

Database

База данных - это упорядоченный набор структурированной информации или данных, которые обычно хранятся в электронном виде в компьютерной системе. Данные вместе с СУБД, а также приложения, которые с ними связаны, называются системой баз данных, или, для краткости, просто базой данных.

Данные в наиболее распространенных типах современных баз данных обычно хранятся в виде строк и столбцов формирующих таблицу. В большинстве баз данных для записи и запросов данных используется язык структурированных запросов (SQL).

Система управления базами данных (СУБД) - это комплекс программно-языковых средств, позволяющих создать базы данных и управлять данными. Набор программ, позволяющий организовывать, контролировать и администрировать базы данных.

Базы данных - это специально разработанное хранилище для различных типов данных. Каждая база данных, имеет определённую модель (реляционная, документно-ориентированная), которая обеспечивает удобный доступ к данным. Системы управления базами данных (СУБД) - специальные приложения (или библиотеки) для управления базами данных различных размеров и форм.

Виды баз данных:

Текстовые:

Информация об объектах собирается в простых по структуре файлах различных форматов - txt, csv и др. Для разделения полей применяются пробелы, табуляция, запятые, точка с запятой и двоеточие.

```
root:x:0:0:root:/root:/bin/bash
daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin
bin:x:2:2:bin:/bin:/usr/sbin/nologin
sys:x:3:3:sys:/dev:/usr/sbin/nologin
sync:x:4:65534:sync:/bin:/bin/sync
games:x:5:60:games:/usr/games:/usr/sbin/nologin
man:x:6:12:man:/var/cache/man:/usr/sbin/nologin
lp:x:7:7:lp:/var/spool/lpd:/usr/sbin/nologin
mail:x:8:8:mail:/var/mail:/usr/sbin/nologin
news:x:9:9:news:/var/spool/news:/usr/sbin/nologin
backup:x:34:34:backup:/var/backups:/usr/sbin/nologin
list:x:38:38:Mailing List Manager:/var/list:/usr/sbin/nologin
nobody:x:65534:65534:nobody:/nonexistent:/usr/sbin/nologin
syslog:x:102:106::/home/syslog:/usr/sbin/nologin
```

ПЛЮСЫ:

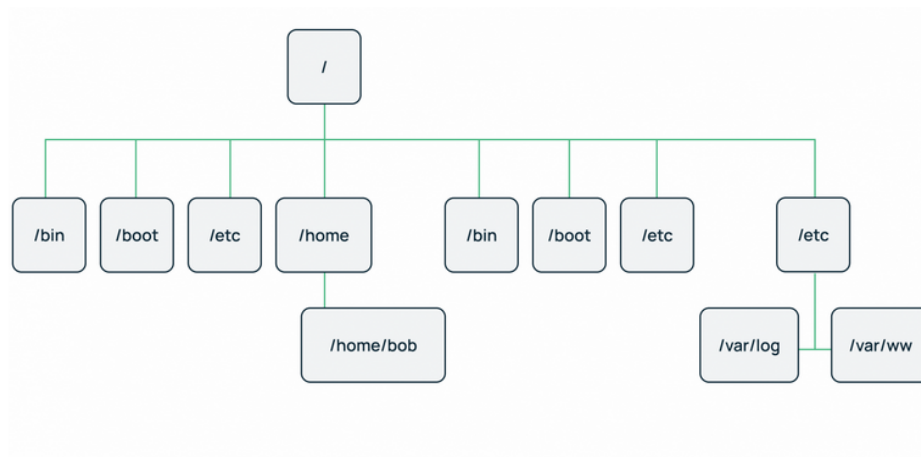
- Просто использовать. Для работы с файлами достаточно примитивного текстового редактора;
- Удобно работать с конфигурационными данными приложений (учетные данные, настройки подключения к удаленным серверам и устройствам, порты и пр.).

МИНУСЫ:

- Сложно установить связи между компонентами данных;
- Не для всех типов информации.

Иерархические:

В отличие от текстовых файлов здесь между хранимыми объектами устанавливаются связи. Объекты делятся на родителей (основные классы или категории объектов) и потомков (экземпляры этих классов или категорий). При этом у каждого потомка может быть не более одного родителя.



