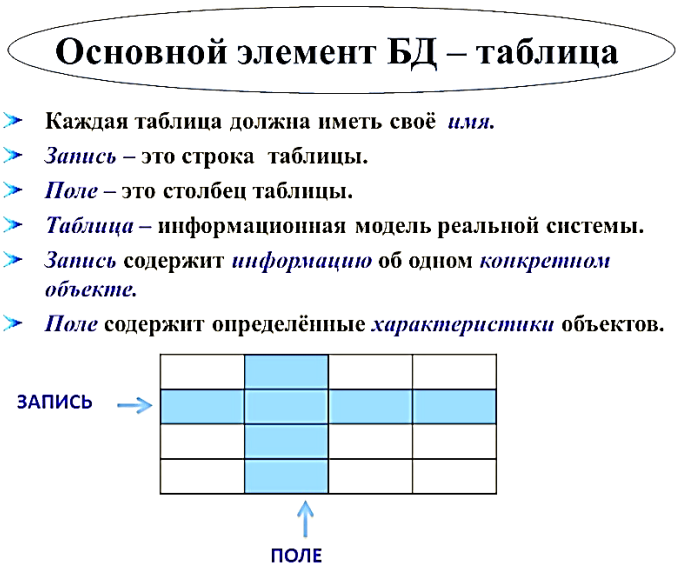
Определение и описание базы данных. Описание СУБД. Отличия БД и СУБД

База данных (БД) — это стандартный программный сервис для упорядоченного хранения данных. Повсеместно используется в том числе в клиент-серверной архитектуре. База данных – **совокупность данных, организованных по определённым правилам, предусматривающим общие принципы описания, хранения и манипулирования данными, независимая от прикладных программ**. Эти данные относятся к определённой предметной области и организованы таким образом, что могут быть использованы для решения многих задач многими пользователями.

Структура БД



Из определения базы данных следует, что в ней:

- всегда есть имя. Если имя не задано, то нет и базы данных;
- фиксируется состояние объектов и их отношений в заданный момент времени. Со временем оно меняется. Например, цена товара может характеризовать его состояние. Вслед за изменением цены меняется и состояние товара;
- фиксируется информация об объектах из определенной предметной области. Например, если рассматриваем предметную область «Библиотека», то в базе могут фиксироваться данные по книгам, их расположению в библиотеке, читателям и читательским билетам. Если наша предметная область — «Магазин», то в БД может находиться информация по товарам и их ценам, по торговым точкам и наличию товара в конкретной торговой точке.

Важной характерной чертой БД является ее постоянство. Оно проявляется в нескольких контекстах:

- данные постоянно накапливаются и используются;
- состав и структура данных обычно постоянны и стабильны во времени. Если они меняются, то скорее всего БД находится в процессе проектирования и разработки;

- элементы данных могут меняться (вслед за изменением состояний объектов и их отношений). Тем самым информация, которую содержит каждая база данных, постоянно актуализируется.

Свойства базы данных

1. Быстродействие. Современные БД проектируются по принципу «получить данные прямо сейчас», чтобы пользователь не ждал отклик на запрос.

2. Простота получения и обновления данных. Какой бы высокой ни была скорость, это бессмысленно, если нужно сделать много сложных операций, чтобы получить, обновить или добавить данные в базу.

3. Независимость структуры. Изменения в любом количестве и качестве информации не должны влиять на структуру базы данных. Также изменения не должны касаться программного обеспечения и средств хранения, например жёсткого диска.

4. Стандартизация. Аналогично свойству независимости структуры: при обновлении программного обеспечения или СУБД (сокр. от «системы управления базами данных») база данных не должна менять свою структуру или свойства.

