

## What is a database?

Let's talk about kitchen, imagine you are cooking Maggi. I am sure everyone has prepared Maggi at least once in their life. But today we are going to prepare a little advanced version of Maggi, not just the boiled stuff, but we will try to make it a little differently, where we will add other ingredients, such as we will add onions, we will add some cucumber, capicum, carrots, we will add some spices for example we will add some black pepper, coriander powder, a little bit of salt, cumin powder etc. etc. and cook it in oil

Now these vegetables that we are going to use, where are we supposed to have them stored? - The Refrigerator?

Where are we going to find the spices that we are planning to use? - The spice rack?

We will need a pot to boil the maggi and a frying pan to cook it right? Where will they be stored? - In the vessels storage rack?

Now imagine if these ingredients that we need to use while cooking are not organised what will happen??

It will slow down our cooking process?? We will have to first search for them, find them from wherever they have been kept?

So, see in our normal lives also we like to keep our things organised why because it makes it easier to locate the things which increases our overall efficiency?

Again when we talk about organising, you see we have a Refrigerator - where we usually store food items, Spice rack to store our spices, Vessels Rack to store the vessels?

We don't store spices inside the refrigerator do we?

So, if we see Refrigerator is the database for food items, similarly spice rack is the database for storing spices, vessels rack for storing the vessels.

Which means we are storing all similar type of items together right? This is exactly how we define a database

Let's quickly take a look at the definition of a database - The definition says that database is a collection of related Data.

link: <https://www.oracle.com/in/database/what-is-database/>

A database is an organized collection of structured information, or data, typically stored electronically in a computer system. So what is a database essentially help us with? It helps us with organising the data, which kind of

data, similar type of data, so that it can increase the efficiency of retrieval.

Now let's talk from an application standpoint - do you like visiting websites which takes a lot of time to load?? For example: I am sure you have used government websites or college websites to retrieve your semester mark sheets or admit cards. And usually it can become very annoying at times because they take a lot of time to load. And if you see these informations or records are stored in some sort databases definitely, as we spoke in our previous classes that these data should be saved or persisted somewhere so that they can be retrieved on time. Thus, one of the most probable reasons that these records take such a long time to download could be because they are not stored efficiently or in the right kind of database because of which the entire process of retrieving the file takes such a long time.

### **What is a Database Management System?**

Now coming back to our kitchen. I am sure all of us have done grocery shopping from the market and then once we return home, we have handed over the bag to our mothers and our job kind of finishes there right? The next thing we know is that the items are properly organised and kept where they should be. But the person who does the job of organising these stuffs, when they see the item they would know exactly where they should be kept so that the next time the item needs to be used, it can be retrieved in the quickest and the most efficient way possible, also sometimes it happens that only a portion of the item is used for example the spices and they have to be stored back in the right place. So, this job belongs to the organiser who takes care of organising the stuffs and placing them in its desired location. Also the organiser makes sure that items are not placed where they should not be, basically protecting the integrity of the item.

Now let's take a look at the the definition of DBMS

A database typically requires a comprehensive database software program known as a database management system (DBMS). A DBMS serves as an interface between the database and its end users or programs, allowing users to retrieve, update, and manage how the information is organized and optimized.

Now let's talk about Databases in terms of Computers

### **Types of Databases**

#### **File System**

We often try to organise and keep things in a file system right? For example let me quickly share my screen and show you that I have been organising files for my classes. So basically I am trying to organise these and keep them properly.

Now the problem with storing files is a file systems could be retrieval of a particular data from inside a particular file. Once these start growing in size, it may be very difficult to query it and querying these would take a lot of time. For example: imagine after taking 20-25 classes, if I were to search for the class notes from all classes that I have taken so far which consists of the keyword say "html", it can be a tedious job to do that and it may end up taking a lot of time to retrieve the same thing..

## Spreadsheets

[https://docs.google.com/spreadsheets/d/1D4w3CtHqKPPQpZF5\\_ZHTT6jEVm5\\_2NYX94nFduRHINY/edit#gid=0](https://docs.google.com/spreadsheets/d/1D4w3CtHqKPPQpZF5_ZHTT6jEVm5_2NYX94nFduRHINY/edit#gid=0)

- Data redundancy - In file based databases, a lot of time we lead towards addition of redundant data. For example: In an organisation, multiple teams can work on similar data that can lead to storage of redundant data.
- Difficult to access data - In order to access data from a file based database we need to know the exact location of the data in the spreadsheet. If there is a huge chunk of data stored in a spreadsheet this can be a very tedious task.
- Data inconsistency - With time, if due to data redundancy there exist multiple different copies of the same data then it can lead to a situation when a data is not updated in all the copies leading to inconsistency.
- Security issues - It is difficult to control unauthorised access to these files.
- Concurrent access issues - Handling updation of data by multiple users at same point in time is generally not available because if one user is updating a cell, it is generally locked and no one else can access it.

