Лабораторная №2

Запросы с группировкой данных. Условия для групп.

Предложение GROUP BY оператора SELECT позволяет объединить строки получающейся в результате запроса таблицы в группы и выдать для каждой полученной группы обобщающую информацию.

Группа формируется путем объединения всех строк, для которых столбец (или группа столбцов), указанный в предложении GROUP BY, имеет одинаковое значение. Столбец или группа столбцов, указанные в предложении GROUP BY, называются элементами группировки.

На запросы, в которых используется группировка, накладываются дополнительные ограничения:

- в качестве элементов группировки могут быть использованы только столбцы таблицы или выражения, вычисляемые на основе значений в столбцах таблицы;
- столбцами таблицы, формируемой оператором SELECT, могут быть или элементы группировки, или константы, или выражения с использованием агрегатных функций для столбцов, не являющихся элементами группировки;
- обратите внимание, что при использовании группировки каждая группа уникальна, следовательно, дубликаты невозможны и использование distinct бессмысленно.

Пример 1. Для каждого города, в котором расположены отделения фирмы определить суммарный бюджет этих отделений.

```
SELECT LOCATION, SUM(BUDGET) AS TOTAL_BUDGET
       FROM DEPARTMENT
       GROUP BY LOCATION
```

Пример 2. Определить суммарный бюджет каждого проекта в 1994 году.

```
SELECT PROJ_ID, SUM (PROJECTED_BUDGET) AS TOTAL_BUDGET
     FROM PROJ_DEPT_BUDGET
```

```
WHERE FISCAL_YEAR=1994
GROUP BY PROJ_ID;
```

Агрегатными функциями являются `SUM()`, `MIN()`, `MAX()` и `AVG()`. В качестве аргумента в этих функциях может быть использован столбец или выражение, содержащее столбец, не являющийся элементом группировки. Функции `SUM()` и `AVG()` применимы только к столбцам, содержащим числовые данные. Функции `MIN()` и `MAX()`, кроме числовых значений могут обрабатывать данные типа «дата–время». При вычислении значений агрегатных функций отсутствующие значения (`null`–значения) игнорируются.

Агрегатная функция `LIST()` строит из значений поля-параметра функции список. Может использоваться для проверки или передачи набора значений в виде списка.

Пример 3. Определить средний бюджет каждого проекта в 1995 году, указав количество отделов, принимающих участие в проекте.

```
SELECT PROJ_ID,
     AVG (PROJECTED_BUDGET) AS TOTAL_BUDGET,
     COUNT (DEPT_NO) AS NUM_DEPARTMENTS
     FROM PROJ_DEPT_BUDGET
     WHERE FISCAL_YEAR=1995
     GROUP BY PROJ_ID;
```

Для проверки можно добавить в запрос поле – список отделов, которые попадают в каждую из групп.

```
SELECT PROJ_ID,
     AVG (PROJECTED_BUDGET) AS TOTAL_BUDGET,
     COUNT (DEPT_NO) AS NUM_DEPARTMENTS,
     LIST (DEPT_NO) AS LIST_DEPT
         FROM PROJ_DEPT_BUDGET
         WHERE FISCAL_YEAR=1995
         GROUP BY PROJ_ID;
```

Функция COUNT() в группирующих запросах также ведет себя как агрегатная функция. В случае использования COUNT(*) функция возвращает количество записей в группе. Если в качестве параметра функции COUNT используется имя столбца, не являющегося элементом группировки, функция вернет количество непустых значений в этом столбце для каждой из групп. Наконец, если совместно с указание имени столбца использовать DISTINCT, функция COUNT() вернет количество различных значений в группе для указанного столбца.

Пример 4. Для каждого кода работы определить в скольких странах предлагаются вакансии по этому виду работы.

```
SELECT JOB_CODE, COUNT (DISTINCT JOB_COUNTRY)
     FROM JOB
     GROUP BY JOB_CODE;
```

Проверьте, что функция COUNT дает другой результат, если убрать в аргументе условие DISTINCT.

Пример 5. Для каждого отдела определить, сколько отделений подчиняется ему непосредственно.

Подчинение отдела вышестоящему определяется полем HEAD_DEPT, именно по этому полю нужно выполнить группировку

```
SELECT   DP.HEAD_DEPT, COUNT(*)
       FROM DEPARTMENT DP
       GROUP BY DP.HEAD_DEPT;
```

Пример 6. Для каждого города определить количество отделений фирмы, расположенных в этом городе, количество отделений, у которых есть вышестоящее отделение и количество этих вышестоящих отделений

```
SELECT LOCATION,
            COUNT(*),
            COUNT(HEAD_DEPT),
            COUNT(DISTINCT HEAD_DEPT)
FROM DEPARTMENT
GROUP BY LOCATION
```

При использовании группировки можно наложить дополнительное условие на результирующую таблицу, какие группы включать. Это условие задается предложением HAVING. Условие проверяется после того, как сформированы группы. Его рекомендуется применять, если условие отбора задается с помощью агрегатной функции.

Пример 7. Определить коды проектов, суммарный бюджет которых в 1994 году превысил 100000.

```
SELECT PROJ_ID, SUM (PROJECTED_BUDGET) AS TOTAL_BUDGET
     FROM PROJ_DEPT_BUDGET
     WHERE FISCAL_YEAR=1994
     GROUP BY PROJ_ID
   HAVING SUM (PROJECTED_BUDGET)>100000;
```

Условие группировки может содержать вычислимое выражение. Чтобы не повторять это выражение в списке SELECT и списке GROUP BY можно использовать псевдоним столбца.

Пример 8. Для каждого года определить, сколько сотрудников было принято в этом году на работу в фирму.

```
SELECT  EXTRACT(YEAR FROM E.HIRE_DATE) AS YH, COUNT(*)
     FROM EMPLOYEE E
     GROUP BY YH
```

Список группировки может содержать любое количество элементов группировки.

Пример 9. Для каждого отдела и каждого года определить количество принятых на работу сотрудников.

```
SELECT E.DEPT_NO,
       EXTRACT(YEAR FROM E.HIRE_DATE) AS YH,
       COUNT(*)
     ROM EMPLOYEE E
     GROUP BY E.DEPT_NO, YH
```

Задачи для самостоятельного выполнения

1. Для каждой страны определить количество заказчиков, находящихся в этой стране.
2. Для каждого года и каждой должности определить, сколько сотрудников было принято на работу в фирму на эту должность и в этом году.
3. Для каждого года определить, сколько инженеров было принято на работу в этом году.
4. Вывести суммарный бюджет всех проектов для каждого отдела за конкретный год.
5. Для каждого отдела вычислить суммарную и среднюю зарплату сотрудников.
6. Сгруппировать список сотрудников по первой букве (функция `SUBSTRING()`) в поле FIRST_NAME и указать для каждой буквы количество сотрудников.
7. Найти отделы, в которых работает не больше чем 2 сотрудника. Для проверки сформировать список этих сотрудников.

В документе «Использование выражений в запросах» приводится краткий обзор некоторых операторов, предикатов и функций, позволяющих выполнять обработку информации непосредственно в запросе. Рассмотрите приводимые там примеры использования этих элементов SQL самостоятельно. Более подробную информацию можно найти в официальной документации по Firebird.