

Министерство образования и науки Российской Федерации
Федеральное государственное бюджетное образовательное
учреждение высшего образования "Московский
авиационный институт (национальный исследовательский университет)"
Институт №8

**Реферат по теме:
«Язык запросов SQL»**

Выполнил:
Студент 2 курса, группы М8О-204Б-22
Филиппов Фёдор Иванович
Форма обучения: Очная

Проверил:
Максим Алексеевич Потенко

Москва 2023

Введение.

Structured Query Language (SQL) позволяет пользователям создавать таблицы, добавлять, обновлять и удалять данные, а также выполнять запросы для извлечения информации из базы данных. Язык SQL имеет широкий набор команд и операторов, которые обеспечивают возможность эффективного и гибкого управления данными. SQL также поддерживает другие операции, такие как сортировка,

Таблица, созданная в БД SQL, включает в себя следующий ряд компонентов:

Ключи – то, что идентифицирует сущность в таблице.

Существуют три категории ключей: первичные, внешние и потенциальные. «Уникальность» сведений в БД проверяется первичными ключами (например, ни у одного физического лица не может быть одного и того же ИНН, потому что у каждого гражданина он уникальный). Сами таблицы, а также различные объекты внутри них (например, таблица с фильмами и таблица с создателями фильмов, снимающими эти фильмы) связаны внешним ключом. Потенциальный ключ - это значение, отличное от первичного ключа, которое может применяться для установления объекта в БД. К примеру, полное имя или дата рождения человека могут применяться для идентификации в таблице в дополнение к его ИНН. Однако, очевидно, что такой способ поиска сущности дает гораздо менее точные результаты, нежели первичный ключ, выдающий одно единственное значение сущности).

Сущность – это фрагмент информации в базе данных, а ключевым элементом, например, может быть порядковый номер строки.

Столбец «атрибут» определяет характеристики объектов, найденных в ячейках таблицы этого столбца. «Имя», «Дата» или «Стоимость» в качестве примеров.

Категория значения атрибута - домен. Например, стоимость должна быть представлена десятичным значением в столбце «Цена».

Строка таблицы с номером называется кортежем.

Значение – это информация, найденная в одной ячейке.

Элемент БД, упрощающий поиск по ней, называется индексом. Из-за большого количества записей в БД поиск требуемой информации может занять много времени. Индекс ускоряет процедуру.

Поскольку SQL является непроцедурным языком, в нем отсутствуют операторы управления, рутинная структура, ввод-вывод и другие подобные элементы. При этом SQL фокусируется на описании и манипуляции данными в базе данных, а не на управлении процессом выполнения программы. Однако, SQL обычно используется в сочетании с встроенными языками программирования СУБД, чтобы дополнить его функциональностью. Например, в случае использования SQL вместе с Visual Basic DBMS Access или другим языком программирования, можно использовать возможности этих языков для управления выполнением SQL-запросов, обработкой результатов запросов, взаимодействием с пользователями и другими операциями. Система управления базами данных (СУБД) представляет собой совокупность программ, которые позволяют управлять организацией, хранением и доступом к данным в базе данных. СУБД обеспечивает механизмы для создания, изменения и управления структурой базы данных, а также контроля доступа к данным и обеспечения их целостности и безопасности.

Принято выделять 2 ключевых способа использования встроенного языка SQL - динамический и статический.

Динамическое использование SQL подразумевает формирование запросов в виде текстовых строк на основе условий, заданных во время выполнения программы. Такой подход обычно используется, когда точный SQL-запрос, который будет выполняться, неизвестен заранее или должен быть динамически сгенерирован. Динамический подход позволяет гибко создавать и изменять запросы на основе различных условий и требований.

Статическое использование SQL, с другой стороны, предполагает написание структурно фиксированного текста программы, который содержит вызовы функций или операторов SQL для выполнения запросов. Статический

подход более предсказуем, поскольку SQL-запросы определяются заранее и не изменяются в процессе выполнения программы. Однако для обеспечения гибкости и возможности использования различных параметров запроса, также могут быть использованы переменные компьютерного языка.

