Кафедра КИБЭВС

ТУСУР
Томский государственный университет систем управления и радиоэлектроники

Лекция №8
Встроенный SQL

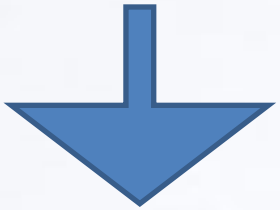
Ст. преподаватель
каф. КИБЭВС
М.А. Сопов

Существуют два способа применения SQL в прикладных программах:

1. Встроенный SQL. При таком подходе операторы SQL встраиваются непосредственно в исходный текст программы на базовом языке.

2. Интерфейс программирования приложений (API application program interface). При использовании данного метода прикладная программа взаимодействует с СУБД путем применения специальных функций.

Процесс выполнения операторов SQL может быть условно разделен на 5 этапов:



Синтаксический анализ оператора SQL	**Этап 1**
Проверка параметров оператора SQL	**Этап 2**
Оптимизация оператора SQL	**Этап 3**
Генерация плана выполнения запроса SQL	**План**
Двоичная форма выполнения запроса SQL	**Этап 4**
Исполнение плана запроса	**Этап 5**

Синтаксический анализ оператора SQL:

На этом этапе проверяется корректность записи SQL - оператора в соответствии с правилами синтаксиса.

Проверка параметров оператора SQL:

На этом этапе проверяется корректность параметров оператора SQL: имен отношений, имен полей данных, привилегий пользователя по работе с указанными объектами. Здесь обнаруживаются семантические ошибки.

Оптимизация оператора SQL:

СУБД проводит разделение целостного запроса на ряд минимальных операций и оптимизирует последовательность их выполнения с точки зрения стоимости выполнения запроса. На этом этапе строится несколько планов выполнения запроса и выбирается из них один — оптимальный для данного состояния БД.

Генерация плана выполнения запроса SQL:

СУБД генерирует двоичную версию оптимального плана запроса, подготовленного на этапе 3. Двоичный план выполнения запроса в СУБД фактически является эквивалентом объектного кода программы.

Двоичная форма выполнения запроса SQL:

СУБД реализует (выполняет) разработанный план, тем самым выполняя оператор SQL.

При объединении операторов SQL с базовым языком программирования должны соблюдаться следующие принципы:

- Операторы SQL включаются непосредственно в текст программы на исходном языке программирования. Исходная программа поступает на вход препроцессора SQL, который компилирует операторы SQL.

- Встроенные операторы SQL могут ссылаться на переменные базового языка программирования.

- Встроенные операторы SQL получают результаты SQL-запросов с помощью переменных базового языка программирования.

При объединении операторов SQL с базовым языком программирования должны соблюдаться следующие принципы:

- Для присвоения неопределенных значений (NULL) атрибутам отношений БД используются специальные функции.

- Для обеспечения построчной обработки результатов запросов во встроенный SQL добавляются несколько новых операторов, которые отсутствуют в интерактивном SQL.

Обычно *курсоры* используются для выбора из *базы данных* некоторого подмножества хранимой в ней информации. В каждый момент времени прикладной программой может быть проверена одна строка *курсора*. *Курсоры* часто применяются в операторах *SQL*, встроенных в написанные на языках процедурного типа прикладные программы.

В соответствии со стандартом *SQL* при работе с курсорами можно выделить следующие основные *действия*:

- создание или *объявление курсора* ;
- ***открытие курсора,*** т.е. наполнение его данными, которые сохраняются в многоуровневой памяти ;
- *выборка из курсора* и *изменение* с его помощью строк данных;
- *закрытие курсора*, после чего он становится недоступным для пользовательских программ;
- *освобождение курсора*, т.е. удаление *курсора* как объекта, поскольку его *закрытие* необязательно освобождает ассоциированную с ним память.

Для работы с курсором добавляется несколько новых операторов SQL

Встроенный оператор SELECT должен создавать структуры данных, которые согласуются с базовыми языками программирования.

Во встроенном SQL запросы делятся на 2 типа:

- Однострочные запросы, где ожидаемые результаты соответствуют одной строке данных. Эта строка может содержать значения нескольких столбцов.

- Многострочные запросы, результатом которых является получение целого набора строк. При этом приложение должно иметь возможность проработать все полученные строки. Значит, должен существовать механизм, который поддерживает просмотр и обработку полученного набора строк.

Первый тип запроса — однострочный запрос во встроенном SQL вызвал модификацию оператора SQL, которая выглядит следующим образом:

SELECT [{ALL | DISTINCT}] <список возвращаемых столбцов>

INTO <список переменных базового языка>

FROM <список исходных таблиц>

[WHERE <условия соединения и поиска>]

Для реализации многострочных запросов вводится новое понятие — понятие курсора или указателя набора записей.

Курсор в SQL – это область в памяти базы данных, которая предназначена для хранения последнего оператора SQL. Если текущий оператор – запрос к базе данных, в памяти сохраняется и строка данных запроса, называемая текущим значением, или текущей строкой *курсора*. Указанная область в памяти поименована и доступна для прикладных программ.

:

Оператор DECLARE CURSOR — определяет выполняемый запрос, задает имя курсора и связывает результаты запроса с заданным курсором. Этот оператор не является исполняемым для запроса, он только определяет структуру будущего множества записей и связывает ее с уникальным именем курсора. Этот оператор подобен операторам описания данных в языках программирования.

Оператор OPEN дает команду СУБД выполнить описанный запрос, создать виртуальный набор строк, который соответствует заданному запросу. Оператор OPEN устанавливает указатель записей (курсор) перед первой строкой виртуального набора строк результата.

Оператор FETCH продвигает указатель записей на следующую позицию в виртуальном наборе записей. В большинстве коммерческих СУБД оператор перемещения FETCH реализует более широкие функции перемещения, он позволяет перемещать указатель на произвольную запись, вперед и назад, допускает как абсолютную адресацию, так и относительную адресацию, позволяет установить курсор на первую или последнюю запись виртуального набора.

Оператор CLOSE закрывает курсор и прекращает доступ к виртуальному набору записей. Он фактически ликвидирует связь между курсором и результатом выполнения базового запроса. Однако в коммерческих СУБД оператор CLOSE не всегда означает уничтожение виртуального набора записей.

Курсоры — удобное средство для формирования бизнес-логики приложений, но следует помнить, что если вы открываете курсор с возможностью модификации, то СУБД блокирует все строки базовой таблицы, вошедшие в ваш курсор, и тем самым блокируется работа других пользователей с данной таблицей.

Чтобы свести к минимуму количество требуемых блокировок, при работе интерактивных программ следует придерживаться следующих правил:

- Необходимо делать транзакции как можно короче.
- Необходимо выполнять оператор завершения COMMIT после каждого запроса и как можно скорее после изменений, сделанных программой.

- Необходимо избегать программ, в которых осуществляется интенсивное взаимодействие с пользователем или осуществляется просмотр очень большого количества строк данных.

- Если возможно, то лучше не применять прокручиваемые курсоры (SCROLL), потому что они требуют блокирования всех строк выборки, связанных с открытым курсором.

- Использование простого последовательного курсора позволит системе разблокировать текущую строку, как только будет выполнена операция FETCH, что минимизирует блокировки других пользователей, работающих параллельно с вами и использующих те же таблицы.

- Если возможно, определяйте курсор как READ ONLY.

Спасибо за внимание!!!