SQL — [декларативный](https://ru.wikipedia.org/wiki/%D0%94%D0%B5%D0%BA%D0%BB%D0%B0%D1%80%D0%B0%D1%82%D0%B8%D0%B2%D0%BD%D0%BE%D0%B5_%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5) [язык программирования](https://ru.wikipedia.org/wiki/%D0%AF%D0%B7%D1%8B%D0%BA_%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D1%8F), применяемый для создания, модификации и управления данными в [реляционной базе данных](https://ru.wikipedia.org/wiki/%D0%A0%D0%B5%D0%BB%D1%8F%D1%86%D0%B8%D0%BE%D0%BD%D0%BD%D1%8B%D0%B5_%D0%B1%D0%B0%D0%B7%D1%8B_%D0%B4%D0%B0%D0%BD%D0%BD%D1%8B%D1%85), управляемой соответствующей [системой управления базами данных](https://ru.wikipedia.org/wiki/%D0%A1%D0%B8%D1%81%D1%82%D0%B5%D0%BC%D0%B0_%D1%83%D0%BF%D1%80%D0%B0%D0%B2%D0%BB%D0%B5%D0%BD%D0%B8%D1%8F_%D0%B1%D0%B0%D0%B7%D0%B0%D0%BC%D0%B8_%D0%B4%D0%B0%D0%BD%D0%BD%D1%8B%D1%85).

В начале [1970-х годов](https://ru.wikipedia.org/wiki/1970-%D0%B5_%D0%B3%D0%BE%D0%B4%D1%8B) в одной из исследовательских лабораторий компании [IBM](https://ru.wikipedia.org/wiki/IBM) была разработана экспериментальная реляционная [СУБД](https://ru.wikipedia.org/wiki/%D0%A1%D0%B8%D1%81%D1%82%D0%B5%D0%BC%D0%B0_%D1%83%D0%BF%D1%80%D0%B0%D0%B2%D0%BB%D0%B5%D0%BD%D0%B8%D1%8F_%D0%B1%D0%B0%D0%B7%D0%B0%D0%BC%D0%B8_%D0%B4%D0%B0%D0%BD%D0%BD%D1%8B%D1%85) [IBM System R](https://ru.wikipedia.org/wiki/IBM_System_R), для которой затем был создан специальный язык SEQUEL, позволявший относительно просто управлять данными в этой СУБД. Аббревиатура SEQUEL расшифровывалась как Structured English QUEry Language — «структурированный английский язык запросов».

Поскольку к началу 1980-х годов существовало несколько вариантов СУБД разных производителей, причём, каждый из них обладал собственной реализацией языка запросов, было принято решение разработать стандарт языка, который будет гарантировать переносимость ПО с одной СУБД на другую (при условии, что они будут поддерживать этот стандарт).

Чтобы разобраться, как именно работает магия запроса, давайте представим его путь от пользователя до нужных ему данных:

Клиент — способ введения запроса. В случае с Google, например, клиентом будет поисковая строка браузера, в которую пользователь вводит сформулированный запрос.

Система управления базами данных (СУБД) — комплекс программ, которые позволяют управлять данными. Эта система помогает таблицам понять, чего хочет пользователь, а пользователю — что ему отвечают таблицы.

База данных — система хранения таблиц, в которой они связаны между собой. База данных сама по себе не умеет манипулировать информацией — это просто хранилище, где у каждого объекта есть свое место.

SQL-операторы

Работать с данными помогают операторы — определенные слова или символы, которые используются для выполнения конкретной операции — например, для выбора из множества по конкретному параметру. Если нам нужно из всех видов пиццы отсортировать те, в которых есть пармезан, — нужно использовать оператор SELECT (выбор в соответствии с условием).

Операторы в SQL делятся на несколько групп в соответствии с задачами, которые они решают.

DDL (Data Definition Language) — операторы определения данных. Они работают с объектами, то есть с целыми таблицами. Если базу нужно дополнить таблицей с новыми данными или, наоборот, убрать одну из таблиц с ошибочными данными — используется этот набор операторов.

CREATE — создание объекта в базе данных

ALTER — изменение объекта

DROP — удаление объекта

DML (Data Manipulation Language) — операторы манипуляции данными. Эти операторы уже работают с содержимым таблиц — строками, атрибутами и значениями. С их помощью можно вносить изменения в конкретное значение. Например, заменить поле в колонке «Фамилия» в строке с данными сотрудницы компании посте того, как она вышла замуж. Или удалить строку с данными уволенного сотрудника.

