УДК 004.6

ИСПОЛЬЗОВАНИЕ SQL-ЗАПРОСОВ

Кайсанова Ж.Ж., старший преподаватель, Жумабаева А.А., старший преподаватель
Казахский гуманитарно-юридический инновационный университет, г.Семей

Базы данных позволяют не только удобно размещать большие объёмы данных, но и достаточно быстро получать желаемую информацию. Для этого используется специальная команда, называемая запросом.

Это специальное строковое обращение к базе, в котором отражаются поля (частицы данных) и условия, по которым эту информацию необходимо предоставить.

Логика составления запроса данных из базы на большинстве языков программирования максимальна проста. Для получения информации необходимо указать системе управления, настроенной для конкретного хранилища, основные шесть параметров:

1. Названия таблиц, из которых необходимо забрать данные;
2. Поля, которые требуется вернуть; связи между таблицами;
3. Условия выборки (при необходимости);
4. Вспомогательные (сортировка, способы представления, ограничения и другие).

Благодаря такой структуре, пользователям достаточно понять только структуру желаемого запроса, после чего реализовать его "на бумаге" будет очень легко. Работает и обратная схема - для понимания того, какая информация получается той или иной командой, необходимо знать основы, чтобы разобраться с запросом. Это сделало язык SQL-запросов очень популярным как среди ИТ-специалистов, так и среди желающих освоить непростую науку программирования.

Язык SQL-программирования гибкий, благодаря чему для различных целей можно модифицировать запрос. Это может быть связано с особенностью архитектуры базы, которая будет сказываться на времени выполнения запроса, предотвращением возможных проблем на определённом этапе работы, а также удобочитаемостью полученного результата. [1-4]

Язык программирования SQL предназначен для конкретных, ограниченных целей - запросов данных, содержащихся в реляционной базе данных. Как таковой, он представляет собой набор инструкций языка программирования для создания выборок данных, а не процедурный язык, такой как C или BASIC , которые предназначены для решения гораздо более широкого круга проблем. Расширения языка, таких как «PL/SQL» предназначены для решения этого ограничения, добавив процедурные

элементы для SQL при сохранении преимуществ SQL. Другой подход заключается в том, что позволяется в запросы SQL встраивать команды процедурного языка программирования и взаимодействовать с базой данных.

В сравнении с большинством других языков программирования, SQL-запросы всегда возвращают чётко структурированный результат в виде таблицы. Поэтому при разработке инструментов, требующих активную работу с большими массивами данных, в код программы встраивают специализированный модуль, который обеспечивает быстрый и чёткий обмен информацией с базой данных, что может увеличить скорость работы до нескольких раз, особенно при работе пользователей. Помимо плюсов, есть одна существенная отрицательная черта у SQL-запроса. Это работа с полями, имеющими одинаковые имена. В этом случае необходимо максимально чётко контролировать выстраиваемую связь, указывать самостоятельно из какой таблицы должны забираться данные.

Прежде чем приступать к созданию таблицы SQL, необходимо определить модель базы данных. Спроектировать ER-диаграмму, в которой определить сущности, атрибуты и связи.

Сущности – предметы или факты, информацию о которых необходимо хранить. Например, сотрудник фирмы или проекты, реализуемые предприятием. Атрибуты – составляющая, которая описывает или квалифицирует сущность. Например, атрибут сущности «работник» - заработная плата, а атрибут сущности «проект» - сметная стоимость. Связи – ассоциации между двумя элементами. Она может быть двунаправленная. Также существует рекурсивная связь, то есть связь сущности с самой собой.

Также необходимо определить ключи и условия, при которых сохранится целостность базы данных.

Переход от ER-диаграммы к табличной модели

Правила перехода к табличной модели:

1. Преобразовать все сущности в таблицы.
2. Преобразовать все атрибуты в столбцы, то есть каждый атрибут сущности должен быть отображен в имени столбца таблицы.
3. Уникальные идентификаторы преобразовать в первичные ключи.
4. Все связи преобразовать во внешние ключи.
5. Осуществить создание таблицы SQL.

SQL слова делятся на ряд групп.

Первая – это **Data Manipulation Language или DML** (язык управления данными). DML является подмножеством языка, используемого для запроса к базам данных, добавления, обновления и удаления данных.

- SELECT является одной из наиболее часто используемых команд DML и позволяет пользователю задать запрос как описание желаемого результата в виде множества. В апросе не указано, каким образом результаты должны быть расположены - перевод запроса в форму, которая может быть выполнена в базе данных, является работой системы баз данных, более конкретно оптимизатора запросво.

- INSERT используется для добавления строк (формального набора) для существующей таблицы.
- UPDATE используется для изменения значений данных в существующей строке таблицы.
- DELETE определение существующих строк, которые будут удалены из таблицы.

Три другие ключевых слова, можно сказать, что попадают в группу DML:

- BEGIN WORK (или START TRANSACTION, в зависимости от диалекта SQL) могут быть использованы, чтобы отметить начало транзакции базы данных, которые либо выполнятся все полностью или вообще не выполнятся.
- COMMIT устанавливает, что все изменения данных в после совершения операций сохраняются.
- ROLLBACK определяет, что все изменения данных после последней фиксации или отката должны быть уничтожены, до того момента, который был зафиксирован в БД как «откат».

COMMIT и ROLLBACK применяются в таких областях, как контроль транзакций и блокировки. Обе инструкции завершают все текущие транзакции (наборы операций над БД) и снимают все блокировки на изменение данных в таблицах. Присутствие или отсутствие BEGIN WORK или аналогичного заявления зависит от конкретной реализации SQL.

Вторая группа ключевых слов относится к группе **Data Definition Language или DDL (язык определения данных)**. **DDL** позволяет пользователю определять новые таблицы и связанные с ними элементы. Большинство коммерческих баз данных SQL имеют собственные расширения в DDL, которые позволяют осуществлять контроль над нестандартныыми, но обычно жизненно важными элементами конкретной системы. Основные пункты DDL являются команды создавать и удалять.

- CREATE определяет объекты (например, таблицы), которые будут созданы в базе данных.
- DROP определяет, какие существующие объекты в базе данных будут удалены, как правило, безвозвратно.
- Некоторые системы баз данных также поддерживают команду ALTER, которая позволяет пользователю изменять существующий объект по-разному - например, так можно произвести добавление столбцов в существующую таблицу.

Третьей группой ключевых слов SQL является **Data Control Language или DCL (язык контроля данных)**. **DCL** отвечает за права доступа к данным и позволяет пользователю контролировать, кто имеет доступ, чтобы просматривать или манипулировать данными в базе данных. Здесь два основных ключевых слова:

- GRANT - разрешает пользователю выполнять операции

- REVOKE - удаляет или ограничивает возможность пользователю выполнять операции.

Системы баз данных с использованием SQL:

- DB2
- InterBase
- MySQL
- Oracle
- PostgreSQL
- SQL Server

Список литературы:

1. Браст, Э.Дж. Разработка приложений на основе Microsoft SQL Server 2008/Э.Дж. Браст. - М.: Русская Редакция, 2010. – 751 с.
2. Вишневский, Алексей Microsoft SQL Server. Эффективная работа/Алексей Вишневский. - М.: Питер, 2009. – 143 с.
3. Долгих, А. Microsoft SQL Server 2005. Практические методы работы (+ CD-ROM) / А. Долгих. - М.: Эком, 2007. – 356 с.
4. Дэвидсон, Луис Проектирование баз данных на SQL Server 2000/Луис Дэвидсон. - М.: Бином. Лаборатория знаний, 2003. - 662 с.