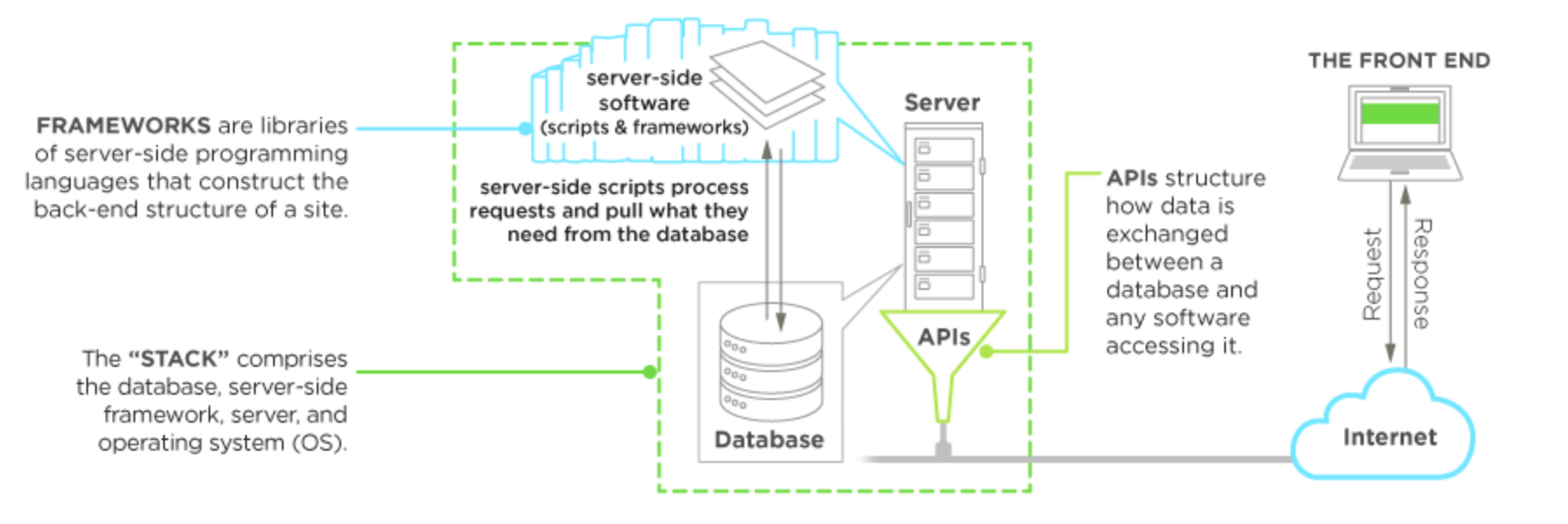
Backend

The traditional backend is a mix of the server, database, API and operating systems that power the applications front end.

The backend of an application can look very different from application to application whether it's the use of a cloud-based-server and data warehouses, containerization with a service like docker, backend-as-a-service(BaaS), or APIs to replace more complex processing.



PHP

When it comes to backend technologies PHP is one of the most common and most popular languages. It has a syntax very similar to C or Java.

Python

Python is a languages that users find simpler than PHP. It is designed to have a very readable code.

Its well tested and google choose to develop their services with it.

One of the most popular framework for python id Django.

Ruby

Ruby is designed to be a run language. It has a focus on simplicity and productivity with an syntax.

