

История развития SQL.

SQL (Structured Query Language) — Структурированный Язык Запросов — стандартный язык запросов по работе с реляционными БД. Язык *SQL* появился после *реляционной алгебры*, и его прототип был разработан в конце 70-х годов в компании *IBM Research*. Он был реализован в первом прототипе реляционной СУБД фирмы *IBM System R*. В дальнейшем этот язык применялся во многих коммерческих СУБД и в силу своего широкого распространения постепенно стал стандартом "де-факто" для языков манипулирования данными в реляционных СУБД.

Первый *международный стандарт* языка *SQL* был принят в 1989 г. (далее мы будем называть его *SQL/89* или *SQL1*). Иногда стандарт *SQL1* также называют стандартом *ANSI/ISO*, и подавляющее большинство доступных на рынке СУБД поддерживают этот стандарт полностью. Однако развитие информационных технологий, связанных с базами данных, и необходимость реализации переносимых приложений потребовали в скором времени доработки и расширения первого стандарта *SQL*.

В конце 1992 г. был принят новый *международный стандарт* языка *SQL*, который в дальнейшем будем называть *SQL/92* или *SQL2*. И он не лишен недостатков, но в то же время является существенно более точным и полным, чем *SQL/89*. В настоящий момент большинство производителей СУБД внесли изменения в свои продукты так, чтобы они в большей степени удовлетворяли стандарту *SQL2*.

В 1999 году появился новый стандарт, названный *SQL3*. Если отличия между стандартами *SQL1* и *SQL2* во многом были количественными, то стандарт *SQL3* соответствует качественным серьезным преобразованиям. В *SQL3* введены новые типы данных, при этом предполагается возможность задания сложных *структурированных типов данных*, которые в большей степени соответствуют объектной ориентации. Наконец, добавлен раздел, который вводит стандарты на события и триггеры, которые ранее не затрагивались в стандартах, хотя давно уже широко использовались в коммерческих СУБД. В стандарте определены возможности четкой спецификации триггеров как совокупности события и действия. В качестве действия могут выступать не только *последовательность операторов SQL*, но и *операторы управления ходом выполнения программы*. В рамках управления транзакциями произошел возврат к старой модели транзакций, допускающей *точки сохранения (savepoints)*, и возможность указания в операторе отката **ROLLBACK** точек возврата позволит откатывать транзакцию не в начало, а в промежуточную ранее сохраненную точку. Такое решение

повышает гибкость реализации сложных алгоритмов обработки информации.

Разработка любой информационной системы, ориентированной на технологию баз данных (а других информационных систем на настоящий момент и не бывает), является трудоемким процессом, занимающим несколько десятков и даже сотен человеко-месяцев. Следует отдавать себе отчет, что нельзя разработать сколько-нибудь серьезную систему за несколько дней. Кроме того, развитие вычислительной техники, систем телекоммуникаций и программного обеспечения столь стремительно, что проект может устареть еще до момента внедрения. Но развивается не только вычислительная техника, изменяются и реальные объекты, поведение которых моделируется использованием как самой *БД*, так и процедур обработки информации в ней, то есть конкретных приложений, которые составляют реальное наполнение разрабатываемой информационной системы. Именно поэтому проект информационной системы должен быть рассчитан на *расширяемость* и *переносимость* на другие платформы. Большинство поставщиков аппаратуры и программного обеспечения следуют стратегии поддержки стандартов, в противном случае пользователи просто не будут их покупать. Однако каждый поставщик стремится улучшить свой продукт введением дополнительных возможностей, не входящих в стандарт. Выбор разработчиков, следовательно, таков: ориентироваться только на экзотические особенности данного продукта либо стараться в основном придерживаться стандарта. Во втором случае весь интеллектуальный труд, вкладываемый в разработку, становится более защищенным, так как система приобретает свойства переносимости. И в случае появления более перспективной платформы проект, ориентированный в большей степени на стандарты, может быть легче перенесен на нее, чем тот, который в основном ориентировался на особенности конкретной платформы. Кроме того, стандарты — это верный ориентир для разработчиков, так как все поставщики *СУБД* в своих перспективных разработках обязательно следуют стандарту, и можно быть уверенным, что в конце концов стандарт будет реализован практически во всех перспективных *СУБД*. Так произошло со стандартом *SQL1*, так происходит со стандартом *SQL2* и так будет происходить со стандартом *SQL3*.

Для поставщиков *СУБД* стандарт — это путеводная звезда, которая гарантирует правильное направление работ. А вот *эффективность реализации* стандарта — это гарантия успеха.

SQL нельзя в полной мере отнести к традиционным языкам программирования, он не содержит традиционные операторы, управляющие ходом выполнения

программы, операторы описания типов и многое другое, он содержит только набор стандартных операторов доступа к данным, хранящимся в базе данных. Операторы SQL встраиваются в базовый язык программирования, которым может быть любой стандартный язык типа C++, PL, COBOL и т. д. Кроме того, операторы SQL могут выполняться непосредственно в интерактивном режиме.

