

[Open in app](#)

Christoph Bussler

Jul 17 · 2 min read · [Listen](#)

Save



Database Theorems

Recently I came across a specific microservices pattern, and that in turn during research led me to a database theorem. I did not know it, so I decided collecting database theorems in one place.

If you know of an additional database related theorem that is missing here, let me know and I'll add it (with acknowledgement, of course).

What is a database theorem?

A database theorem is essentially a non-functional property trade-off (system or operations related) where specific properties cannot be provided at the same time (or only to certain degrees).

Some theorems are core database theorems that apply to the architecture of a database system implementation itself, while others apply to overall architectures (like application architecture) as well, not just database system implementations.

CAP Theorem (Brewer's Theorem)

The CAP theorem is one of the most known theorems: "... any distributed data store can only provide two of the following three guarantees: [...] Consistency [...] Availability [...] Partition tolerance" (https://en.wikipedia.org/wiki/CAP_theorem).

BAC Theorem

The BAC theorem appeared on my radar in context of researching microservice



[Open in app](#)

As a clarifying note, this applies when the microservice architecture deploys two or more independent database systems. This is the case when the architecture follows the principle that each microservice encapsulates and abstracts from its data in its own database system. (I have to discuss a few aspects of this pattern, and that'll come up in a separate blog later).

PIE Theorem

“The PIE theorem posits that we can choose two out of three desirable features in a data system: Pattern Flexibility [...] Efficiency [...] Infinite scale [...]” (<https://learning-notes.kovacevic.dev/Databases/Theory/PIE-Theorem>).

PACELC Theorem

The PACELC theorem is not as known as CAP, but for global or wide-area applications it is more relevant: “... in case of network partitioning (P) in a distributed computer system, one has to choose between availability (A) and consistency (C) (as per the CAP theorem), but else (E), even when the system is running normally in the absence of partitions, one has to choose between latency (L) and consistency (C)” (https://en.wikipedia.org/wiki/PACELC_theorem).

This theorem is very applicable to application architectures that are “global”, meaning, in the context of public clouds, one state managing application is running in several regions in different geolocations that are far apart from each other (e.g., Asia, Europe and USA) without the ability to strictly partition data by region or geolocation.

Request: if you know of additional database related theorems ...

... please let me know and I'll include those. Thank you in advance.





Open in app