ПЛЮСЫ:

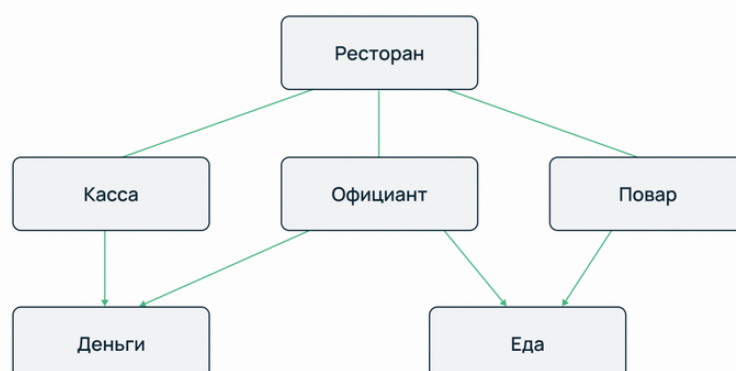
- Отношения между объектами реализованы в виде физических; указателей. Например, в файловой системе путь к папке или файлу строится из имен корневых и вложенных каталогов;
- Моделирование отношений вложенности и подчиненности.

МИНУСЫ:

Технология иерархической организации не предполагает связи «многие-ко-многим», а значит, система хранения данных довольно ограничена.

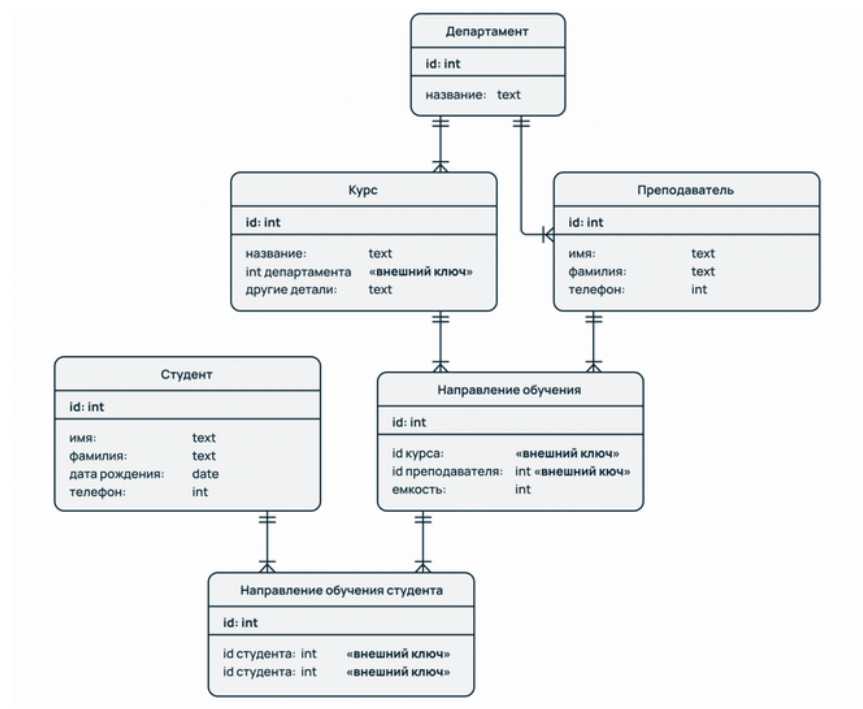
Сетевые:

Эта технология развивает иерархический подход за счет моделирования сложных отношений между объектами. Здесь потомки могут иметь более одного родителя, однако ограничения иерархического подхода сохраняются.



Реляционные:

Данные формируются в таблицы из строк и столбцов. В строках приводятся сведения об объектах (значения свойств), а в столбцах — сами свойства объектов (поля). Сложные взаимоотношения объектов в реляционных БД моделируются с помощью внешних ключей - ссылок на другие таблицы. Это позволяет подходить к вопросу проектирования базы данных с позиций нормализации - минимизации избыточности при описании свойств объектов.



ПЛЮСЫ:

- Минимизируется объем базы данных;
- Повышается целостность системы;
- Упрощается масштабирование;
- Повышается отказоустойчивость.

