и последнего сотрудника:

		SPENDING_PATTERN
01/Jul/2000	2580	35155
01/Jul/2000	3000	35155
01/Jul/2000	2495	35155
01/Jul/2000	2050	35155
01/Jul/2000	1750	35155
01/Jul/2000	2580	35155
01/Jul/2000	2510	35155
01/Jul/2000	3540	35155
01/Jul/2000	2650	35155
01/Jul/2000	3000	35155
01/Jan/2001	1850	7650
01/Jan/2001	1950	7650
01/Jan/2001	1900	7650
01/lan/2001	1050	7650

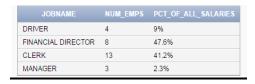
#### Рекомендации

Используйте функции SUM OVER, оператор ORDER BY и RANGE BETWEEN 90 PRECEDING AND CURRENT ROW.

## 6. Определение доли от целого в процентном выражении

# Задача

Требуется вывести множество числовых значений, представив каждое из них как долю от целого в процентном выражении. Например, требуется получить результирующее множество, отражающее распределение заработных плат по должностям, чтобы можно было увидеть, какая из позиций ЈОВ обходится компании дороже всего:



#### Рекомендации

Используйте оконную функцию COUNT OVER и RATIO\_TO\_REPORT.

# Задания

1. Получить результирующее множество, содержащее количество сотрудников в каждом отделе, а также общее количество сотрудников.



- 2. Требуется найти количество сотрудников по отделам, по должностям и для каждого сочетания DEPTNAME/JOBNAME.
- 3. Требуется найти среднее значение суммы всех заработных плат по отделам, по должностям и для каждого сочетания DEPTNAME/JOBNAME.
- 4. Создайте запрос на распознавание строк, сформированных оператором GROUP BY, и строк, являющихся результатом выполнения CUBE.
- 5. Создайте запросы по заданиям пунктов 4-6.