Выбор между динамическим и статическим подходом зависит от конкретных требований и условий программы. Динамический подход предоставляет большую гибкость, но требует дополнительной логики для формирования и выполнения запросов во время выполнения программы. Статический подход предоставляет более предсказуемую работу, но менее гибок при изменении запросов.

Описание языка SQL.

В 1970-х годах в исследовательском центре IBM была создана экспериментальная система управления базами данных под названием IBM System R. Данная система стала одной из первых реляционных БД и сыграла значительную роль в развитии реляционной модели данных. Для удобного доступа и манипуляции данными в IBM System R был разработан специальный язык запросов, который назывался SEQUEL (Structured English Query Language). SEQUEL позволял использовать структурированный английский язык для формулирования запросов к базе данных. С течением времени, из-за конфликта с названием другой компании, SEQUEL был переименован в SQL (Structured Query Language).

Хотя SQL был задуман и создан как полноценный язык баз данных, способный использовать каждый желающий, включая людей без знания программирования, SEQUEL не был первым языком, специально разработанным для работы с базами данных. В Калифорнийском университете в Беркли была разработана некоммерческая СУБД под названием Ingres. Ingres использовал свой собственный язык запросов, называемый QUEL (Query Language) для выполнения операций с базами данных. Несмотря на то, что QUEL имел свои достоинства и использовался в некоторых системах, он не

смог конкурировать с SQL по количеству поддерживаемых СУБД. SQL стал преобладающим языком запросов благодаря его простоте, понятности и широкой поддержке от многих коммерческих СУБД.

Стандартизация.

В 1986 году было создано первое официальное языковое стандартное описание языка SQL под названием SQL-86. Стандарт SQL-86 определял основные элементы языка и функции, которые должны были поддерживаться СУБД для совместимости.

На тот момент существовало множество различных СУБД от разных производителей, и каждая из них имела свои собственные особенности и способы реализации языка запросов. С разработкой стандарта SQL, целью было сделать язык более переносимым, чтобы программное обеспечение, написанное для одной СУБД, могло быть перенесено и использовано с минимальными изменениями в другой СУБД, которая полностью соответствовала стандарту.

С течением времени были разработаны более новые и обновленные версии стандарта SQL, такие как SQL-89, SQL-92, SQL:1999, SQL:2003 и т.д., каждая из которых добавляла новые возможности и улучшения в язык SQL. Эти стандарты помогли обеспечить совместимость и переносимость программного обеспечения, написанного на SQL, между разными СУБД.

Создание единого стандарта языка SQL было совместным усилием Американского национального института стандартов (ANSI) и Международной организации по стандартизации (ISO). Процесс стандартизации SQL начался в 1983 году с обсуждений и консультаций между ANSI и ISO. Начальная версия стандарта, известная как "Язык баз данных SQL", была представлена ANSI и после нескольких раундов обсуждений и исправлений была принята и опубликована как стандарт ANSI SQL-86. Также этот стандарт получил технический номер ISO 9075-1987.

Неофициальное название SQL-86 было SQL1, и стандарт включал два уровня. Первый уровень был более подробным описанием подмножества

второго уровня, что обеспечивало возможность выборочной реализации элементов стандарта языка SQL. Поставщики СУБД, заявляющие о поддержке этого стандарта, должны были указать реализацию, соответствующую уровню поддержки SQL. Это позволяло производителям постепенно внедрять поддержку стандарта и способствовало принятию и поддержке стандарта.

После изначального стандарта SQL-86, стандарт SQL-89 был обновлен и улучшен в 1989 году. В этом обновлении введены и использованы понятия основных и внешних ключей, которые служат для обеспечения целостности данных. В связи с этим, документ был переименован в "Язык баз данных SQL с улучшениями целостности" (ISO 9075-1989) для издания через ICE (International Electrotechnical Commission).