SELECT — выбор данных в соответствии с условием

INSERT — добавление новых данных

UPDATE — изменение существующих данных

DELETE — удаление данных

DCL (Data Control Language) — оператор определения доступа к данным. Он определяет, кто из пользователей может отправлять запросы к базе, менять объекты и значения. Например, можно отозвать доступ у сотрудника, перешедшего в другой отдел, а также открыть доступ к базе новому маркетологу или разработчику.

GRANT — предоставление доступа к объекту

REVOKE — отзыв ранее выданного разрешения

DENY — запрет, который является приоритетным над разрешением

TCL (Transaction Control Language) — язык управления транзакциями. Транзакции — это набор команд, которые выполняются поочередно. Если все команды выполнены, транзакция считается успешной, а если где-то произошла ошибка — транзакция откатывается назад, отменяя все выполненные команды. Наглядный пример такой транзакции — оплата онлайн, когда банк просит сначала ввести сумму и получателя, затем проверить и подтвердить операцию, а после ввести одноразовый код. На каждом из этих этапов оплату можно отменить и транзакция откатится назад.

BEGIN TRANSACTION — обозначение начала транзакции

COMMIT TRANSACTION — изменение команд внутри транзакции

ROLLBACK TRANSACTION — откат транзакции

SAVE TRANSACTION — указание промежуточной точки сохранения внутри

Сами по себе таблицы или база данных не способны выполнять операции, а в СУБД можно создавать новые таблицы, удалять ненужные данные, настраивать ключи и обрабатывать запросы. Основные задачи СУБД:

поддержка языков баз данных;

непосредственное управление данными;

управление буферами оперативной памяти;

управление транзакциями;

резервное копирование и восстановление после сбоев.

Существуют разные виды таких систем, которые разрабатывает и техногиганты, вроде Google, Microsoft и Amazon, и более нишевые студии. Разработчики стремятся сделать свой продукт лучше, чтобы их система быстрее и качественнее других обрабатывала данные. Из-за этого появились разные виды языка SQL — так называемые SQL-диалекты. У каждой СУБД диалект имеет что-то общее со всеми, а также свои особенности, которые не будут работать в другой системе.

PostgreSQL — это объектно-ориентированная система, то есть она обрабатывает данные как абстрактные объекты. Каждый объект, в отличие от простых табличных значений, может иметь собственные характеристики и уникальные методы взаимодействия с другими объектами. Это позволяет PostgreSQL обрабатывать более сложные структуры данных и выполнять более сложные процедуры. Например, Яндекс.Почта в свое время [перешла](https://habr.com/ru/post/321756/) на эту систему, чтобы поддерживать стабильное соединение десятков тысяч пользователей к одной базе.

MySQL — простая в изучении и функциональная система, которая работает с сайтами и веб-приложениями. Чаще всего используется в системах управления контентом сайтов (CMS), на сайтах с возможностью регистрации пользователей, в корпоративных системах CRM, в планировщиках, чатах и форумах. MySQL считается одним из самых безопасных и высокоскоростных решений, которое существует на рынке.

SQLite — это облегченная встраиваемая версия СУБД. В ней нет возможности поделиться правами доступа, как во многих других системах, но благодаря своему устройству эта система быстрая и мощная. SQLite подходит для обработки запросов на сайтах с низким и средним трафиком, а также в однопользовательских мобильных приложениях и играх. Преимущество такой системы — файловая структура, то есть база в SQLite состоит из одного файла, поэтому ее очень легко переносить.

Oracle — одна из первых СУБД, которая появилась еще в 1977 году и развивается до сих пор. Это кроссплатформенная система, которая может работать на Windows, Linux, MacOS, мобильных и других ОС. Система используется в крупных коммерческих проектах. Например, в России с Oracle сотрудничают операторы МТС и Теле2, банк «Открытие» и ВТБ.

Google Cloud Spanner — это облачная система управления данными, которую Google разработал для управления собственными сервисами, например AdWords и Google Play. В 2017 году систему сделали общедоступной. Cloud Spanner относят к категории NewSQL — это системы, которые совмещают в себе преимущества реляционных и нереляционных СУБД.

Преимущества[[править](https://ru.wikipedia.org/w/index.php?title=SQL&stable=0&veaction=edit&section=8) | [править код](https://ru.wikipedia.org/w/index.php?title=SQL&action=edit&section=8)]

Независимость от конкретной СУБД

Несмотря на наличие диалектов и различий в синтаксисе, в большинстве своём тексты SQL-запросов, содержащие DDL и DML, могут быть достаточно легко перенесены из одной СУБД в другую. Существуют системы, разработчики которых изначально ориентировались на применение по меньшей мере нескольких СУБД (например: система электронного [документооборота](https://ru.wikipedia.org/wiki/%D0%94%D0%BE%D0%BA%D1%83%D0%BC%D0%B5%D0%BD%D1%82%D0%BE%D0%BE%D0%B1%D0%BE%D1%80%D0%BE%D1%82) [Documentum](https://ru.wikipedia.org/w/index.php?title=Documentum&action=edit&redlink=1) может работать как с [Oracle Database](https://ru.wikipedia.org/wiki/Oracle_Database), так и с [Microsoft SQL Server](https://ru.wikipedia.org/wiki/Microsoft_SQL_Server) и [DB2](https://ru.wikipedia.org/wiki/DB2)). Естественно, что при применении некоторых специфичных для реализации возможностей такой переносимости добиться уже очень трудно.