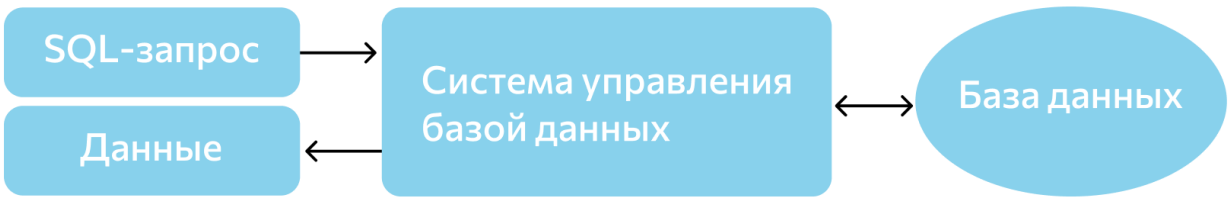
5. Безопасность данных. Для каждой категории пользователей делают список ограничений и доступов, согласно которым можно взаимодействовать с информацией.

6. Интегрированность. Данные должны быть логически связаны. И эти связи должны прослеживаться по структуре таблиц.

7. Многопользовательский доступ. Удалённо вносить изменения и получать информацию могут сразу несколько человек с разных устройств.

Чтобы управлять данными в базе, используют СУБД (систему управления базами данных). Каждому типу баз подходят свои СУБД.

СУБД — это интерфейс или система *управления базами данных* — совокупность языковых и программных средств, предназначенных для создания, наполнения, обновления и удаления баз данных. Программное обеспечение, которое необходимо для создания, редактирования и обслуживания файлов БД. С его помощью можно упростить процесс работы — от ввода данных до отчетности. Кроме того, система управления базами данных помогает выполнять резервное копирование, поддерживать безопасность, предоставлять общий доступ к БД. СУБД позволяет работать с базами данных одновременно нескольким пользователям.



Чем отличается СУБД от БД:

Система управления базами данных (СУБД) является субъектом управления и программой, а база данных объектом управления и собственно данными, которыми управляет СУБД.

- ✓ БД - это данные хранимые в структурированном виде. Обычно в виде набора файлов.
- ✓ СУБД — это программа или библиотека, которая умеет с этими файлами работать.

Виды баз данных и их описание

На сегодняшний день насчитывается более 50 разных видов баз данных. Однако, несмотря на разнообразие, существует несколько категорий, на которые их можно разделить:

- **Иерархические базы данных.** Такие БД имеют древовидную структуру. Объекты такой модели хранения представлены на разных уровнях. Есть корневой каталог, в нем — иерархия подкаталогов и файлов. Простейшая структура, где записи, как ветви, отходят от «родителя». Информация связана по аналогии с папками на рабочем столе. У каждой записи есть физическая связь только с одной предыдущей, а отношение многих ко многим невозможно.
- **Сетевые базы данных** или СБД — это теоретико-графовые модели БЗ. Они находятся в одной группе с иерархическими моделями. На основе математики именно сетевых баз созданы разные системы управления баз данных или СУБД, как правило, коммерческого характера. В отличие от иерархической структуры, у каждой записи может быть более одного родителя. Сетевые БД представляют собой не древовидный, а общий граф. Для СБД есть ряд характерных операций:

    - ✓ навигация;
    - ✓ манипуляция;
    - ✓ управление.

- **Объектные или объектно-ориентированные.** Объектные БД предназначены для высокопроизводительной работы со сложными структурами данных. Элементы такой базы имеют свои атрибуты, методы и классы. Базы данных, где информация о реальных вещах представлена в виде объектов под уникальным идентификатором. К ООБД обращаются на языке объектно-ориентированного программирования (ООП). Состояние объекта описывается атрибутами, а его поведение — набором методов. Объекты с одинаковыми атрибутами и методами образуют классы. Объект в ООП создаётся как отдельная сущность со своими свойствами и методами работы. И как только объект создан, его можно вызвать по «имени», или коду, а не разрабатывать заново.
- **Реляционные**. Реляционные базы данных — БД, которые используются для хранения и предоставления доступа к взаимосвязанным элементам информации. Их также называют SQL — как язык программирования, с помощью которого создают, преобразовывают и управляют данными в реляционных БД. Записи и связи между ними организованы при помощи таблиц. В таблицах есть поле для внешнего ключа со ссылками на другие таблицы. Благодаря высокой организации и гибкости структуры реляционные БД применяются для многих типов данных.

- **Объектно-реляционные** — это базы данных, совмещающие в себе свойства объектных и реляционных БД.