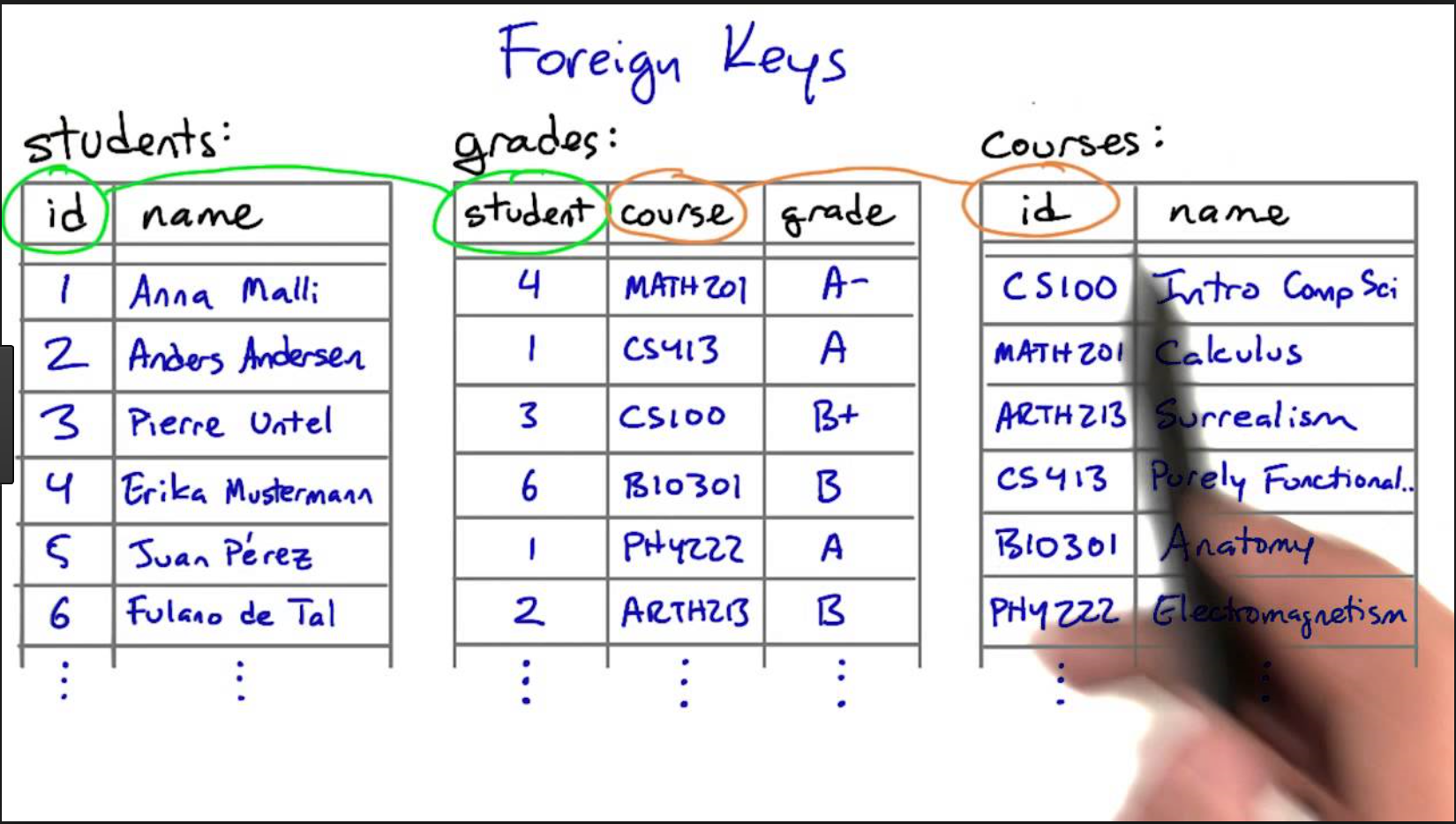
In ruby everything is an object and that's interesting because it encourages the programmer to think this way when developing.

The most popular framework for ruby on rails.

Databases

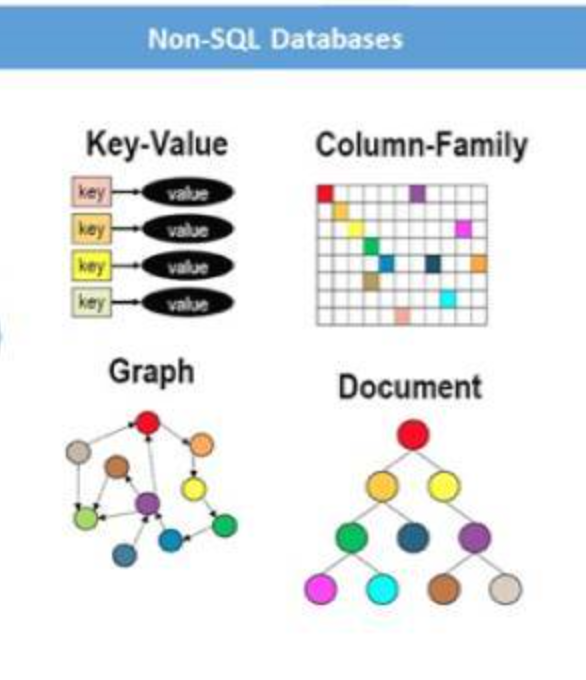
I will be covering two different types of databases Relational and Non-Relational databases.

Relational databases use SQL which stands for Structured Query Language. SQL is used to communicate with a database. It is the standard for relational database management systems. SQL statements are used to perform tasks such as update data or retrieve data from a database.



Non-Relational databases encompasses a wide variety of different database technologies that were developed in response to the demands presented in building modern applications.

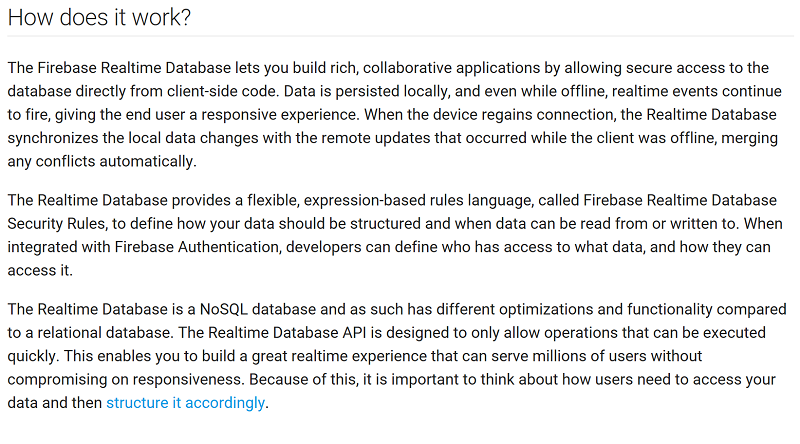
* Developers are working with application that create massive volumes of new rapidly changing data types structured, semi-structured, unstructured and polymorphic data.
* Applications that once served a finite audience are now delivered as services that must be always-on accessible from many different devices and scaled globally to millions of users.

