Лабораторная работа №8

«Процедуры»

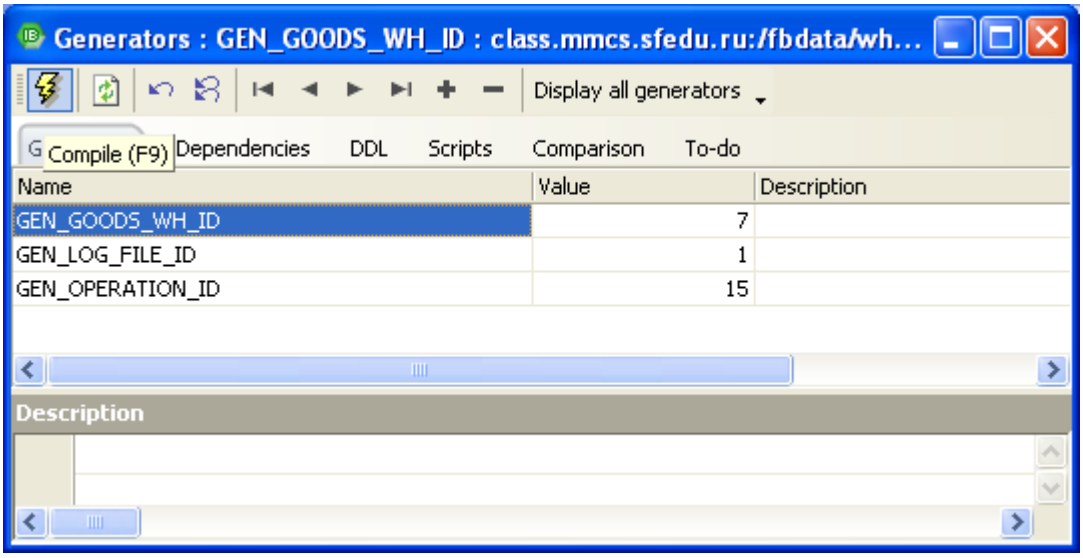
В лабораторной работе рассматриваются возможности процедур, не возвращающих результат. Эти процедуры вносят изменения в хранящуюся в базе данных информацию. Предполагается одновременное изучение возможностей операций INSERT, UPDATE и DELETE.

Все задачи выполняются в созданной студентами личной копии базы данных «Склад».

ВНИМАНИЕ!!! Для корректной работы создавать копию базы данных необходимо ТОЛЬКО в кодировке UTF-8. А при регистрации соединения с базой данных из клиента IBExpert указывать кодировку WIN1251.

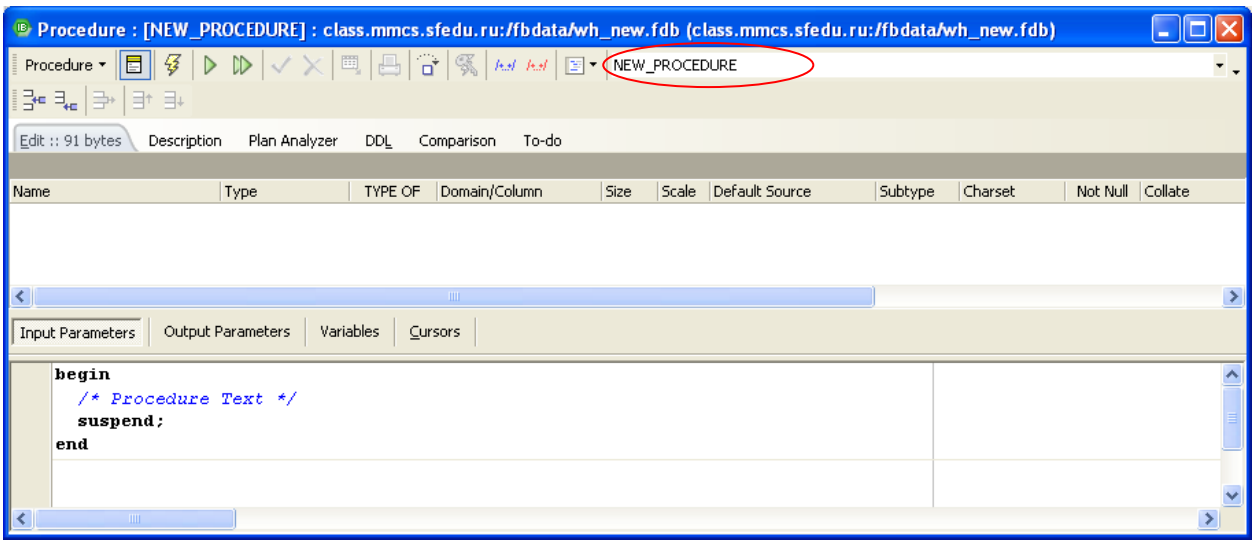
После создания копии необходимо проверить значения генераторов для автоматического формирования поля ID в таблицах GOODS_WH, LOG_FILE и OPERATION.

