

# ESCOLA E FACULDADE DE TECNOLOGIA SENAI ROBERTO MANGE

## DESENVOLVIMENTO DE SISTEMAS

JULHO DE 2022 — SENAI-SP



ESCOLA SENAI "ROBERTO MANGE"  
UMA REALIZAÇÃO DA INDÚSTRIA

**SENAI**  
SÃO PAULO



# ORGANIZAÇÃO CURRICULAR

Unidades Curriculares <sup>6</sup>	Ano/Semestre (hora-aula de 50min)				Total (hora-aula de 50min)	Total (hora)
	1º ano		2º ano			
	1º	2º	3º	4º		
Linguagem de Programação	80				80	66,67
Banco de Dados	100	40			140	116,67
Tecnologia da Informação e Conectividade	100				100	83,33
Automação Industrial	100				100	83,33
Ciência de Dados	100				100	83,33
Desenvolvimento Web		100	100	80	280	233,33
Desenvolvimento Mobile		80	100	100	280	233,33



# ORGANIZAÇÃO CURRICULAR

Unidades Curriculares <sup>6</sup>	Ano/Semestre (hora-aula de 50min)				Total (hora-aula de 50min)	Total (hora)
	1º ano		2º ano			
	1º	2º	3º	4º		
Integração Vertical e Horizontal*		100			100	83,33
Integração com IIoT*		120			120	100,00
Inteligência Artificial e Big Data*		120			120	100,00
Cibersegurança*			100		100	83,33
Computação em Nuvem*			100		100	83,33
Interfaces Industriais			100		100	83,33



# ORGANIZAÇÃO CURRICULAR

Unidades Curriculares <sup>6</sup>	Ano/Semestre (hora-aula de 50min)				Total (hora-aula de 50min)	Total (hora)
	1º ano		2º ano			
	1º	2º	3º	4º		
Eletiva I			60		60	50,00
Engenharia de Software				100	100	83,33
Gestão de Projetos*				140	140	116,67
Gestão de Pessoas e Cultura Organizacional*				80	80	66,67
Eletiva II				60	60	50,00
Extensão Universitária Aplicada	60	60	60	60	240	200,00
<b>Carga Horária Total (hora-aula de 50min)</b>	540	620	620	620	2400	————
<b>Carga Horária Total (hora)</b>	449,99	516,67	516,67	516,67	————	2000



# ORGANIZAÇÃO CURRICULAR

<b>Eletivas</b>						
Empreendedorismo					60	50
Governança de TI					60	50
Planejamento Estratégico de TI					60	50
Chatbot					60	50
Realidade Aumentada (EaD)					60	50
<b>Optativas</b>						
Libras (EaD)					60	50
Estágio Supervisionado						400

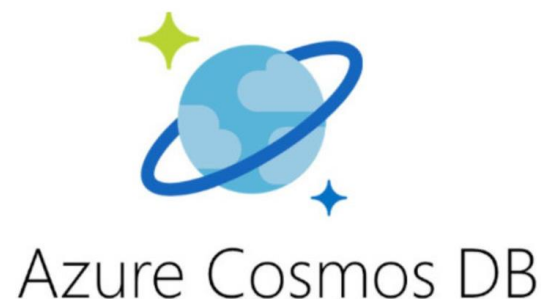


## 1 e 2º SEMESTRE – Banco de Dados: 140h (100+40)

Será abordado os bancos SQL e NoSQL principais, que são os mencionados abaixo, além de fornecer uma base sólida para criação do raciocínio da arquitetura inicial da base de dados para projetos de T.I.!

No 2º semestre deve-se integrar a base de dados já com as aplicações construídas em outras unidades curriculares (desenvolvimento web e mobile)

Será abordado a criação de bancos de dados em CLOUD usando os provedores Google Cloud, Azure e AWS!



