# Patience is not a Virtue

In a world of technological abundance, it can be quite a challenge to pick the right **database**.

Whether you are building an application from scratch or expanding the functionality of the existing app, your choice of DB can either make it or break it.

From performance to programmability, the right database will not leave you waiting on its latency. This is significant because unnecessary waiting turns patience from a virtue into foolishness.

https://media.giphy.com/media/UR2r2iWXjq65O/giphy.gif

## History of EDB

In the past, the term ‘embedded database’ (EDB) was used to describe a database system that existed *as part of an application’s code*.

Today, an ‘**embedded DB**’ refers to when a database lives in your app ***and*** is synced with the same **server**. Meaning, that if your application is running in container/server A (for example), the database will also run in this location. Therefore, all of the database queries and updates that need to be performed are local.

Structurally, they may be relational databases or **non-relational databases.**

https://media.giphy.com/media/405kIEbYhpP0GHjSOK/giphy.gif

**EDB Features**

EDBs come complete with many advantages, including:

- **Reduced latency;**
- Network I/O load from network calls between the application and the database;
- Maximizing the potential of RDBMS features;
- Facilitating **data export** in a flexible manner;
- Granting access to your processed data to external clients;
- Hiding a database, since it's functionally invisible to the users.

## To Embed or not to Embed

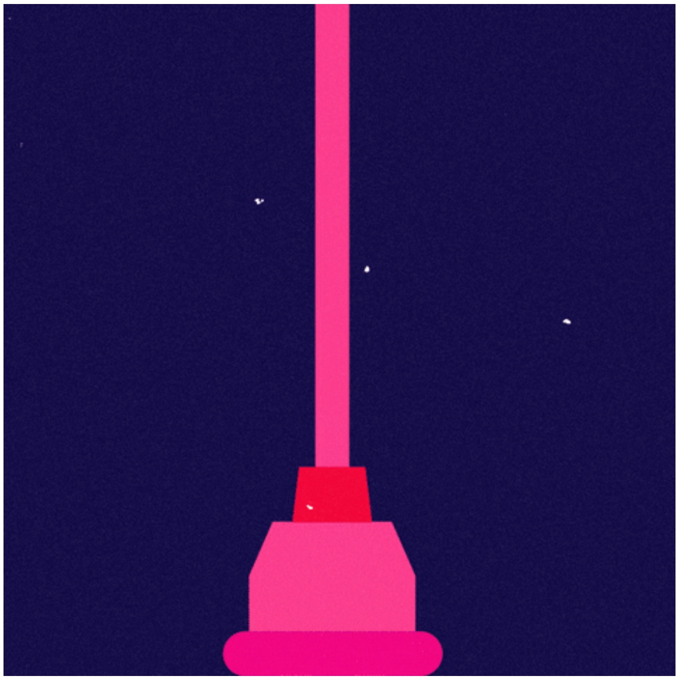
**Embedded databases** are ideal for mobile applications for a plethora of reasons.

The **reduced latency** in EDBs leads to faster data access providing a superior user experience.

Furthermore, having offline access can come in handy. This includes instances when your application can afford to store data locally without transmitting it across the network.

Many people have used these types of databases, especially in organizations working with aid and relief initiatives (such as refugee camps). Working in refugee camps without an infrastructure or a cell phone network is quite a challenge from a technical standpoint.

Thanks to the capabilities of embedded DBs, organizations who employ EDBs can have their workers go offline (to collect data on sanitation or health for example) and then put all their data together using the local WiFi. It's as simple as that.



https://media.giphy.com/media/xKRb2BlJ1CgYfrNFbu/giphy.gif

Moreover, one of the biggest challenges that **developers** have when they're creating **mobile apps** is keeping the local data in sync with whatever's on the server. In this case, embedded databases eliminate the latency of having to communicate with the server all of the time.

## Wrapping Up

An embedded database is the right choice when your application calls for quick data processing. This stand-alone solution happens to be elegant, sparing you annoying lags down the road.

*Join us today to thank our newsletter sponsor, Couchbase. In an increasingly virtual world, companies operating on cloud services need to be able to scale fast without adding extra lag time for their users.  Couchbase can serve as a reliable system of record, handling key-value operations in microseconds latency, SQL queries and text searches in milliseconds, and Ad-hoc analytical queries spanning tens of seconds, one not impeding the other. Couchbase is an award-winning, distributed cloud database that offers reliable services critical for business applications.*

***

**Got a tech story to share with our readers?** Everything you've ever wanted to know about how to get published on Hacker Noon - **get it here**