Коротко о других видах баз данных:

- **Объектно-ориентированные базы данных** - информация в объектно-ориентированной базе данных представлена в форме объекта, как в объектно-ориентированном программировании;
- **Распределенные базы данных** - распределенная база данных состоит из двух или более частей, расположенных на разных серверах. Такая база данных может храниться на нескольких компьютерах;

- **Хранилища данных** - будучи централизованным репозиторием для данных, хранилище данных представляет собой тип базы данных, специально предназначенной для быстрого выполнения запросов и анализа;
- **Oracle NoSQL Database** - или нереляционная база данных, дает возможность хранить и обрабатывать неструктурированные или слабоструктурированные данные;
- **Графовые базы данных** - хранит данные в контексте сущностей и связей между сущностями;
- **OLTP** - это база данных предназначенная для выполнения бизнес-транзакций, выполняемых множеством пользователей;
- **Базы данных с открытым исходным кодом** - Такие базы данных имеют открытый исходный код и могут управляться средствами как SQL, так и NoSQL;
- **Облачные базы данных** - представляет собой набор структурированных или неструктурированных данных, размещенный на частной, публичной или гибридной платформе облачных вычислений. Существует два типа моделей облачных баз данных: традиционная база данных и база данных как услуга (DBaaS). В модели DBaaS административные задачи и обслуживание выполняются поставщиком облачных услуг;
- **Многомодельные базы данных** - многомодельная база данных объединяет разные типы моделей баз данных в единую интегрированную серверную СУБД. Это означает, что она может содержать различные типы данных;
- **Документные базы данных/JSON** - предназначены для хранения, извлечения и обработки документоориентированной информации и предоставляют современный способ хранения данных в формате JSON, а не в виде строк и столбцов;
- **Автономные базы данных** - это новейшие и самые революционные облачные базы данных, которые используют машинное обучение для автоматизации настройки, защиты, резервного копирования, обновления и других стандартных задач обслуживания, обычно выполняемых администраторами баз данных.

Типы баз данных:

Существуют 3 наиболее распространённые СУБД:

- MySQL;
- PostgreSQL;
- Microsoft SQL.

Далее приведу таблицу различий и их особенностей между ними:

Название	Microsoft SQL	MySQL	PostgreSQL
Модель базы данных	Реляционная	Реляционная	Реляционная
Вторичная модель	Документная Графовая Пространственная	Документная Пространственная	Документная Пространственная
Сайт	www.microsoft.com/en-us/sql-server	www.mysql.com	www.postgresql.org
Техническая документация	docs.microsoft.com/en-US/sql/sql-server	dev.mysql.com/doc	www.postgresql.org/docs
Изготовитель	Microsoft	Oracle	PostgreSQL Global Development Group
Год выпуска	1989	1995	1989
Реализовано на языке	C++	C и C++	C
Операционные системы сервера	Linux, Windows	FreeBSD Linux OS X Solaris Windows	FreeBSD HP-UX Linux NetBSD OpenBSD OS X Solaris Unix Windows
Схематичные данные	Да	Да	Да
Предопределение типа данных	Да	Да	Да
Поддержка XML	Да	Да	Да, но нет функционала очереди
SQL	Да	Да, с проприетарными расширениями	Да, с многочисленными расширениями
API и другие методы доступа	ADO.NET JDBC ODBC OLE DB Tabular Data	ADO.NET JDBC ODBC Proprietary	ADO.NET JDBC native C library