Параллельно с разработкой SQL-89, также был разрабатывался стандарт SQL-92 (SQL-2), который впоследствии стал заменой для SQL-89. SQL-92 представлял собой расширение предыдущего стандарта SQL-89, представляя его в окончательной форме. В SQL-92 использовалась та же система уровней, что и в SQL-89, но с некоторыми изменениями в отношении названий. Вместо использования порядковых номеров, уровни стали именами: начальный, средний и полный.

Следовательно, SQL является информационно-логическим языком, который используется для определения, изменения и извлечения данных из баз данных. Изначально SQL задумывался как основной способ взаимодействия пользователя с базой данных и предоставлял набор возможностей для работы с данными.

Основные задачи, которые можно выполнять с помощью SQL, включают:

1. Создание новых таблиц в базе данных.
2. Добавление, обновление и удаление записей в таблице.
3. Выборка записей из одной или нескольких таблиц на основе заданного условия.
4. Манипулирование структурой таблицы, такой как добавление или удаление столбцов.

Со временем SQL существенно расширил свои возможности. Он стал поддерживать различные конструкции и предоставлять возможность определять и управлять хранимыми объектами, такими как индексы, представления, триггеры и хранимые процедуры. SQL стал все больше напоминать язык программирования для обычных пользователей.

Несмотря на эти изменения, прямые соединения между прикладным программным обеспечением и базами данных по-прежнему осуществляются с использованием SQL. SQL остается основным языком для общения с базами данных.

Каждое предложение SQL является вызовом базы данных или запросом, который изменяет базу данных.

Следующие запросы классифицируются в зависимости от того, какие изменения вносятся в базу данных:

- запросы на добавление новых объектов в базу данных или изменение существующих (в этом случае в запросе указывается тип и структура добавляемого или изменяемого объекта);
- запросы на получение данных; запросы на добавление новых данных (записей);
- запросы на удаление данных; и запросы на получение доступа к СУБД.

Таблица является основным объектом хранения реляционной базы данных, следовательно, все SQL-запросы являются действиями над таблицами. В результате запросы разделены на три категории: те, которые воздействуют непосредственно на таблицы (построение и изменение таблиц); те, которые работают с конкретными записями; и те, которые работают с коллекциями записей. Каждая таблица представлена перечислением ее полей (столбцов таблицы) с информацией о типах значений, хранящихся в каждом поле, связях между таблицами (путем указания первичного и вторичного ключей) и подробностях, необходимых для создания индексов.

Запросы первого типа далее подразделяются на те, которые предназначены для создания новых таблиц базы данных, и те, которые

предназначены для изменения существующих. Вторая категория запросов включает следующие три типа операций:

1. Добавление новой строки. Данный тип запроса позволяет добавить новую запись в таблицу БД.

2. Изменение значений строки или набора полей строк. Посредством данного типа запроса можно обновить значения в существующей строке или наборе полей строк.

3. Удаление строки или набора строк. Данный запрос позволяет удалить конкретную строку или набор строк из таблицы БД.

Наиболее важным и широко используемым видом запроса является запрос выборки (SELECT), который предоставляет пользователю определенный набор строк, соответствующих заданным условиям. Полученный набор строк можно использовать для изучения данных, изменения значений или удаления строк. Это действительно эффективно сводит использование SQL к созданию различных выборок строк и выполнению операций над каждой записью в наборе данных.

Преимущества и недостатки.

Преимущества.

Свобода от определенной СУБД.

Вообще говоря, тексты SQL-запросов, содержащих DDL и DML, могут быть просто перенесены из одной СУБД в другую, несмотря на существование диалектов и различий в синтаксисе. Существует несколько систем, создатели которых изначально придавали большое значение использованию по крайней мере нескольких СУБД (например, система электронного документооборота Documentum может использоваться с Oracle, Microsoft SQL Server и IBM DB2). Естественно, внедрение некоторых специфичных для реализации функций уже сейчас делает достижение такой переносимости очень сложной задачей.