Генераторы должны иметь следующие значения:



Если необходимо, измените значения генераторов и выполните компиляцию (F9).

Для создания, компиляции и отладки процедур используется визуальный компонент IBExpert. Обратите внимание на необходимость сразу задать имя процедуре, заменив имя по умолчанию NEW_PROCEDURE на любое другое.



Окно визуального компонента для создания процедуры разделено на два фрейма – в верхнем добавляются локальные переменные: входные параметры – на вкладке Input Parameters, выходные параметры – на вкладке Output Parameters, вспомогательные переменные – Variables и курсоры – Cursors. Добавлять/удалять переменные можно через контекстное меню, вызываемое в верхнем фрейме правой кнопкой мыши.

Тело процедуры вводится в нижнем фрейме – там, где указано:

/* Procedure Text */

Оператор suspend для процедур, не возвращающих результат, можно оставить. Просмотреть текст процедуры без разбиения на фреймы можно переключая режим «Lazy on/off».

В режиме разделения на фреймы можно перейти на вкладку DDL и просмотреть полный скрипт создания процедуры, готовы для выполнения/экспорта.

После создания текста процедуры необходимо выполнить ее компиляцию (Ctrl+F9 или кнопка «молния»). Если компиляция прошла успешно – завершить создание процедуры операцией COMMIT. Откомпилированную процедуру можно выполнить здесь же (F9) используя оконный интерфейс для ввода параметров процедуры или выполнить в SQL редакторе команду execute procedure и указать имя процедуры и список параметров.

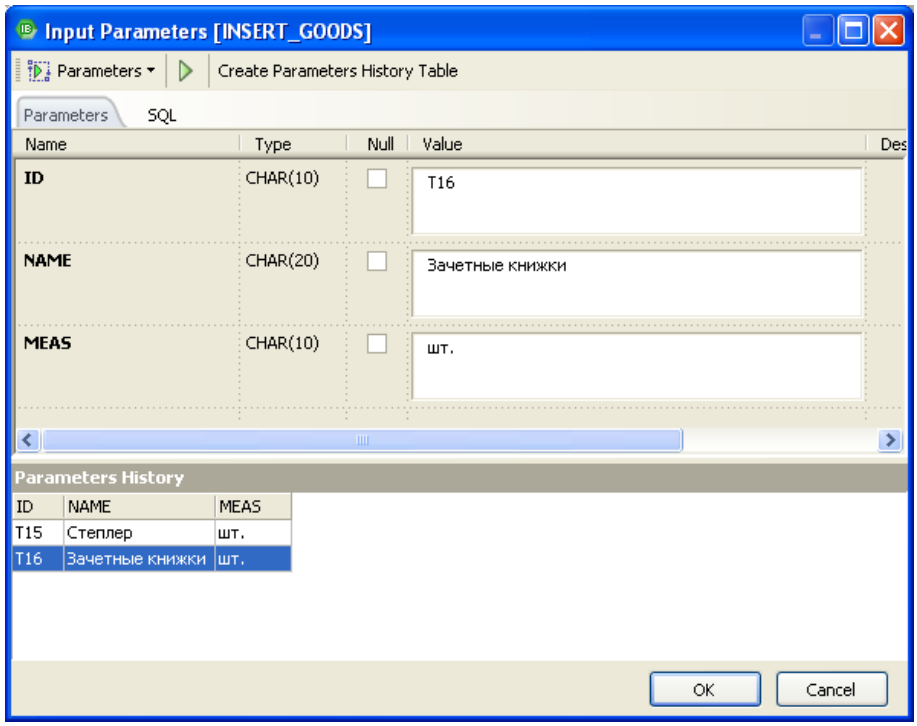
ПРИМЕР 1.