Наличие стандартов

Наличие стандартов и набора тестов для выявления совместимости и соответствия конкретной реализации SQL общепринятому стандарту только способствует «стабилизации» языка. Правда, стоит обратить внимание, что сам по себе стандарт местами чересчур формализован и раздут в размерах (например, базовая часть стандарта SQL:2003 состоит из более чем 1300 страниц текста).

Декларативность

С помощью SQL программист описывает только то, какие данные нужно извлечь или модифицировать. То, каким образом это сделать, решает СУБД непосредственно при обработке SQL-запроса. Однако не стоит думать, что это полностью универсальный принцип — программист описывает набор данных для выборки или модификации, однако ему при этом полезно представлять, как СУБД будет разбирать текст его запроса. Чем сложнее сконструирован запрос, тем больше он допускает вариантов написания, различных по скорости выполнения, но одинаковых по итоговому набору данных.

Недостатки[[править](https://ru.wikipedia.org/w/index.php?title=SQL&stable=0&veaction=edit&section=9) | [править код](https://ru.wikipedia.org/w/index.php?title=SQL&action=edit&section=9)]

Несоответствие реляционной модели данных

Создатели реляционной модели данных [Эдгар Кодд](https://ru.wikipedia.org/wiki/%D0%9A%D0%BE%D0%B4%D0%B4,_%D0%AD%D0%B4%D0%B3%D0%B0%D1%80), [Кристофер Дейт](https://ru.wikipedia.org/wiki/%D0%94%D0%B5%D0%B9%D1%82,_%D0%9A%D1%80%D0%B8%D1%81%D1%82%D0%BE%D1%84%D0%B5%D1%80) и их сторонники указывают на то, что SQL не является истинно реляционным языком. В частности, они указывают на следующие дефекты SQL с точки зрения реляционной теории[10]:

допущение строк-дубликатов в таблицах и результатах выборок, что в рамках реляционной модели данных невозможно и недопустимо;

поддержка неопределённых значений ([NULL](https://ru.wikipedia.org/wiki/NULL_(SQL))), создающая фактически многозначную логику;

значимость порядка столбцов, возможность ссылок на столбцы по номерам (в реляционной модели столбцы должны быть равноправны);

допущение столбцов без имени, дублирующихся имён столбцов.

В опубликованном Кристофером Дейтом и [Хью Дарвеном](https://ru.wikipedia.org/w/index.php?title=%D0%94%D0%B0%D1%80%D0%B2%D0%B5%D0%BD,_%D0%A5%D1%8C%D1%8E&action=edit&redlink=1) [Третьем манифесте](https://ru.wikipedia.org/w/index.php?title=%D0%A2%D1%80%D0%B5%D1%82%D0%B8%D0%B9_%D0%BC%D0%B0%D0%BD%D0%B8%D1%84%D0%B5%D1%81%D1%82&action=edit&redlink=1)[11] они излагают принципы СУБД следующего поколения и предлагают язык [Tutorial D](https://ru.wikipedia.org/wiki/Tutorial_D), который является подлинно реляционным.

Сложность

Хотя SQL и задумывался как средство работы конечного пользователя, позже он стал настолько сложным, что превратился в инструмент программиста.

Отступления от стандартов

Несмотря на наличие международного стандарта ANSI SQL-92, многие разработчики СУБД вносят изменения в язык SQL, применяемый в разрабатываемой СУБД, тем самым отступая от стандарта. Таким образом появляются специфичные для каждой конкретной СУБД диалекты языка SQL.

Сложность работы с иерархическими структурами

Ранее диалекты SQL большинства СУБД не предлагали способа манипуляции древовидными структурами. Некоторые поставщики СУБД предлагали свои решения (например, в [Oracle Database](https://ru.wikipedia.org/wiki/Oracle_Database) используется выражение CONNECT BY). В настоящее время в ANSI стандартизована рекурсивная конструкция WITH из диалекта SQL [DB2](https://ru.wikipedia.org/wiki/DB2). В [Microsoft SQL Server](https://ru.wikipedia.org/wiki/Microsoft_SQL_Server) рекурсивные запросы (Recursive Common Table Expressions) появились с версии 2005[12].