Сетевые базы данных. Они являются более продвинутым вариантом иерархических БД. Отличие состоит только в структуре — в иерархических БД элемент-потомок может иметь только одного предка, а в сетевой структуре данных у потомка может быть любое число предков.

Типы баз данных, их особенности и отличия

Типы баз данных

Существует множество критериев определения видов баз данных, в т.ч. следующие:

Форма представления информации:

Фактографические. Данные представлены в виде фактов об объектах предметной области в формате пар «параметр — значение». Пример: БД сайта www.ozon.ru.
Документальные. Данные представлены в виде полнотекстовых документов. Пример: БД сайта www.vedomosti.ru.
Мультимедийные. Данные представлены в виде графического, аудио- или видеоконтента. Пример: БД сайта www.youtube.com.

Тип используемой модели данных:

Реляционные. Данные представлены в виде таблиц и связей между ними. Пример: БД Microsoft SQL Server, MySQL, PostgreSQL.
Нереляционные. Данные представлены в виде структур, отличных от таблиц. Например, JSON-подобных объектов, иерархических или сетевых структур. Пример: БД ElasticSearch, MongoDB.

Топология хранения:

Большинство современных БД может быть размещено как на одной, так и на нескольких машинах.
Локальные. Размещены на одной машине.
Распределенные. Размещены на нескольких машинах.

Функциональное назначение:

Операционные. Большую часть времени используются для операций записи (добавление, изменение, удаление данных). Пример: БД 1С.
Справочно-информационные. Большую часть времени используются для операций чтения. Пример: БД сайта www.consultant.ru.

Степень доступности:

Общедоступные. Открыты широкому кругу пользователей. Обычно доступ к базам данных бесплатный. Пример: БД энциклопедии Wikipedia.
С ограниченным доступом. Доступ к базам данных ограничен и обычно платный. Пример: БД энциклопедии Encarta.

Наиболее применяемые на сегодня СУБД, использующие свои стандарты (расширения) SQL:

- **MySQL** – СУБД, принадлежащая компании Oracle. Реляционная СУБД с открытым исходным кодом, то есть доступна для просмотра, исправления ошибок и создания новых версий программ. MySQL — бесплатная, быстрая и гибкая система, подходящая для таблиц разных типов.
- **PostgreSQL** – свободная СУБД, поддерживаемая и развиваемая сообществом. Объектно-реляционная СУБД, которую используют для сайтов, сервисов и платформ. Бесплатный доступ и поддержка многих языков программирования делают эту СУБД одной из самых популярных. По её лицензии создано немало расширенных версий, в том числе для коммерческого использования.

- **Microsoft SQL Server** – СУБД, принадлежащая компании Microsoft. Применяет диалект Transact-SQL (T-SQL). Оптимальная СУБД для операционных систем Windows, но совместима и с Linux. Легко интегрируется с другими продуктами Microsoft, удобна в использовании, но потребляет много ресурсов, а лицензия стоит дорого.

Благодаря тому, что диалекты SQL что создаются, специфицируются и используются разными организациями, имеют как общие черты, так и ряд отличий в возможностях расширений.
Общими чертами диалектов являются основные конструкции, применимые практически без отличий во многих реляционных БД.

Основные отличия диалектов состоят в различиях использованных типов данных, количеством, реализацией и детальными возможностями команд. Разные диалекты применяют как разные наборы зарезервированных слов, так и разные наборы команд.

«SQL»: описание, использование, команды.

SQL расшифровывается как **язык структурированных запросов** (Structured Query Language) и является языком, который вы используете для управления данными в базах данных. SQL состоит из команд и декларативных операторов, которые действуют как инструкции для базы данных, чтобы она могла выполнять задачи.
Сегодня этот **язык программирования** является стандартом для оптимиза**ции** и обслуживания реляционных баз данных. Язык SQL оказался очень живучим, он актуален и по сей день. Фактически, **SQL** является набором стандартов, для написания запросов к БД. Последняя действующая редакция стандартов языка SQL **– *ISO/IEC 9075:2016.*** Основываясь на указанных стандартах языка SQL, ряд организаций выпустили свои, расширенные версии стандартов указанного языка. Подобные версии иногда называют диалектами SQL.