Структура SQL

В отличие от реляционной алгебры, где были представлены только операции запросов к БД, SQL является полным языком, в нем присутствуют не только операции запросов, но и операторы, соответствующие Data Definition Language (DDL) — языку описания данных. Кроме того, язык содержит операторы, предназначенные для управления (администрирования) БД.

SQL содержит разделы, представленные в табл. 5.1:

Таблица 5.1. Операторы определения данных DDL

Оператор	Смысл	Действие
CREATETABLE	Создать таблицу	Создает новую таблицу в БД
DROP TABLE	Удалить таблицу	Удаляет таблицу из БД
ALTER TABLE	Изменить таблицу	Изменяет структуру существующей таблицы или ограничения целостности, задаваемые для данной таблицы
CREATE VIEW	Создать представление	Создает виртуальную таблицу, соответствующую некоторому SQL-запросу
ALTER VIEW	Изменить представление	Изменяет ранее созданное представление
DROP VIEW	Удалить представление	Удаляет ранее созданное представление
CREATEINDEX	Создать индекс	Создает индекс для некоторой таблицы для обеспечения быстрого доступа по атрибутам, входящим в индекс
DROP INDEX	Удалить индекс	Удаляет ранее созданный индекс

Таблица 5.2. Операторы манипулирования данными Data Manipulation Language (DML)

Оператор	Смысл	Действие
DELETE	Удалить строки	Удаляет одну или несколько строк, соответствующих условиям фильтрации, из базовой таблицы. Применение оператора согласуется с принципами

		поддержки целостности, поэтому этот оператор не всегда может быть выполнен корректно, даже если синтаксически он записан правильно
INSERT	Вставить строку	Вставляет одну строку в базовую таблицу. Допустимы модификации оператора, при которых сразу несколько строк могут быть перенесены из одной таблицы или запроса в базовую таблицу
UPDATE	Обновить строку	Обновляет значения одного или нескольких столбцов в одной или нескольких строках, соответствующих условиям фильтрации

Таблица 5.3. Язык запросов Data Query Language (DQL)

Оператор	Смысл	Действие
SELECT	Выбрать строки	Оператор, заменяющий все операторы <i>реляционной алгебры</i> и позволяющий сформировать результирующее отношение, соответствующее запросу

Таблица 5.4. Средства управления транзакциями

Оператор	Смысл	Действие
COMMIT	Завершить транзакцию	Завершить комплексную взаимосвязанную обработку информации, объединенную в транзакцию
ROLLBACK	Откатить транзакцию	Отменить изменения, проведенные в ходе выполнения транзакции
SAVEPOINT	Сохранить промежуточную точку выполнения транзакции	Сохранить промежуточное состояние БД, пометить его для того, чтобы можно было в дальнейшем к нему вернуться

Таблица 5.5. Средства администрирования данных

Оператор	Смысл	Действие
ALTERDATABASE	Изменить БД	Изменить набор основных объектов в базе данных, ограничений, касающихся всей базы данных

ALTER DBAREA	Изменить область хранения БД	Изменить ранее созданную область хранения
ALTERPASSWORD	Изменить пароль	Изменить пароль для всей базы данных
CREATEDATABASE	Создать БД	Создать новую базу данных, определив основные параметры для нее
CREATEDBAREA	Создать область хранения	Создать новую область хранения и сделать ее доступной для размещения данных
DROPDATABASE	Удалить БД	Удалить существующую базу данных (только в том случае, когда вы имеете право выполнить это действие)
DROP DBAREA	Удалить область хранения БД	Удалить существующую область хранения (если в ней на настоящий момент не располагаются активные данные)
GRANT	Предоставить права	Предоставить права доступа на ряд действий над некоторым объектом БД
REVOKE	Лишить прав	Лишить прав доступа к некоторому объекту или некоторым действиям над объектом

Таблица 5.6. Программный SQL

Оператор	Смысл	Действие
DECLARE	Определяет курсор для запроса	Задаёт некоторое имя и определяет связанный с ним запрос к БД, который соответствует виртуальному набору данных
OPEN	Открыть курсор	Формирует виртуальный набор данных, соответствующий описанию указанного курсора и текущему состоянию БД
FETCH	Считать строку из множества строк, определенных курсором	Считывает очередную строку, заданную параметром команды из виртуального набора данных, соответствующего <i>открытому</i>

		<i>курсор</i>
CLOSE	Закрывает курсор	Прекращает доступ к виртуальному набору данных, соответствующему указанному курсору
PREPARE	Подготовить оператор SQL к динамическому выполнению	Сгенерировать план выполнения запроса, соответствующего заданному оператору SQL
EXECUTE	Выполнить оператор SQL, ранее подготовленный к динамическому выполнению	Выполняет ранее подготовленный план запроса