Лабораторная №6

Подзапросы с предикатами

В θ–условиях WHERE и HAVING при сравнении на равенство/неравенство можно использовать подзапрос, если он скалярный, т.е. возвращает таблицу из одного столбца и одной строки. Нескалярные подзапросы можно использовать только с помощью предикатов.

Предикат IN/NOT IN проверяет, присутствует ли значение в указанном списке. Список может быть составлен явно из константных значений или быть результатом выполнения подзапроса, возвращающего таблицу из одного столбца.

Пример 1. Получить список заказчиков, с которыми был оформлен договор 18 декабря 1993 года.

```
SELECT CUSTOMER
     FROM CUSTOMER
     WHERE CUST_NO IN (SELECT CUST_NO
                              FROM SALES
                              WHERE ORDER_DATE='1993-dec-18');
```

Пример 2. Выдать список служащих отделов, у которых нет менеджера.

```
SELECT FULL_NAME, DEPT_NO
     FROM EMPLOYEE
     WHERE DEPT_NO IN (SELECT DEPT_NO
                          FROM DEPARTMENT
                          WHERE MNGR_NO IS NULL);
```

Предикат ALL проверяет, является ли сравнение (обычно в виде неравенства) истинным для всех значений, возвращаемых подзапросом.

Пример 3. Найти проекты, годовой бюджет которых превышает бюджет любого отдела.

```
SELECT PROJ_ID, FISCAL_YEAR ,SUM(PROJECTED_BUDGET)
     FROM PROJ_DEPT_BUDGET
     GROUP BY  PROJ_ID, FISCAL_YEAR
     HAVING SUM (PROJECTED_BUDGET)>
               ALL(SELECT BUDGET FROM DEPARTMENT);
```

Предикат ANY (SOME) проверяет, является ли сравнение истинным, по крайней мере, для одного значения, возвращаемого подзапросом. Применяются при сравнении или при проверке условий с помощью LIKE.

Пример 4. Выдать список всех служащих, у которых было, по крайней мере, одно изменение оклада в течение года после приема на работу.

```
SELECT E.EMP_NO, E.FULL_NAME, E.HIRE_DATE
     FROM EMPLOYEE E
     WHERE E.HIRE_DATE+365>
          ANY(SELECT SH.CHANGE_DATE
                    FROM SALARY_HISTORY SH
                    WHERE SH.EMP_NO = E.EMP_NO);
```

Предикат EXISTS (NOT EXISTS) определяет, существует или нет, по крайней мере, одно значение в выходном результате подзапроса. В большинстве случаев подзапросы в предикате EXISTS являются коррелированными, то есть условия поиска подзапроса связаны с одним или более столбцами основного запроса. Так как смысл этого предиката – проверить существование кортежей в подзапросе, сам подзапрос может иметь любой список возвращаемых столбцов, в частности это может быть столбец, содержащий любую константу.

Пример 5. Найти сотрудников отдела 130, которые никогда не оформляли заказы.

```
SELECT FULL_NAME FROM EMPLOYEE e
WHERE DEPT_NO ='130'
     AND NOT EXISTS (SELECT * FROM SALES s
                     WHERE s.SALES_REP=e.EMP_NO);
```

Эту же задачу можно решить, используя предикат NOT IN

```
SELECT FULL_NAME FROM EMPLOYEE E
WHERE DEPT_NO ='130'
    AND E.EMP_NO NOT IN(SELECT S.SALES_REP FROM SALES S);
```

Пример 6. Выдать список сотрудников, которые являются руководителями проектов.

```
SELECT A.FULL_NAME
     FROM EMPLOYEE A
     WHERE EXISTS(SELECT 1 FROM PROJECT P
                  WHERE P.TEAM_LEADER = A.EMP_NO);
```

Пример 7. Найти сотрудников, которые оформили только один заказ.

```
SELECT FULL_NAME, E.EMP_NO
     FROM EMPLOYEE E
     WHERE EXISTS (SELECT 1 FROM SALES S
                         WHERE S.SALES_REP=E.EMP_NO
                         GROUP BY S.SALES_REP
                         HAVING COUNT(*)=1);
```

Общие табличные выражения (CTE)

В лабораторной работе 5 была рассмотрена возможность описывать с помощью подзапроса временную таблицу, к которой потом формируется запрос. Такой вариант синтаксиса как бы «заворачивает» описание этой временной таблицы. Как альтернатива вводится определение временной таблицы через механизм «общих табличных выражений». Такая таблица описывается перед собственно запросом.