Варианты спецификаций SQL разрабатываются компаниями и сообществами и служат, соответственно, для работы с разными СУБД (Системами Управления Базами Данных) – системами программ, заточенных под работу с продуктами из своей инфраструктуры.

Вы можете использовать команды SQL для создания таблицы в базе данных, для добавления и изменения больших объемов данных, для поиска в ней, чтобы быстро найти что-то конкретное, или для удаления таблицы целиком.

В некоторых системах баз данных **требуется указывать точку с запятой в конце каждого оператора**. Точка с запятой является стандартным указателем на конец каждого оператора в SQL. В **MySQL** точка с запятой требуется.

База данных SQL - это набор связанной информации, хранящейся в таблицах. В каждой таблице есть столбцы, описывающие данные в них, и строки, содержащие фактические данные. Поле - это отдельный фрагмент данных в строке.

SQL команды – база, которую необходимо знать при работе с языком SQL. Язык SQL или Structured Query Language (язык структурированных запросов) предназначен для управления данными в системе реляционных баз данных (RDBMS).
- SHOW DATABASES - SQL-команда, которая отвечает за просмотр доступных баз данных.
- CREATE DATABASE - Команда для создания новой базы данных.
- USE - С помощью этой SQL-команды USE <database_name> выбирается база данных, необходимая для дальнейшей работы с ней.

- SOURCE - A SOURCE <file.sql> позволит выполнить сразу несколько SQL-команд, содержащихся в файле с расширением .sql.
- DROP DATABASE - Стандартная SQL-команда для удаления целой базы данных.
- SHOW TABLES - С помощью этой несложной команды можно увидеть все таблицы, которые доступны в базе данных.
- CREATE TABLE - SQL-команда для создания новой таблицы
- DESCRIBE - С помощью DESCRIBE <table_name> можно просмотреть различные сведения (тип значений, является ключом или нет) о столбцах таблицы.
- INSERT - Команда INSERT INTO <table_name> в SQL отвечает за добавление данных в таблицу
- UPDATE - SQL-команда для обновления данных таблицы:
- DELETE - SQL-команда DELETE FROM <table_name> используется для удаления данных из таблицы.
- DROP TABLE - можно удалить всю таблицу целиком.
- Основные команды SQL, которые позволяют работать непосредственно с данными:
- SELECT - Далее мы рассмотрим. К одной из таких SQL-команд относится SELECT для получения данных из выбранной таблицы
- SELECT DISTINCT - В столбцах таблицы могут содержаться повторяющиеся данные. Используйте SELECT DISTINCT для получения только неповторяющихся данных.
- WHERE - Можно использовать ключевое слово WHERE в SELECT для указания условий в запросе
- GROUP BY - Оператор GROUP BY часто используется с агрегатными функциями, такими как COUNT, MAX, MIN, SUM и AVG, для группировки выходных значений.
- HAVING - Ключевое слово HAVING было добавлено в SQL по той причине, что WHERE не может использоваться для работы с агрегатными функциями.
- ORDER BY - используется для сортировки результатов запроса по убыванию или возрастанию. ORDER BY отсортирует по возрастанию, если не будет указан способ сортировки ASC или DESC.
- BETWEEN - используется для выбора значений данных из определённого промежутка. Могут быть использованы числовые и текстовые значения, а также даты.
- LIKE - используется в WHERE, чтобы задать шаблон поиска похожего значения.
- IN - можно указать несколько значений для оператора WHERE:
- JOIN - используется для связи двух или более таблиц с помощью общих атрибутов внутри них. На изображении ниже показаны различные способы объединения в SQL. Обратите внимание на разницу между левым внешним объединением и правым внешним объединением:
- VIEW - виртуальная таблица SQL, созданная в результате выполнения выражения. Она содержит строки и столбцы и очень похожа на обычную SQL-таблицу. VIEW всегда показывает самую свежую информацию из базы данных.

Агрегатные функции

Это не совсем основные команды SQL. Агрегатные функции используются для получения совокупного результата, относящегося к рассматриваемым данным:

- COUNT(col_name) — возвращает количество строк;
- SUM(col_name) — возвращает сумму значений в данном столбце;
- AVG(col_name) — возвращает среднее значение данного столбца;
- MIN(col_name) — возвращает наименьшее значение данного столбца;
- MAX(col_name) — возвращает наибольшее значение данного столбца.