## Relational Databases and

### Relational Database Management System

Represent informations in the form of tables. Each row is called a tuple and columns are known as attributes.

Each table usually contains data that are of the same type.

And each row in a table can be uniquely identified by what we call as a Primary Key.

But that's not the actual power of Relational Databases. Lets talk a little more in details with an example.

## Relational Database

Let's talk about the structure of a relational database system.

Now why is this way of structuring the data beneficial to certain applications?? Well if you see, you can avoid storing the same redundant data over and over again in each and every row, so it saves a lot of disk space.

What can be another benefit of storing data in this way?? Can someone tell me what is another benefit of storing data in separate tables instead of having them in the same table?

Just imagine that one of these redundant entries, for example the department name, it's changed from IT to information technology, that has to be changed in all the different tables where it would have been referenced otherwise to maintain integrity which can be a very tedious task, instead now we will only have to update in the department table and automatically, every time it gets referenced it will return the updated value.

Now one downside of relational databases is that it requires joins. That is joining two tables to get the complete data and that can be quite time consuming specially if the tables are growing large in size.. There are also other problems with traditional relational databases that is they can't be shared, meaning imagine the size of the table grows so much that you can't store it in one single computer and needs to be split across multiple computers, because if we do that then joins might not return the correct result. Because for a join to work correctly all of the data of both tables should reside on the same computer. Although these days there are upcoming distributed relational databases such as Cockroach DB, Google Spanner, which lets us perform joins across multiple computers. they are also known as New Sql.. Anyways these are just FYI and don't have to break your head on at this point.

### Graph Databases

Next let's talk about graph databases.

Let me quickly share my screen and fire up LinkedIn

<https://www.linkedin.com/in/nayanavade/>

Search for Meta, click on connections and show the 1st, 2nd, 3rd+ These numbers basically mean how far I am related to a person in terms of connections. In other words, for example when the count is 2nd it means that we both are not connected but we have mutual connections. When the count is 3rd, it means that we both have a connection which has a mutual connection but we both don't have any direct mutual connection. So basically, does it make sense that these are levels..

Similarly, if I talk in terms of family, we can say that I am directly connected to my parents

*if I were to represent in a diagram  
it should be*

So basically we are able to treat each person as a node and their relationships can be represented using edges. When we study about trees in our basic data structure classes you will be able to get a better picture of the same.

### Key Value Databases

Key Value databases don't usually have a table like structure because it is usually only a single key and a single value. They are more like data structures than databases. In our upcoming classes we will see a data structure called Map in our upcoming classes. I will quickly show a demo, for non-technical students present in this class, please don't worry about the syntax because we will learn all of it in the coming classes.

Let me quickly show a demo on the Node terminal to show how a key-value data structure looks like. This is almost how a key value store is implemented and are preferably stored in the RAM or the primary memory of the computer so that retrieval of the data can be very fast.

Some examples are: Redis, Memcached, Aerospike

### Document Databases

The primary idea of a document database is also pretty similar to a key-value store only. All that do is instead of the value being a single object, we can dump an entire document as a value in the database and then we can perform operations on the objects present in the value as well.

For example: We can dump an object something like

A screenshot of a terminal window titled 'anki — mongo — 89x17'. The terminal shows a MongoDB query command and its result. The command is 'db.mystudentData.find().pretty()'. The result is a JSON array containing two document objects. The first document has fields: '\_id' (ObjectId), 'name' (Sumit), 'branch' (CSE), 'course' (DSA), 'amount' (4999), and 'paid' (Yes). The second document has fields: '\_id' (geeks\_for\_geeks\_201), 'name' (Rohit), 'branch' (ECE), 'course' (Sudo Gate), and 'year' (2020).

```
> db.mystudentData.find().pretty()
[
  {
    "_id" : ObjectId("5e37b67303ab1253cde7afe6"),
    "name" : "Sumit",
    "branch" : "CSE",
    "course" : "DSA",
    "amount" : 4999,
    "paid" : "Yes"
  },
  {
    "_id" : "geeks_for_geeks_201",
    "name" : "Rohit",
    "branch" : "ECE",
    "course" : "Sudo Gate",
    "year" : 2020
  }
]
```

And it also allows us to query on each of the individual keys present in the document. One example of document database is MongoDB which we will study in our course.

## What is a Database Schema

Database schema is actually a kind of structure that we define around the data that we are going to store in that particular table and this would include the table itself, views, fields, relationships, keys etc. etc.

