

## Лабораторная работа

### «Процедуры, возвращающие таблицы»

В лабораторной работе рассматриваются возможности процедур, возвращающих результат в виде таблицы. Рассматривается возможность выполнения отдельных операций в процедуре в автономной транзакции. Рассматривается возможность использования 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.

#### **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
```

Для сигнализации об ошибке в параметрах процедуры (неправильное имя агента) использовано исключение, которое можно создать следующей командой SQL (или воспользоваться возможностями редактора IVExpert)

```
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 удачным 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 выдавала 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  
    /* Procedure Text */  
    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.

#### 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.