Наличие стандартов.

"Стабилизация" языка становится возможной только благодаря существованию стандартов и набора тестов для определения совместимости и соответствия конкретной реализации SQL широко признанному стандарту. Важно отметить, что сам стандарт иногда может быть очень формальным и пространным (например, основной раздел стандарта SQL:2003 содержит более 1300 страниц).

Декларативность.

При использовании SQL программист просто указывает данные, которые необходимо извлечь или изменить. СУБД сама принимает это решение при обработке SQL-запроса. Однако не думайте, что это применимо ко всем ситуациям. Программист может указать набор данных для выборки или изменения, но ему будет полезно, если он сможет увидеть, как СУБД интерпретирует формулировку его запроса. Возможность вариаций написания они различаются по скорости выполнения, но схожи по конечному набору данных, увеличивается со сложностью запроса.

Недостатки.

Реляционная модель данных и ее несоответствие с языком SQL.

По мнению разработчиков реляционной модели данных Эдгара Кодда, Кристофера Дейта и их последователей, SQL не является настоящим реляционным языком. Они конкретно упоминают следующие проблемы SQL: повторяющиеся строки, неопределенные значения (null), явное указание порядка столбцов слева направо, столбцы без имени и повторяющиеся имена столбцов, отсутствие поддержки свойства "=", использование указателей и высокая избыточность (в контексте языка SQL означает, что в запросе или структуре базы данных есть лишние, неэффективные или избыточные элементы, которые не вносят дополнительной ценности, но могут затруднять понимание и управление базой данных). Хью Дарвен и Кристофер Дейт представляют язык под названием Tutorial D, который на самом деле является реляционным, и излагают осново-

полагающие идеи следующего поколения систем управления базами данных в своей книге "Третий манифест".

Сложность.

Хотя SQL задумывался как инструмент конечного пользователя, со временем он превратился в инструмент программирования из-за своей сложности.

Отклонение от нормы.

Несмотря на существование всемирного стандарта ANSI SQL-92, несколько фирм-разработчиков СУБД (таких как Oracle, Sybase, Microsoft и MySQL AB) изменяют язык SQL, используемый в создаваемой СУБД, отклоняясь от стандарта. Следовательно, существуют диалекты языка SQL, которые уникальны для каждой отдельной СУБД.

Использование иерархических фреймворков затруднительно.

Метод изменения древовидных структур ранее не был доступен в диалектах SQL большинства СУБД. Фраза CONNECT BY используется Oracle, среди других поставщиков СУБД, в качестве одного из своих решений. На данный момент стандарты ANSI включают рекурсивную конструкцию WITH из диалекта SQL DB2. Только версия MS SQL Server 2005 поддерживала рекурсивные запросы.

Типы данных.

Основные типы данных, используемые в SQL, формы которых могут существенно отличаться для различных СУБД, следующие: Целое число (часто до 10 значащих цифр и знака) называется целым числом. SMALLINT – "короткое целое число" (часто до пяти цифр с существенными различиями и знаком); Десятичное число, содержащее p цифр (0–p–16) плюс знак, называется DECIMAL (p,q). При использовании q (q ≤ p, q = 0 можно опустить) указывается количество

цифр после десятичной точки; Тип СУБД определяет целочисленный порядок действительного числа с плавающей точкой, которое содержит 15 значащих цифр; CHAR(n) – это символьная строка с заданной длиной n символов (0–n-256);

n > 0 и варьируется в зависимости от СУБД, но не может быть меньше 4096, это максимальное количество символов, которое может быть включено в строку VARCHAR(n); Поля даты могут содержать только истинные даты, начинающиеся за несколько тысячелетий до нашей эры, и разрешено содержать даты только с пятого по десятое тысячелетие нашей эры; DATA – это дата в формате, предоставленном определенной командой (по умолчанию мм/дд/гг); ВРЕМЯ - время в формате, выбранном определенной командой (по умолчанию: чч.мм.сс); DATA-TIME – это слияние даты и времени; MONEY - деньги в формате, который определяет символ денежной единицы (\$, руб,...) и его положение (суффикс или префикс), точность дробной составляющей и необходимые условия для отображения денежного значения.