Now imagine if we were given some data that were all jumbled up together for example data of emails, sales, finance data, call centre data, chat data, rating and reviews data in one single place? We would definitely go crazy to organise them or structure them right?? That is why we need a database schema.

I am not sure but I am hoping that people from the technical background would have heard of this term schema less when it comes to databases. For example: we can read here that MongoDB is a schema less database.

As a NoSQL database, **MongoDB** is considered schemaless because it does not require a rigid, pre-defined schema like a relational database. The database management system (DBMS) enforces a partial schema as data is written, explicitly listing collections and indexes.

<https://www.mongodb.com/unstructured-data/schemal...>

[Schemaless Database | MongoDB](#)

Well to be honest there is nothing as such called a schema less when we are designing for an application.

The code that reads from the table will already have some kind of structure, yes the database doesn't impose any strict restrictions that a schema should be defined from before hand but in reality we have a schema or a sctructure on the code and this called as **Schema-On-Read**.

For example: Let's see this image here once more that we saw for document databases



```
anki — mongo — 89x17
> db.mystudentData.find().pretty()
{
  "_id" : ObjectId("5e37b67303ab1253cde7afe6"),
  "name" : "Sumit",
  "branch" : "CSE",
  "course" : "DSA",
  "amount" : 4999,
  "paid" : "Yes"
}
{
  "_id" : "geeks_for_geeks_201",
  "name" : "Rohit",
  "branch" : "ECE",
  "course" : "Sudo Gate",
  "year" : 2020
}
>
```

And if we take a look at some code that is written using Mongoose which is the

javascript client for MongoDB. Javascript client meaning the mediator through which our application can communicate with the MongoDB database.

```
1  import mongoose from "mongoose";
2
3  const tweetSchema = new mongoose.Schema({
4    text: {
5      type: String,
6      required: true,
7      maxLength: 120
8    },
9    status: {
10     type: String,
11     enum: ['draft', 'published'],
12     default: 'draft'
13   },
14   user: {
15     type: mongoose.Schema.Types.ObjectId,
16     ref: 'user'
17   }
18 }, {timestamps: true});
19
20 export const Tweet = mongoose.model('tweet', tweetSchema);
```

However, when we talk about Relational Databases they have a strict predefined schema requirement, meaning we have to define a structure before we can store any data in the tables. This is known as **Schema-on-write**

**Student Table**

ID	First_Name	Last_Name	Class	Major
----	------------	-----------	-------	-------

**Course Table**

Course_ID	Course_Name	Course_credits
-----------	-------------	----------------

**Department Table**

Department_Code	Department_Name
-----------------	-----------------

**Instructor Table**

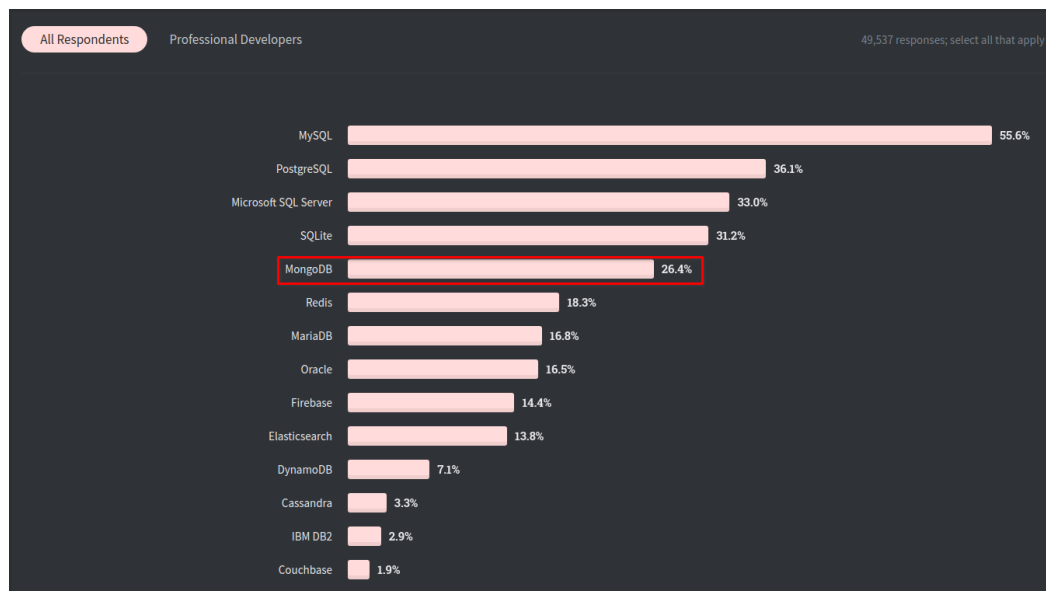
Instructor_ID	Instructor_fname	Department_Code
---------------	------------------	-----------------