Написать процедуру для добавления нового товара в таблицу Goods. Параметры процедуры – код товара (id_goods), номенклатура и единица измерения. Чтобы не возникало ошибок в типах параметров, рекомендуется использовать типы столбцов соответствующей таблицы

Текс процедуры

```
create or alter procedure INSERT_GOODS (
    ID type of column GOODS.ID_GOODS,
    NAME type of column GOODS.NOMENCLATURE,
    MEAS type of column GOODS.MEASURE)
as
begin
  /* Procedure Text */
  insert into goods
    (id_goods, nomenclature, measure)
        values (:id, :name, :meas);
  suspend;
end
```

Проверка через окно ввода параметров



Поле id_goods в таблице GOODS является первичным ключом, поэтому при выполнении необходимо задавать для него различные значения (для поля nomenclature тоже рекомендуется использовать разные значения параметра). Чтобы увидеть результат работы процедуры в таблице необходимо завершить транзакцию операцией COMMIT (кнопка с зеленой галочкой). Если необходимо отменить все результаты – ROLLBACK (кнопка с красным крестиком). Не забывайте выполнять завершение транзакции и в окне для просмотра содержимого таблицы!!!

Команда для выполнения процедуры в SQL-редакторе

```
EXECUTE PROCEDURE
INSERT_GOODS('T10', 'Ручка', 'шт.');
```

Задание 1. Создайте процедуры для добавления новых данных в таблицы AGENT и WAREHOUSE.

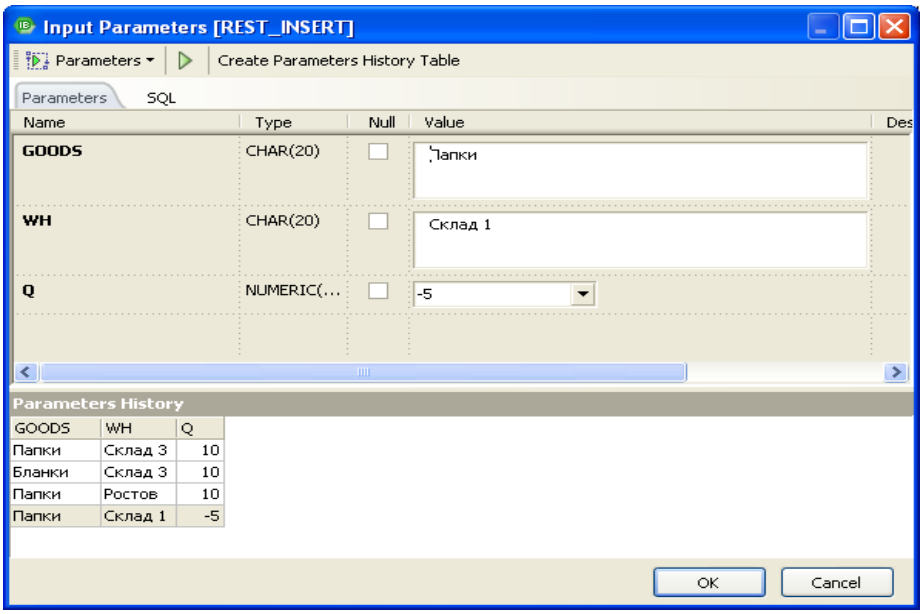
ПРИМЕР 2.

Написать процедуру для добавления остатка товара на склад. Параметры процедуры – название товара, название склада, количество. В значениях параметров могут быть ошибки. В этом случае процедура не должна завершаться аварийно, но должна поместить сообщение об ошибке в таблицу LOG_FILE .

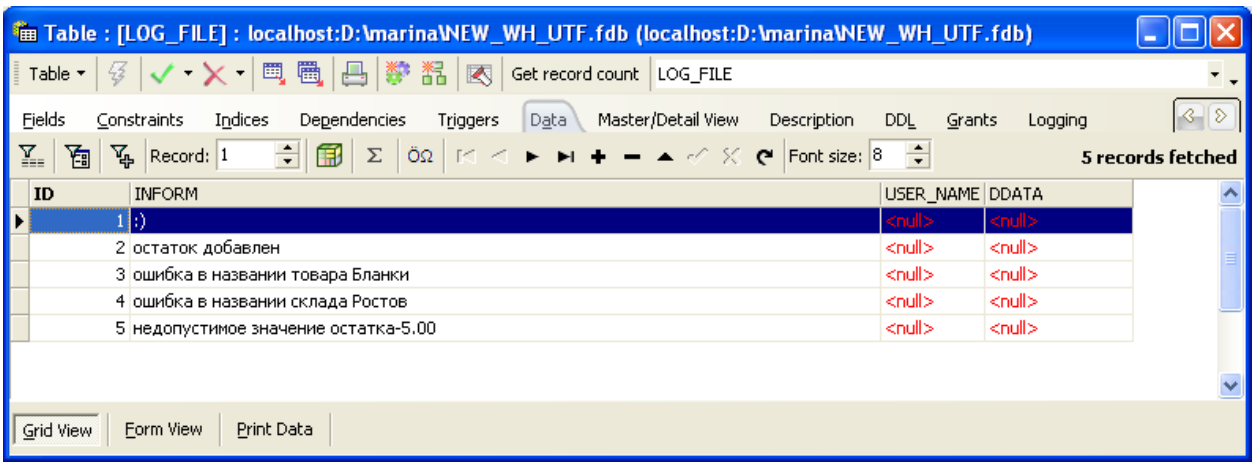
```
create or alter procedure REST_INSERT (
          GOODS type of column GOODS.NOMENCLATURE,
          WH type of column WAREHOUSE.NAME,
          Q type of column GOODS_WH.QUANTITY)
as
      declare variable IDG type of column GOODS.ID_GOODS;
      declare variable IDW type of column WAREHOUSE.ID_WH;
begin
        /* Procedure Text */
idg = (select A.id_goods
     from goods A where A.nomenclature = :goods);
if (idg is null ) then
        begin
```

```
          insert into log_file (inform)
          values ('ошибка в названии товара '||:goods);
           exit;
        end
idw = (select W.id_wh   from warehouse W where W.name=:wh);
if (idw is null ) then
        begin
          insert into log_file (inform)
          values ('ошибка в названии склада '||:wh);
           exit;
        end
if (q<=0) then
          begin
           insert into log_file (inform)
               values ('недопустимое значение остатка'||:q);
           exit;
        end
insert into goods_wh values (null, :idw, :idg, :q);
insert into log_file (inform) values ('остаток добавлен');
end
```

Примеры выполнения процедуры



Результаты выполнения в таблице LOG_FILE



В случае, если бы не было необходимости в проверке входных параметров (контроль выполнялся на стороне клиента), можно было бы упростить процедуру, используя подзапросы:

```
create or alter procedure REST1 (
    GOODS type of column GOODS.NOMENCLATURE,
    WH type of column WAREHOUSE.NAME,
    Q type of column GOODS_WH.QUANTITY)
as
begin
  /* Procedure Text */
  insert  into goods_wh
    values (null,
     (select id_wh from warehouse W where W.name=:wh),
     (select A.id_goods from goods A
          where A.nomenclature = :goods), :q);
  insert into log_file (inform) values ('остаток добавлен');
end
```

Задание 2. Создайте процедуру изменения остатка на складе. Параметры – название товара, название склада, новое количество. Предположить, что параметры заданы правильно.

