

Тема 11. Программный SQL.

Язык SQL можно использовать как автономно в интерактивном режиме, так и в составе прикладных программ, написанных на стандартных языках программирования.

Имеются следующие способы применения SQL в программах:

- Модульный SQL. Операторы SQL собираются в отдельный модуль. Используется в хранимых процедурах.
- Встроенный SQL. Операторы SQL включаются непосредственно в текст программы.
- Интерфейс API. Операторы SQL передаются СУБД во время выполнения с помощью специальных функций.
- Компонентный SQL. SQL-операторы задаются как значения свойств компонентов приложения.

Встроенный SQL обычно отвечает следующим положениям:

- Компиляция операторов SQL производится препроцессором,
- Операторы SQL могут ссылаться на программные переменные и получать с их помощью результаты запросов.
- Для построчной обработки результатов запросов в SQL добавлены новые операторы.

Различают динамический и статический SQL: статический - операторы задаются до выполнения приложения, динамический - операторы формируются во время выполнения приложения.

Принято различать однострочные и многострочные запросы. Для однострочных запросов используется оператор SELECT вида:

```
SELECT [{ALL | DISTINCT}] список_столбцов  
INTO список_переменных  
FROM список_таблиц  
[WHERE условие].
```

Оператор получает результирующую таблицу и заносит ее первую строку в указанные переменные.

Пример.

Определить количество заказов.

```
SELECT COUNT (*)  
INTO V  
FROM Заказы
```

После выполнения оператора значение переменной V будет равно числу заказов.

Многострочные запросы

Используются курсоры – указатели результирующего набора записей, с помощью которых производится навигация и построчная обработка набора.

Операторы:

- DECLARE CURSOR – описание курсора,
- OPEN – открытие курсора,
- FETCH – навигация в наборе данных,
- CLOSE – закрытие курсора.

Простое определение курсора:

```
DECLARE имя_курсора CURSOR FOR запрос-SELECT
```

Расширенное определение:

```
DECLARE имя_курсора [INSENSITIVE] [SCROLL] CURSOR  
FOR оператор-SELECT  
[FOR {READ ONLY | UPDATE [OF {имя_столбца},...]}]
```

Параметры:

INSENSITIVE (нечувствительный) – нечувствительность к изменениям в исходных таблицах после открытия курсора.

SCROLL – можно использовать все операции навигации. Если SCROLL не указано, то доступна только NEXT.

READ ONLY – исходные таблицы не будут обновляться.

UPDATE – исходные таблицы будут обновляться.

Открытие курсора

```
OPEN имя_курсора [USING список_переменных]
```

Иницируется выполнение запроса. Код завершения операции возвращается в системной переменной SQLCODE. Сформированный набор данных доступен вплоть до закрытия курсора.

Чтение строки:

FETCH имя_курсора INTO список_переменных – простой оператор чтения.

Сложный оператор:

FETCH [NEXT | PRIOR | FIRST | LAST |

ABSOLUTE {n | имя_переменной} | RELATIVE {n | имя_переменной}]

FROM имя_курсора INTO список_переменных.

Здесь NEXT – чтение следующей строки, PRIOR – предыдущей, FIRST - первой, LAST – последней, ABSOLUTE – строки с указанным номером, RELATIVE строки с указанным смещением относительно текущей.

Закрытие курсора

CLOSE имя_курсора