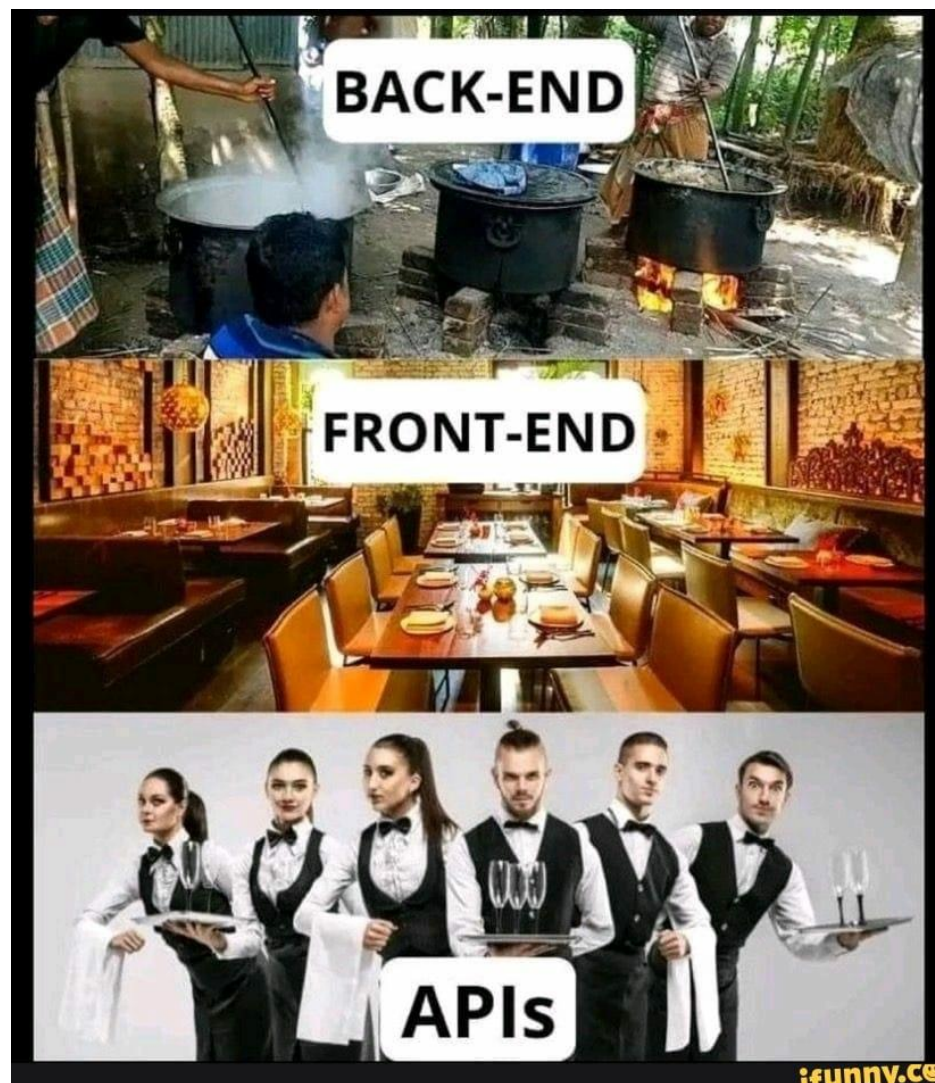


## FrontEnd x Backend

### Programming: BACKEND vs FRONTEND

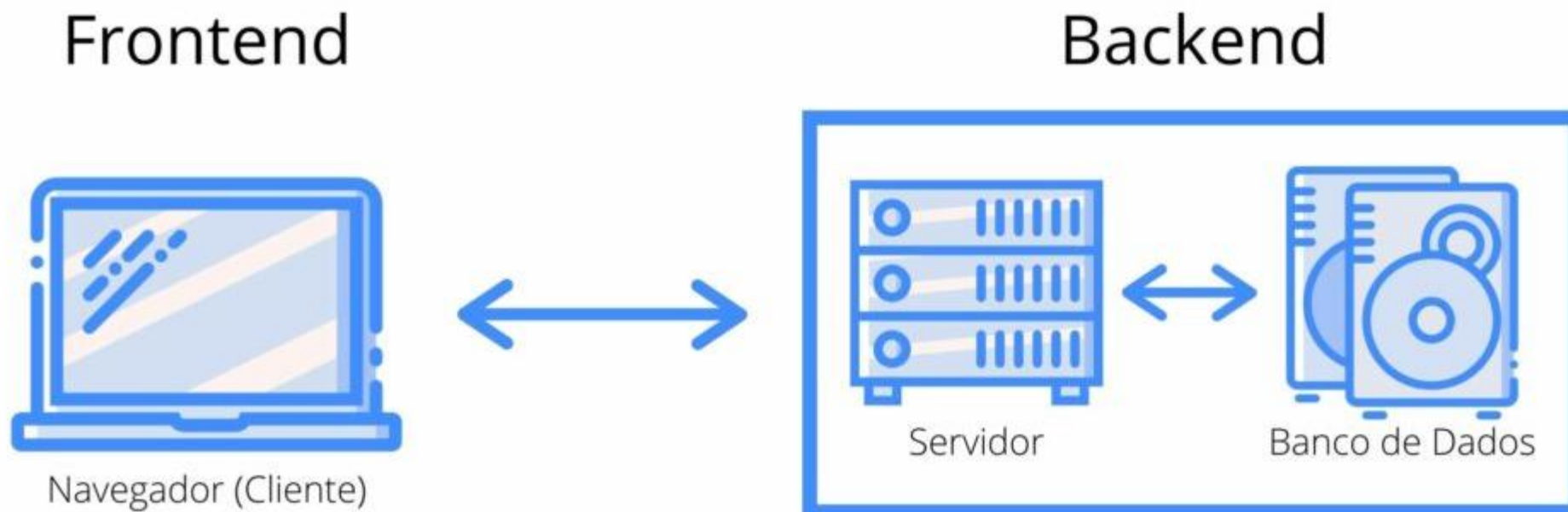


## FrontEnd x Backend





## Banco de Dados → Backend ou Frontend??



Normalmente, podemos considerar o projeto, interação e utilização do Banco de Dados pertencentes ao lado do backend por um motivo muito simples: segurança dos dados, não sendo recomendado que o Frontend faça alterações diretas no base de dados sem validações via código pertencente ao backend.

**Entretanto**, o fato de pertencerem ao lado do backend não quer dizer que obrigatoriamente devem estar no mesmo Servidor, isto é, na mesma máquina, sendo atualmente com a propagação do **Cloud Computing** que estejam em servidores distintos.



## Show! Então Banco de Dados é apenas um simples armazenamento, certo???



Certo, banco de dados é realmente armazenar dados para serem utilizados na aplicação, porém é necessário analisar COMO serão armazenados e estruturados tais dados pois decisões erradas em tal área podem prejudicar TODA A APLICAÇÃO, independente do quão bom estão o Frontend e o Backend!!!!



Existem várias categorias para decidir:  
SQL, NoSQL, NewSQL





## Existem várias categorias para decidir: SQL, NoSQL, NewSQL



- **SQL:** Structured Query Language, isto é, Linguagem de Consulta Estruturada, semelhante-se sua estrutura como a analogia de uma tabela em excel, por exemplo.
- **NoSQL:** Not Only SQL, ou seja, Não apenas uma Linguagem de Consulta Estruturada, tendo um conceito mais disruptivo e mais aberto do que a formalidade e padronização de uma tabela, por exemplo.
- **NewSQL:** Novo SQL, significando uma nova Linguagem de Consulta Estruturada, sendo uma união dos pontos positivos existentes no SQL junto com o NoSQL.