SQL-функции.

SQL-функции - это класс специализированных стандартных функций. Каждая из этих функций работает с набором значений столбцов из таблицы и генерирует одно значение, описанное следующим образом, за исключением специального значения COUNT(*): Существует четыре столбца: COUNT, SUM, AVG, MAX и MIN. COUNT - это количество значений в столбце; SUM - общее количество значений в столбце; MAX - наибольшее значение в столбце; и AVG - среднее значение в столбце. Рассматриваемый столбец должен иметь числовые значения, чтобы использовать методы SUM и AVG.

Следует подчеркнуть, что столбец в данном контексте относится к столбцу виртуальной таблицы, который может содержать информацию, полученную не только из столбца базовой таблицы, но также из функциональных преобразова-

ний или сопряжения арифметических операций с символами, представляющими значения из одного или нескольких столбцов. Хотя оператор, описывающий столбец такой таблицы, может быть произвольно сложным, функции SQL не должны содержаться внутри (вложенность функций SQL не допускается). Но вы можете комбинировать любые функции SQL для создания любого выражения.

Термин DISTINCT (вариетум) может использоваться для указания на то, что избыточные повторяющиеся значения должны быть удалены перед применением любой функции, за исключением COUNT(*), в аргументе. Чтобы подсчитать каждую строку в таблице без исключения (включая дубликаты), используйте специальный метод COUNT(*).

Возможности SQL.

В начале 1970-х годов SQL был просто языком запросов (QL). На самом деле, все, что присутствовало, - это предложение SELECT, которое позволяло создавать запросы для получения данных из базы данных. Затем в язык были добавлены еще две части, необходимые для использования баз данных. Первый из них - это метод определения структуры базы данных, известный на жаргоне теории баз данных как язык описания данных (DL). Второй - это способ добавлять данные в базу данных, отредактировать их, а затем удалить. Язык манипулирования данными (NMD) - это название, данное этому элементу в теории баз данных. Кроме того, было решено, что для всего интерфейса базы данных следует использовать единый язык, предоставляющий SQL возможности, необходимые для управления базой данных.

Вот несколько задач управления базой данных: определение, переопределение и удаление таблиц в базе данных и других ее элементов (доменов, представлений, индексов, триггеров, хранимых процедур, функций и т.д.); описание физической структуры данных; поддержка целостности базы данных и ограничений соответствия; защита данных от незаконного использования путем идентифи-

кации пользователей (с указанием имен и паролей), прав доступа к данным и прав на изменение состояния базы данных; корректировки данных в таблицах базы данных, включая вставку данных. SQL выпускается в двух вариантах. В интерактивном SQL пользователь вводит инструкции и немедленно получает результаты. Тексты программ на других языках содержат встроенные инструкции SQL. В данном случае эти программы получают доступ к базе данных и обрабатывают результаты.

SQL-технологии.

Язык структурированных запросов (SQL) используется для взаимодействия с сервером СУБД. Используемый при таком взаимодействии язык не является процедурным. Он был создан специально для создания запросов и работы с данными и структурами данных. В нем отсутствуют все распространенные программные конструкции, такие как переменные, метки, циклы и другие подобные инструменты. Важно понимать, что, хотя SQL описывает метод доставки данных в клиентскую программу, он не содержит никаких положений о том, как эти данные должны обрабатываться или показываться пользователю в клиентской программе. Многие поставщики СУБД предоставляют свои собственные расширения, поскольку исходный стандарт, очевидно, не может удовлетворить всем требованиям. Например, Oracle и IBM предлагают свои собственные версии расширения инструкции SELECT, которые вы можете использовать для быстрого расширения иерархически отсортированных данных в горизонтальное дерево (в Oracle это называется START WITH / CONNECT BY). Для этих целей необходимо создать хранимые процедуры, поскольку в диалекте SQL Informix нет эквивалентного оператора. Для сервера СУБД от одного провайдера могут существовать десятки расширений. Языки SQL также могут быть расширены за счет уникальных процедурных возможностей. Синтаксис SQL довольно гибок, и сам язык идентичен обычным процедурным языкам в том смысле, что содержит все стандартные переменные, метки, циклы и другие функции.