Описание базовых запросов SQL: CRUD (Create, Read, Update, Delete) и Select

Базовые запросы SQL:

- **SQL оператор READ (ядро СУБД)**

Метод Read считывает двоичное представление значения **SqlHierarchyId** из переданного объекта **BinaryReader** и присваивает это значение объекту **SqlHierarchyId**. Метод Read невозможно вызвать с помощью Transact-SQL. Пользуйтесь вместо этого инструкцией CAST или CONVERT.

- **SQL оператор CREATE DATABASE**

Чтобы создать базу данных с именем **engineering**, мы можем использовать следующий код:

```
CREATE DATABASE engineering;
```

SQL оператор CREATE TABLE

```
CREATE TABLE table_name (
   column1 datatype,
   column2 datatype,
   column3 datatype
);
```

Этот запрос создает новую таблицу внутри базы данных.

Он указываем имятаблица, и различные столбцы, которые мы хотим, чтобы наша имела таблица.

Все операции, которые вы можете выполнять с данными, следуют аббревиатуре CRUD.

CRUD обозначает 4 основные операции, которые мы выполняем при запросе базы данных: **C**reate (создание), **R**ead (чтение), **U**pdate (обновление) и **D**elete (удаление).

Ниже мы рассмотрим некоторые базовые запросы SQL вместе с их синтаксисом, необходимые чтобы начать работу с базами данных.

- **SQL оператор UPDATE**

Вот как мы обновляем строку в таблице. Это U(update) часть CRUD.

```
UPDATE table_name
SET column1 = value1,
   column2 = value2
WHERE condition;
```

Условие WHERE указывает запись, которую вы хотите отредактировать.

```
UPDATE engineering
SET    country = 'Spain'
WHERE   employee_id = 1
```

- **SQL оператор DELETE**

DELETE - это D-часть CRUD. С помощью оператора DELETE мы удаляем запись из таблицы.

```
DELETE FROM table_name
WHERE condition;
```

Например, удадение из нашей таблицы engineering удаление записи может выглядеть так:

```
DELETE FROM engineering
WHERE employee_id = 2;
```

Этот запрос удаляет запись сотрудника с идентификатором 2 из таблицы engineering.

Запрос SQL зачастую более производителен чем написание кода. В разных базах данных синтаксис SQL почти не отличается. Практически в каждом запросе присутствуют ключевые слова

SELECT, FROM и WHERE — фундаментальные аспекты построения запросов к базе. Более сложные запросы являются надстройками над ними.

Вложенные подзапросы
Вложенные подзапросы — это SQL-запросы, которые включают выражения SELECT, FROM и WHERE, вложенные в другой запрос.

DB Testing: для чего тестировщик обращается к базе данных

Основные причины для тестирования базы данных:

- Необходимость проверки целостности данных
- Проверка достоверности и целостности данных в самой базе данных, поскольку бэк-энд система отвечает за хранение данных, а доступ к ней имеет многоцелевое назначение.

Тестировщик может использовать базы данных для:

- ✓ создания среды тестирования,
- ✓ создания тестовых данных,
- ✓ проверки предполагаемого дефекта
- ✓ в целом для тестирования

Общие причины тестирования базы данных:

1. Облегчение сложности обращения к бэк-энд базы данных.
2. Хранимые процедуры и представления содержат важные задачи, такие как вставка сведений о клиенте (имя, контактная информация и т. д.) и данных о продажах. Эти задачи необходимо протестировать на нескольких уровнях.
3. Тестирование черного ящика на внешнем интерфейсе важно, но затрудняет выявление проблемы. Тестирование в бэк-энд системе повышает надежность данных. По этой причине тестирование базы данных выполняется на внутренней системе.
4. В базу данных данные поступают из нескольких приложений, вследствие этого, есть вероятность того, что в ней могут попасть вредоносные или неправильные данные.

Следовательно, необходимо регулярно проверять компоненты базы данных, а также, их целостность и согласованность.