columns

## SQL



ORACLE

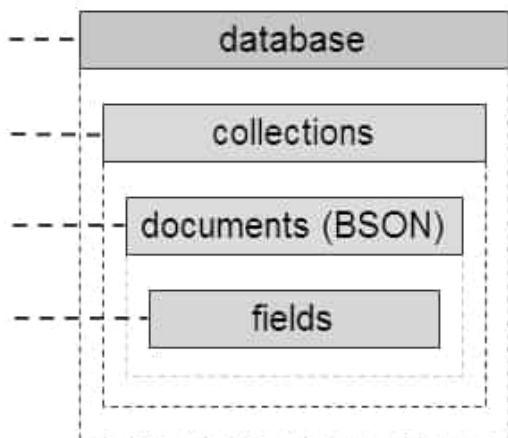


Oracle Sql Solutions



# NoSQL

## MongoDB Terms/Concepts



```
5 > grades : Array
6 cuisine : Lebanese
7 restaurant_id { 40357216
2 _id: "5cf0029caff5056591b0ce7d",
3 firstname: 'Jane',
```

String  
Number

```
1 _id : ObjectId('57717626280c7a2581558533')
2 > address : Object
3 borough : Bronx
```

String

4 name : MongoDB Burger Joint |

String

```
5 > grades : Array
6 cuisine : American
7 restaurant_id : 40357217
```

String  
Number

```
1 _id : ObjectId('57717626280c7a2581558534')
2 > address : Object
3 borough : Queens
```

String

# NoSQL

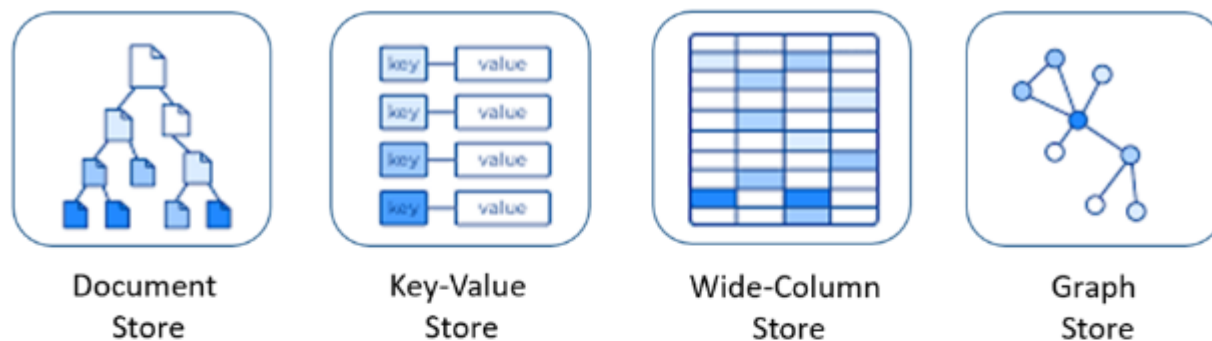


Figura 5-9: modelos de dados para bancos de dados NoSQL

Modelar	Características
Repositório de documentos	Os dados e os metadados são armazenados hierarquicamente em documentos baseados em JSON no banco de dados.
Key Value Store	O mais simples dos bancos de dados NoSQL, que são representados como uma coleção de pares chave-valor.
Wide-Column Store	Os dados relacionados são armazenados como um conjunto de pares de valor/chave aninhados em uma única coluna.
Repositório de grafo	Os dados são armazenados em uma estrutura de grafo como propriedades de nó, borda e dados.