Производители СУБД определяют синтаксис так, как они считают нужным, поскольку формального стандарта для процедурных расширений не существует. Опять же, существует масса проприетарных расширений, и Informix, в частности, предоставляет курсоры с произвольным расположением. До разработки реляционных баз данных все языки манипулирования данными были разработаны для различных систем управления базами данных (СУБД), используемых на персональных компьютерах. Эти языки манипулирования данными в основном использовались для взаимодействия с данными, которые были представлены в виде логических файловых записей. Для достижения этой цели пользователи должны были хорошо разбираться в организации хранения данных, а также приложить достаточно усилий, чтобы определить не только необходимые данные, но и их местоположение и пошаговую процедуру их получения.

Операции с данными, представленными в виде логически связанных групп таблиц, являются основной целью непроцессуального языка SQL. Уникальность слов этого языка заключается в том, что они делают больший акцент на результате обработки данных, чем на самом процессе. Вам не нужно описывать эту информацию в запросе к базе данных, поскольку SQL сам знает, где хранятся данные, какие индексы использовать для их поиска и даже наиболее эффективный порядок выполнения операций. В ответ на теорию реляционных баз данных и "альфа"-язык запросов Кодда, основанный на реляционном исчислении [2, 3], появился ряд языков запросов. Эти языки можно разделить на два класса: алгебраические языки и функциональные языки. Набор рекомендаций по созданию выражения, описывающего новое отношение из существующего набора отношений, известен как язык исчисления предикатов.

Другими словами, исчисление предикатов — это метод выбора желаемого соединения (в качестве ответа на запрос) из связей, которые уже есть в базе данных. Большая часть работы была выполнена в колледжах США (PIQUE, QUEL) и подразделениях IBM (языки ISBL, SQL и QBE) [3]. Последняя была разработана для СУБД INGRES (Интерактивная графическая и поисковая система), ко-

торая была создана в начале 1970-х годов и сегодня продолжает входить в пятерку лучших СУБД для профессионалов. В настоящее время для расширения внутренних языков СУБД используются только наиболее интригующие конструкции из других языков, оставляя только QBE (Query-By-Example – пример запроса) и SQL, которые были полностью сохранены и расширяются.

Выводы.

Определение данных, доступ к ним и их обработка осуществляются с использованием специального языка, известного как SQL (Structured Query Language). Язык SQL является примером непроцедурного языка, поскольку он просто определяет желаемые результаты и необходимые компоненты (такие как таблицы), но не то, как эти результаты должны быть получены.

Каждая реализация SQL является компонентом процессора базы данных (database engine), который понимает инструкции SQL и выбирает соответствующую последовательность доступа к структурам базы данных для правильного и эффективного получения необходимых результатов.

В настоящее время ISO - Международная организация по стандартизации (International Organization for Standardization) утвердила стандарт SQL, который был разработан ANSI - Американским национальным институтом стандартов (American National Standards Institute).

Сервер базы данных получает уведомление о том, что и как должно быть сделано, используя SQL, который не является процедурным языком. Сервер базы данных преобразует команды SQL во внутренние операции для выполнения запроса. SQL прост в использовании, поскольку он скрывает специфику обработки данных.

Использованные источники

1. <https://habr.com/ru/articles/564390/>
2. <https://ru.wikipedia.org/wiki/SQL>
3. <https://academy.yandex.ru/journal/how-sql>
4. <https://support.microsoft.com/ru-ru/office/access-sql-основные-понятия-лексика-и-синтаксис-444d0303-cde1-424e-9a74-e8dc3e460671>