Лабораторная №9

«Процедуры, возвращающие таблицы»

В лабораторной работе рассматриваются возможности процедур, возвращающих результат в виде таблицы. Используется возможность выполнения отдельных операций в процедуре в автономной транзакции. Продемонстрировано использование SQL-операций, подготовленных в виде строки. Все задачи выполняются в созданной студентами личной копии базы данных «Склад».

ПРИМЕР 1.

Написать процедуру, которая для заданного поставщика выдает список товаров и количество операций, выполненных поставщиком с этими товарами.

ВНИМАНИЕ!!!

Для процедур, возвращающих результат в виде таблицы, должны быть определены выходные параметры. Выдача одной сформированной строки результирующей таблицы производится оператором SUSPEND.

Текст процедуры

```
create or alter procedure GOODSBYAGENT (
    AGENT type of column AGENT.NAME_AG)
returns (GOODS type of column GOODS.NOMENCLATURE,
         CNT integer)
as
declare variable ID type of column AGENT.ID_AG;
begin
  id = (select a.id_ag from agent a where a.name_ag=:agent);
  if (id is null) then exception error ;
  for select g.nomenclature,  count(o.id)
          from  operation O join goods  G using(id_goods)
               where O.id_ag = :id
          group by  G.nomenclature
     into :goods, :cnt do
     suspend;
end
```

В предыдущей лабораторной работе для сигнализации об ошибке использовалось безымянное исключение. Если таких ситуаций много и нужно их как-то различать, можно создать исключение как объект схемы базы данных. При этом каждое исключение имеет уникальное имя и сообщение-комментарий. Для сигнализации об ошибке в параметрах процедуры (неправильное имя агента) использовано исключение, которое можно создать следующей командой SQL (или воспользоваться возможностями редактора IBExpert)

```
CREATE EXCEPTION ERROR 'Ошибка в процедуре';
```

После компиляции процедуру можно выполнить прямо в окне создания процедуры (F9). Но поскольку процедура такого вида возвращает таблицу, ее можно использовать в операторе select:

Например, так:

```
select *
     from goodsbyagent('Надежный')
```

или

```
select A.goods,  G.measure,  A.cnt
     from goodsbyagent('Надежный')  A
     join goods G on A.goods=G.nomenclature
```

Задание 1.

В последнем запросе, чтобы подключить дополнительную информацию из таблицы Goods пришлось выполнять join по наименованию товара. Это не всегда является эффективным решением. Если предполагается использовать таблицу, возвращаемую процедурой в соединении с другими таблицами, лучше предусмотреть в ней дополнительные выходные параметры, подходящие для этого, например, код товара (id_goods).

Измените процедуру, так, чтобы она возвращала еще один параметр.

ПРИМЕР 2.

Изменить процедуру в примере 1 так, чтобы в случае, когда выбрасывается исключение, дополнительно шла запись об ошибке в log_file.

Обратите внимание, что если процедура завершается выбрасыванием исключения, все действия, произведенные в рамках этой процедуры, будут аннулированы (транзакция завершится откатом – Rollback). Поэтому попытка просто сделать операцию insert в таблицу log_file не даст результата. Чтобы решить эту проблему, операцию insert нужно выполнить в автономной транзакции, которая завершится Commit, независимо от того, как завершается процедура.

Текст измененной процедуры

```
create or alter procedure GOODSBYAGENT (
    AGENT type of column AGENT.NAME_AG)
returns (
    GOODS type of column GOODS.NOMENCLATURE,
    CNT integer)
as
declare variable ID type of column AGENT.ID_AG;
begin
  id = (select a.id_ag from agent a where a.name_ag=:agent);
  if (id is null) then begin
    in autonomous transaction do
        insert into log_file (inform)
          values ('в БД нет агента '||:agent);
    exception error ;
  end
  for select g.nomenclature, count(o.id)
     from  operation O join goods  G using(id_goods)
          where O.id_ag = id
     group by  G.nomenclature
     into :goods, :cnt do
        suspend;
end
```

Задание 2. Проверьте, что будет происходить, если убрать условие выполнения команды insert в автономной транзакции.

ПРИМЕР 3.

Написать процедуру, которая по заданному имени таблицы возвращает количество строк в этой таблице.

В этой задаче параметр не является значением, которое можно сопоставить с атрибутами таблицы. Поэтому использовать его напрямую в запросе нельзя. Построим текст запроса в виде строковой переменной.

Такую переменную можно использовать в операциях

```
[for] execute statement <выражение в виде строки>
   into <список переменных>
[do <оператор>]
```

Текст процедуры

```
create or alter procedure TAB_RANG (
    NAME_T char(31))
returns (
    KOL integer)
as
declare variable OPER varchar(200);
begin
OPER = 'select count(*) from  ' || name_t;
 execute statement OPER into :KOL ;
suspend;
end
```

Задание 3.Проверьте, как работает процедура, если ввести неправильное имя таблицы.

Задачи для самостоятельного выполнения

Описать селективные процедуры для решения следующих задач. Привести примеры использования созданных селективных процедур в запросах.

Во всех заданиях предусмотреть возможность некорректных значений входных параметров. В этом случае процедура должна выбрасывать исключение, а в таблице log_file должна быть запись о произошедшей ошибке.

1. По названию поставщика выдать все товары, которые он поставлял (операция R) с указанием для каждого товара даты последней операции (R).
2. По паре дат выдать наименования товаров, поставленных за заданный период, с указанием максимальной и минимальной стоимости операции с этим товаром.
3. По названию склада и периоду выдать список поставщиков, поставлявших (операция R) товары на заданный склад в заданный период.
4. Выдать названия складов, для которых количество указанного товара меньше заданного (параметры – название товара и предельное количество).
5. Выдать названия поставщиков, у которых суммарный объем операций с заданным товаром больше заданного значения (параметры – название товара и предельное количество).
6. Выдать список операций за заданный период с указанием названия товара, склада, поставщика. Указать стоимость каждой поставки (цена* количество).
7. Для заданных названия таблицы (TNAME) и названия столбца (ST_NAME) выдать количество записей в таблице TNAME, для которых в этом столбце ST_NAME отсутствует значение (null).