**Grade Table**

ID	Course_ID	Grade
----	-----------	-------

**Different Famous Available****Databases**





The following image is the survey result of the year 2020 conducted by stack overflow, about the most popular databases.  
Let's discuss about a few of them in brief

## MySQL

- It is an open source RDBMS
- Applications like fb and uber uses mysql
- It provides high consistency
- High Data Security
- Can handle write heavy use cases also
- Highly scalable and available
- The community edition is the most widely used free db in the industry
- Used widely in manufacturing, banking, and financial services

## PgSQL

- Open source RDBMS
- Gives more advanced features compared to other rdbms like partial index, bloom filter.
- We can store structured data, unstructured data (JSON), key value data, spatial data etc
- It can handle millions of transactions
- It can handle geospatial data
- Helpful in financial industry, Storing GIS data

## MongoDB

- Open core nosql database
- Supports Map-Reduce query, text search, graph search, and geo-search.
- Offers full text search engine
- Helpful in fast application development where schema is continuously changing.
- Provides more consistency when compared to availability
- Not preferred in case of structured data

- Helpful in product catalogue storage in ecommerce web apps, storing data in social media apps, etc

## Redis

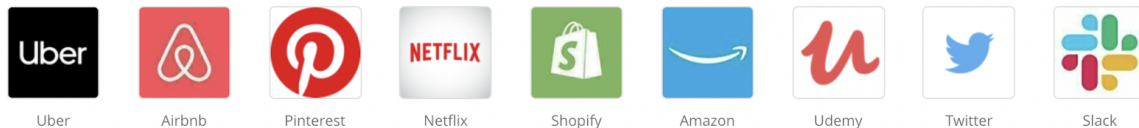
- Distributed in memory key value data store
- Works on nosql data structure principle
- Works as cache
- Highly scalable
- Helpful in real time use cases like inventory systems
- Not helpful in graph like data
- Helpful in real time analytics, gaming leaderboards, caching, messaging queues etc.

Link: [stackshare](#)

### Who uses MySQL?

#### COMPANIES

5571 companies reportedly use MySQL in their tech stacks, including Uber, Airbnb, and Pinterest.



### Who uses Oracle?

#### COMPANIES

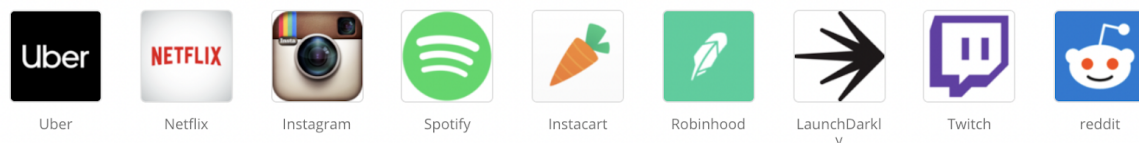
229 companies reportedly use Oracle in their tech stacks, including Netflix, LinkedIn, and ebay.



### Who uses PostgreSQL?

#### COMPANIES

5474 companies reportedly use PostgreSQL in their tech stacks, including Uber, Netflix, and Instagram.



## Who uses MongoDB?

### COMPANIES

4142 companies reportedly use MongoDB in their tech stacks, including Uber, LaunchDarkly, and Delivery Hero.



Uber



LaunchDarkly



Delivery Hero



Lyft



KAVAK



HENNGE K.K.



Stack



Accenture



Hepsiburada

## Who uses Cassandra?

### COMPANIES

507 companies reportedly use Cassandra in their tech stacks, including Uber, Facebook, and Netflix.



Uber



Facebook



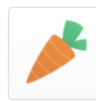
Netflix



Instagram



Spotify



Instacart



reddit



Accenture



Hepsiburada

## Intro to Sql

SQL, also referred to as structured query language, is a computer language, for storing, retrieving and manipulating data from a relational database.

Sql became popular because:

- It can create the table structures and database structures.
- Perform basic database operations
- Can perform complex queries as well.

Let's try out a few simple SQL queries for Relational Databases here

Create Table Query - [https://www.w3schools.com/sql/sql\\_create\\_table.asp](https://www.w3schools.com/sql/sql_create_table.asp)

Insert Query - [https://www.w3schools.com/sql/sql\\_insert.asp](https://www.w3schools.com/sql/sql_insert.asp)

Select Query - [https://www.w3schools.com/sql/sql\\_select.asp](https://www.w3schools.com/sql/sql_select.asp)