	Stream (TDS)	native API	ODBC streaming API for large objects
Поддерживает язык программирования	C# C++ Delphi Go Java JavaScript (Node.js) PHP Python R Ruby Visual Basic	Ada C C# C++ D Delphi Eiffel Erlang Haskell Java JavaScript (Node.js) Objective-C OCaml Perl PHP Python Ruby Scheme Tcl	.Net C C++ Delphi Java info JavaScript (Node.js) Perl PHP Python Tcl
Скрипты на сервере	Transact SQL, .NET languages, R, Python and (with SQL Server 2019) Java	Собственный синтакс	реализовано на проприетарном языке PL/pgSQL или на распространенных языках, таких как Perl, Python, Tcl и т. д.
Методы разбиения	Таблицы могут быть распределены по нескольким файлам (горизонтальное разбиение)	Горизонтальное разбиение, сегментирование с помощью MySQL Cluster или MySQL Fabric	Разбиение по диапазону, списку и (начиная с PostgreSQL 11) по хешу
Методы репликации	Да, но не во всех версиях	Да, репликация с несколькими источниками, репликация источника-реплики	Да, репликация источника-реплики. Другие возможны, при сторонних расширениях
Использование внешних ключей	Да	Да, но не для версии MyISAM	Да
Поддержка обеспечения целостности данных после неатомарных манипуляций с данными	ACID (<i>Atomicity, Consistency, Isolation, Durability</i>) - набор требований к транзакционной системе, обеспечивающий наиболее надёжную и предсказуемую её работу - атомарность,	ACID, но не для версии MyISAM	ACID (<i>Atomicity, Consistency, Isolation, Durability</i>) - набор требований к транзакционной системе, обеспечивающий наиболее надёжную и предсказуемую её

	согласованность, изоляция, устойчивость.		работу - атомарность, согласованность, изоляция, устойчивость.
Поддержка одновременной обработки данных	Да	Да, блокируются таблицы или строки, зависит от метода хранения	Да
Поддержка долгосрочного хранения данных	Да	Да	Да
Возможность определить структуры, которые будут храниться	Да	Да	Нет

SQL

SQL - это структурированный язык запросов, созданный для того, чтобы получать из базы данных необходимую информацию. Если описать схему работы SQL простыми словами, то специалист формирует запрос и направляет его в базу. Та в свою очередь обрабатывает эту информацию, «понимает», что именно нужно специалисту, и отправляет ответ.

Данные хранятся в виде таблиц, они структурированы и разложены по строкам и столбцам, чтобы ими легче было оперировать. Такой способ хранения информации называют реляционными базами данных (от англ. *relation* — «отношения»). Название указывает на то, что объекты в такой базе связаны определенными отношениями.

SQL - это не язык программирования, , но при этом внутренняя работа сайта (backend) невозможна без запросов.

SQL используют разные виды специалистов:

- **Аналитики и продуктовые маркетологи.** Знание SQL помогает этим специалистам не зависеть от программистов, а самостоятельно получать и обрабатывать данные;
- **Разработчики и тестировщики.** С помощью SQL они могут самостоятельно проектировать базы для быстрой и надежной работы с

данными, улучшать с их помощью сайты и приложения;

- **Руководители и менеджеры.** SQL позволит специалистам на руководящих постах самостоятельно обращаться к базам, контролировать работу компании и в реальном времени получать данные о положении дел.

SQL - операторы:

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:

- **SELECT** - выбор данных в соответствии с условием;
 - **SELECT** - выбери данные;
 - **FROM** - вот отсюда;
 - **JOIN** - добавь еще эти таблицы;
 - **WHERE** - при таком условии;
 - **GROUP BY** - сгруппируй данные по этому признаку;
 - **ORDER BY** - отсортируй данные по этому признаку;
 - **LIMIT** - нужно такое количество результатов;
 - **;** - конец предложения.
- **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:

- **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:

- **BEGIN TRANSACTION** - обозначение начала транзакции;
- **COMMIT TRANSACTION** - изменение команд внутри транзакции;
- **ROLLBACK TRANSACTION** - откат транзакции;
- **SAVE TRANSACTION** - указание промежуточной точки сохранения внутри транзакции.

DB Testing:

Как при ручном тестировании, так и при автоматическом могут использоваться прямые запросы в БД. Существуют разработчики и тестировщики баз данных. А иногда, мы можем тестировать базу данных отдельно от всего приложения, как отдельный компонент системы.

- Возможность сравнить реально существующие данные в базе с теми, которые отображаются в приложении, которое мы тестируем. Ведь отображение это дополнительная прослойка, плюс запрос в коде может быть реализован некорректно;
- Возможность напрямую загружать свои тестовые данные в БД, чтобы проверить какие-то сценарии. Например, если создание данных ещё не реализовано или для ускорения тестирования (быстрее добавить напрямую, чем через возможности приложения);
- Возможность создания среды тестирования;
- Возможность данные удалять и редактировать напрямую в БД;
- Тестирование безопасности посредством SQL инъекций.