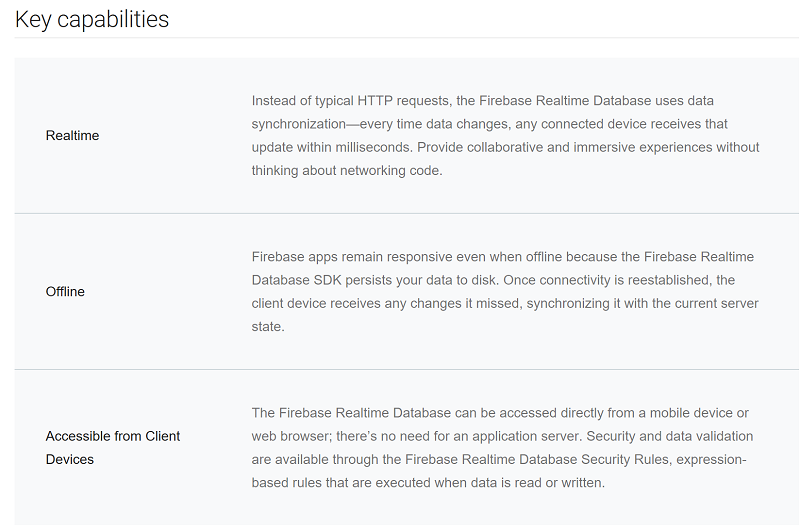


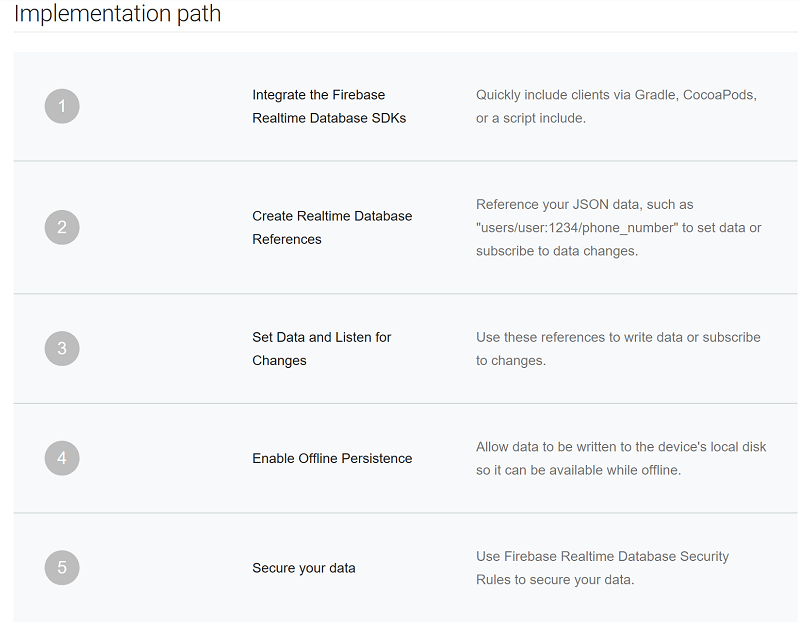
Firebase

Firebase is a NoSQL cloud database data is synced across all clients in real time and remains available when a device goes offline.

Data is stored as JSON and synced in realtime to every connected client if you build cross-platform applications with IOS, Android, and Javascript SDKs all of your clients share one real time database instance and will automatically retrieve updates with the newest data.







Good :

* Can adapt to incoming data
* Grows with your application
* Powered by google AUTH

Bad :

* Relatively new
* Structuring data at beginning can be time consuming

SQLite

SQLite is an in-process library that implements a self-contained serverless, zero-configuration, transactional SQL database engine.

SQLite is an embedded SQL database engine unlike most other SQL databases SQLite does not have a separate server process. SQLite read and writes directly to ordinary disk files this means that a complete SQL database with multiple tables, indices, triggers, and views is contained in a single disk file.

SQLite has been very carefully tested prior to every release and has a reputation for being very reliable. Most SQLite source code is devoted purely to testing and verification an automated test suite runs millions and millions of test cases involving hundreds of millions of individual SQL statements and achieves 100% branch test coverage.

Good :

* A complete database is stored in one file.
* Intensive testing
* Good documentation

Bad :

* Database stored user device
* Fixed size

Backend : <https://davidmles.com/backend-technologies/>

<https://www.upwork.com/hiring/development/a-beginners-guide-to-back-end-development/>

Databases firebase : <https://firebase.google.com/docs/database/>

SQL : <https://www.sqlite.org/>

What is SQL : <http://www.sqlcourse.com/intro.html>

Image 1 <https://www.google.ie/search?q=Relational+databases&source=lnms&tbm=isch&sa=X&ved=0ahUKEwiD6JXM-6XXAhWKfhoKHZygDdgQ_AUICigB&biw=1368&bih=762#imgrc=fXQqS7bHTOCsGM>: