Лабораторная № 10. КУРСОРЫ

Курсор – это основной механизм для перемещения по результирующему множеству исполняемого запроса.

В процедурах и триггерах курсор используется для организации циклов по таблице, получающейся в результате выполнения запроса.

Рассмотрим запрос, выводящий список агентов из города Азов.

```
select AG.name_ag
from agent AG
      where AG.town='Азов'
```

Опишем процедуру, которая выполняет аналогичную задачу для любого города, заданного в параметре.

До сих пор мы использовали так называемый курсорный цикл

```
create or alter procedure AG_LIST ( TOWN
     type of column AGENT.TOWN)
returns (
     NAME type of column AGENT.NAME_AG)
as
begin
for select    AG.name_ag
              from agent AG
              where AG.town=:town
     into :name
     do suspend;
end
```

Если процедурное расширение языка SQL не поддерживает курсорный циклы, то можно использовать явное определение курсора и итерационный цикл

```
create or alter procedure AG_LIST ( TOWN
     type of column AGENT.TOWN)
returns (
     NAME type of column AGENT.NAME_AG)
as
declare C_AG cursor for (
     select AG.NAME_AG from
     AGENT AG
     where AG.TOWN = :TOWN);
begin
open C_AG; -- курсор нужно явно открыть
```

```
 while (1=1) do
 begin
   fetch C_AG into :name;--извлечь строку по курсору
   if (row_count=0) then leave;--строк нет
   suspend;
 end
 close C_AG;--закрыть курсор
end
```

Такой курсор называется явным. Он может быть использован несколько раз для выполнения одного и того же запроса.

Обратите внимание, что курсор фактически описывает таблицу, которая далее будет использована с операции извлечения данных `fetch`. При этом, в соответствии с реляционной моделью, у таблицы должны быть определены имена всех столбцов. Поэтому для столбцов, получающихся в результате скалярных вычислений, необходимо явно использовать операцию переименования. Например, следующий курсор формирует таблицу из двух столбцов, имя первого совпадает с именем столбца в исходной таблице `TOWN`, имя второго столбца определено через операцию переименования `as cnt_ag`.

```
declare C_AG_COUNT cursor for
( select AG.TOWN,count(*) as cnt_ag
     from AGENT AG
    group by AG.TOWN );
```

Поскольку курсор служит для позиционирования в результирующей таблице допустимо использовать операции редактирования (update) и удаления (delete) по позиции курсора. Однако, это возможно только для курсоров, работающих с одной таблицей, без операции join и без подзапросов.

В следующем условном примере, к номеру телефона поставщика добавляется цифра 6, для поставщиков из Ростова, 1 – для Батайска, 4 – для Азова, 0 – для остальных. Поскольку номер телефона в таблице описан как char(10), при необходимости выполняется явное удаление символов справа.

```
create or alter procedure AG_PHONE_UPD
as
declare variable TW type of column AGENT.TOWN;
declare variable CIFR char(1);
declare C_AG cursor for (
    select AG.TOWN
    from AGENT AG);
begin
  open C_AG;
  while (1 = 1) do
```

```
  begin
    fetch C_AG
    into :TW;
    if (row_count = 0) then
      leave;
    CIFR = 0;
    if (TW = 'Ростов') then  CIFR = 6;
    if (TW = 'Батайск') then CIFR = 1;
    if (TW = 'Азов') then    CIFR = 4;
    update AGENT
    set PHONE=
          substring(:CIFR||coalesce(PHONE,'')from 1 for 10)
    where current of C_AG; --позиция курсора
  end
  close C_AG;
end
```

Поскольку курсор позволяет выполнять позиционное редактирование/удаление, в курсорном цикле for допускается определить курсор, связанный с оператором select данного цикла – неявный курсор. При этом можно не сохранять результаты запроса в локальных переменных, т.к. курсор в этом случае можно рассматривать как структуру, поля которой имеют имена, соответствующие именам столбцов в запросе.

Неявный курсор тоже можно использовать для редактирования/удаления по позиции курсора.

Например, в следующей процедуре выбираются все операции поставщика, и в каждой следующей операции увеличивается цена на 1% .

```
create or alter procedure BONUS (
    AG type of column AGENT.NAME_AG)
as
declare variable I integer = 0;
declare variable ID_A type of column AGENT.ID_AG;
begin
  /* Procedure Text */
  id_a=(select id_ag from agent where name_ag=:ag);
  if (id_a is null) then
  begin
  in autonomous transaction do
    insert into log_file(inform) values (' Error '||:ag);
  exception error;
  end
  for select * from operation
```

```
    where id_ag =:id_a
    order by op_date
  as cursor CRN do
  begin
  update operation
        set price = price *(1+0.01*:i)
        where current of CRN ;
  insert into log_file(inform,ddata)
          values ('price*(1+0.01*'||:i||')',CRN.op_date);
  i=i+1;
  end
end
```

Задания для самостоятельного решения

Реализовать с использованием курсоров следующие процедуры:

1. По названию поставщика выдать названия всех товаров, поставленные (R) данным поставщиком, с указанием для каждого товара даты последней поставки. Использовать явный курсор.

2. По паре дат выдать таблицу товаров, по которым были операции в этот период с указанием оборота товара в две колонки: приход, расход. Использовать явный курсор.

3. Увеличить стоимость каждой 3-й операции в заданном году на 10%. Использовать неявный курсор.

4. Для заданного периода времени подсчитать промежуточные итоги по каждому товару, с которым были операции в этот период. Итоги по товару должны иметь такой вид:

Название товара	Дата операции	тип операции	количество	промежуточный итог = итог за предыдущую дату +(-)количество
XXX	дд1	A	10	10
XXX	дд2	R	15	-5
XXX	дд3	A	20	15
YYY	дд4	A	5	5