Задание 3. Создайте процедуру удаления остатка со склада (должна быть удалена запись из таблицы `GOODS_WH`). Параметры – название товара, название склада. Предположить, что параметры могут быть заданы неправильно.

Задание 4. Изменить процедуру `REST_INSERT` так, чтобы дополнительно контролировалось условие – комбинация `id_goods`, `id_wh` в таблице `GOODS_WH` должна быть уникальной.

ПРИМЕР 3.

Написать процедуру, которая для заданного склада определяет количество разных наименований товара, находящихся на этом складе и общее количество товаров на нем. Предположить, что параметр – название склада задается правильно. Полученные значения записать в таблицу LOG_FILE.

```
create or alter procedure REPORT (
    WH type of column WAREHOUSE.NAME)
as
declare variable CNT integer;
declare variable FULL_QT type of column GOODS_WH.QUANTITY;
begin
  /* Procedure Text */
  select count(distinct A.ID_GOODS), sum(A.QUANTITY)
  from GOODS_WH A
  join WAREHOUSE B using (ID_WH)
  where B.NAME = :WH
  into :CNT, :FULL_QT;
  insert into log_file (inform)
     values ( ' типов товара '||:cnt||
              ' всего товаров '||:full_qt);
  suspend;
end
```

ПРИМЕР 4.

Написать процедуру, которая находит товары, которые поставлялись в максимальном количестве (имеется в виду, что с товаром была выполнена одна любая операция A или R). Записать эти товары в таблицу LOG_FILE.

```
create or alter procedure MAX_QT
as
declare variable GOODS type of column GOODS.NOMENCLATURE;
begin
```

```
  /* Procedure Text */
   for select  G.nomenclature
   from operation O join goods G using(id_goods)
     where O.quantity = (select max(A.quantity) from operation A)
          into  :goods
     do
   insert into log_file(inform) values(:goods);

end
```

Если предположить, что список таких товаров будет не слишком большим, можно сделать так:

```
create or alter procedure MAX_QT1
as
declare variable LIST_NOMEN varchar(200);
begin
  /* Procedure Text */
  select  list( G.nomenclature)
   from operation O join goods G using(id_goods)
   where O.quantity = (select max(A.quantity) from operation A)
   into  :list_nomen ;
   insert into log_file(inform) values(:list_nomen);
  suspend;
end
```

Задание 5. Создайте процедуру, которая для заданного (в параметре) наименования товара находила поставщиков, выполнивших максимальное количество операций с этим товаром. Список поставщиков поместить в таблицу LOG_FILE.

Задание 6. Создайте процедуру, которая записывает в LOG_FILE в поле DDATA даты всех операций A, выполненных заданным (в параметре) поставщиком.

ПРИМЕР 5. Изменить процедуру `REPORT` из Примера 3 так, чтобы в случае ошибочного параметра она возвращала признак – исключение.

Добавим в текст процедуры проверку

```
idw = (select W.id_wh   from warehouse W where W.name=:wh);
if (idw is null ) then exception;
```

Теперь при вызове с неправильным параметром в точку вызова (в том числе и в пользовательское приложение) будет возвращаться признак исключительной ситуации, который можно будет обработать. Обратите внимание, что запись информации об ошибке в LOG_FILE при этом мы исключили. При возникновении исключительной ситуации процедура завершается откатом (ROLLBACK). При этом все произведенные процедурой изменения в базе данных, в том числе и запись в таблицу LOG_FILE будут отменены. Как добиться того, чтобы все-таки делать запись об ошибке в LOG_FILE, мы рассмотрим в следующей лабораторной.

Задание 7. Дополните процедуру из заданий 5 и 6 проверкой входных параметров и вызовом исключения в случае ошибки.