Пример 7. Найти сотрудников из USA, у которых зарплата выше средней, указать для них название отдела и город.

```
WITH EMP_DEPT_USA (NAME, DEPT_NAME, TOWN, SALARY)
AS
(SELECT FULL_NAME, DEPARTMENT, LOCATION, SALARY
     FROM EMPLOYEE A JOIN DEPARTMENT D ON A.DEPT_NO=D.DEPT_NO
     WHERE JOB_COUNTRY ='USA')
SELECT NAME, DEPT_NAME, SALARY, TOWN
     FROM EMP_DEPT_USA
     WHERE EMP_DEPT_USA.SALARY >
          (SELECT AVG(SALARY) FROM EMP_DEPT_USA );
```

Пример 8. Для каждого из отделов, участвующих в проектах провести сравнение суммарного бюджета по проектам в 1994 и 1995 годах.

```
WITH
DEPT_YEAR_BUDGET AS (
--ТАБЛИЦА СУММАРНОГО БЮДЖЕТА ПО ОТДЕЛАМ/ГОДАМ
     SELECT FISCAL_YEAR,DEPT_NO,
          SUM(PROJECTED_BUDGET) BUDGET
     FROM PROJ_DEPT_BUDGET
     GROUP BY FISCAL_YEAR, DEPT_NO
)
SELECT D.DEPT_NO, D.DEPARTMENT,
     B_1994.BUDGET BUDGET_94,B_1995.BUDGET AS BUDGET_95
FROM
DEPARTMENT D
     LEFT JOIN DEPT_YEAR_BUDGET B_1994
     ON D.DEPT_NO = B_1994.DEPT_NO AND B_1994.FISCAL_YEAR = 1994
     LEFT JOIN DEPT_YEAR_BUDGET B_1995
     ON D.DEPT_NO = B_1995.DEPT_NO AND B_1995.FISCAL_YEAR = 1995
WHERE EXISTS (SELECT * FROM PROJ_DEPT_BUDGET B
     WHERE D.DEPT_NO = B.DEPT_NO);
```

Здесь табличное выражение используется, чтобы сформировать временную таблицу, которая будет использоваться в запросе дважды для выборки строк за 1994 год и за 1995 год. Обратите внимание, что условие

выбора года включено в условие соединения `LEFT JOIN`. А в `WHERE` используется предикат для того, чтобы выдать только те отделы, которые участвуют в проектах.

Задачи для самостоятельного выполнения

Решить следующие задачи (база данных для задач –**wh_new.fdb**), **используя подзапросы**. При формировании основного запроса и подзапроса может быть использована операция join. Для некоторых задач можно привести несколько вариантов решения.

1. Найти поставщиков, которые работали со складами в городе «Ростов».
2. Найти склады, на которых размещаются товары в упаковках.
3. Найти товары, с которыми не было ни одной операции.
4. Найти поставщиков, которые выполнили только одну поставку.
5. Найти поставщиков, которые поставляли (операции А) карандаши по минимальной цене.
6. Найти склады, с которыми не было ни одной операции.
7. Найти поставщиков, которые работают более чем с одним складом.
8. Найти товары, для которых была выполнена только одна поставка (А).
9. Найти товары, которые поставщик «Надежный» поставлял по наибольшей цене.
10. Составить список складов, на которых нет остатков товаров с указанием города нахождения склада.
11. Найти товары, с которыми было больше всего операций.
12. Для каждого склада вывести для сравнения количество операций в 2006 и 2007 году.