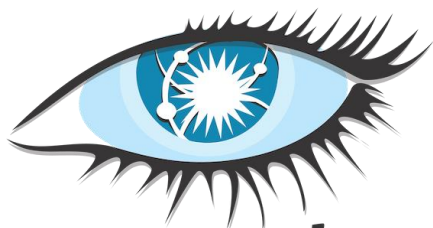
## NoSQL



mongoDB®



Firestore  
Realtime Database



cassandra



Azure Cosmos DB



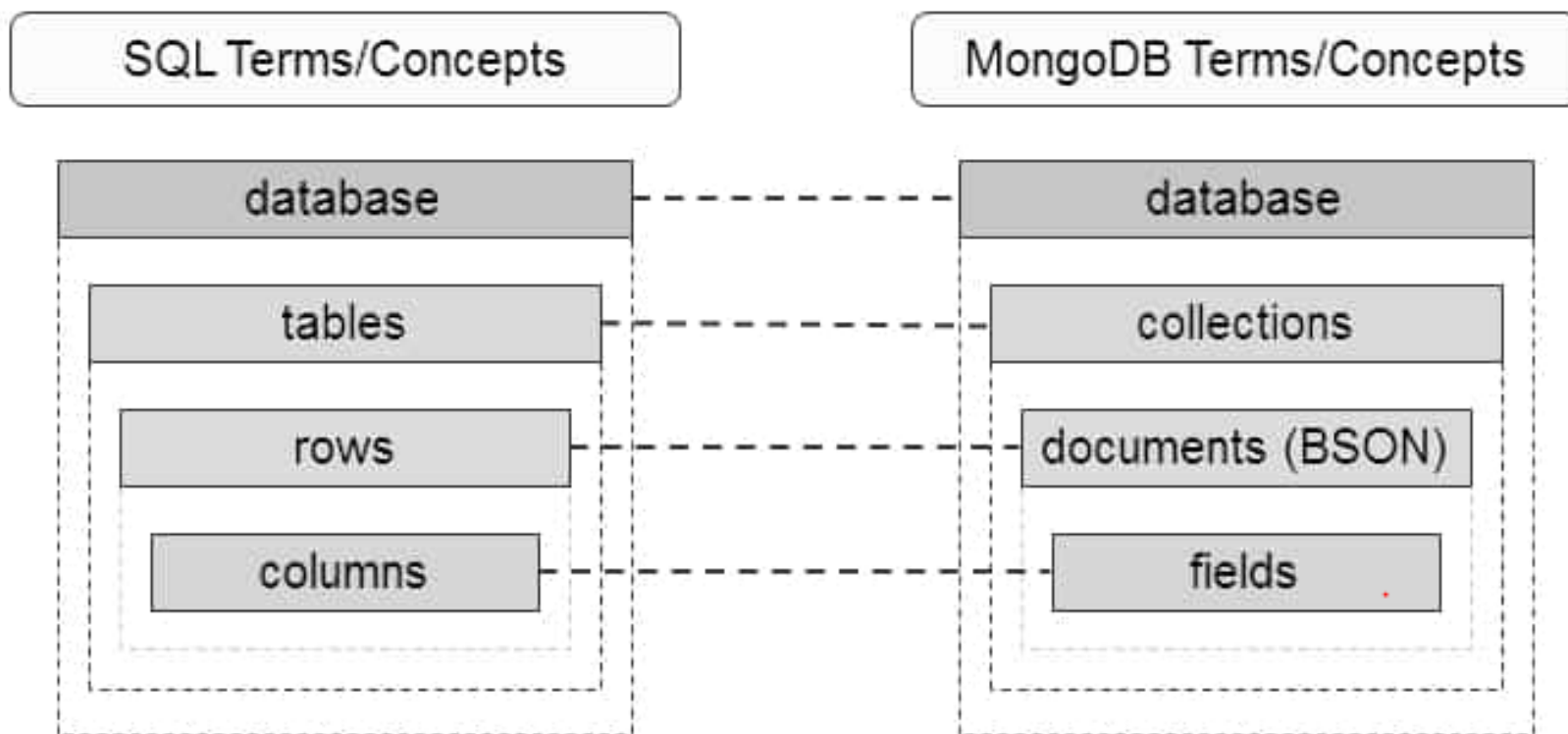
DynamoDB



EROSPIKE

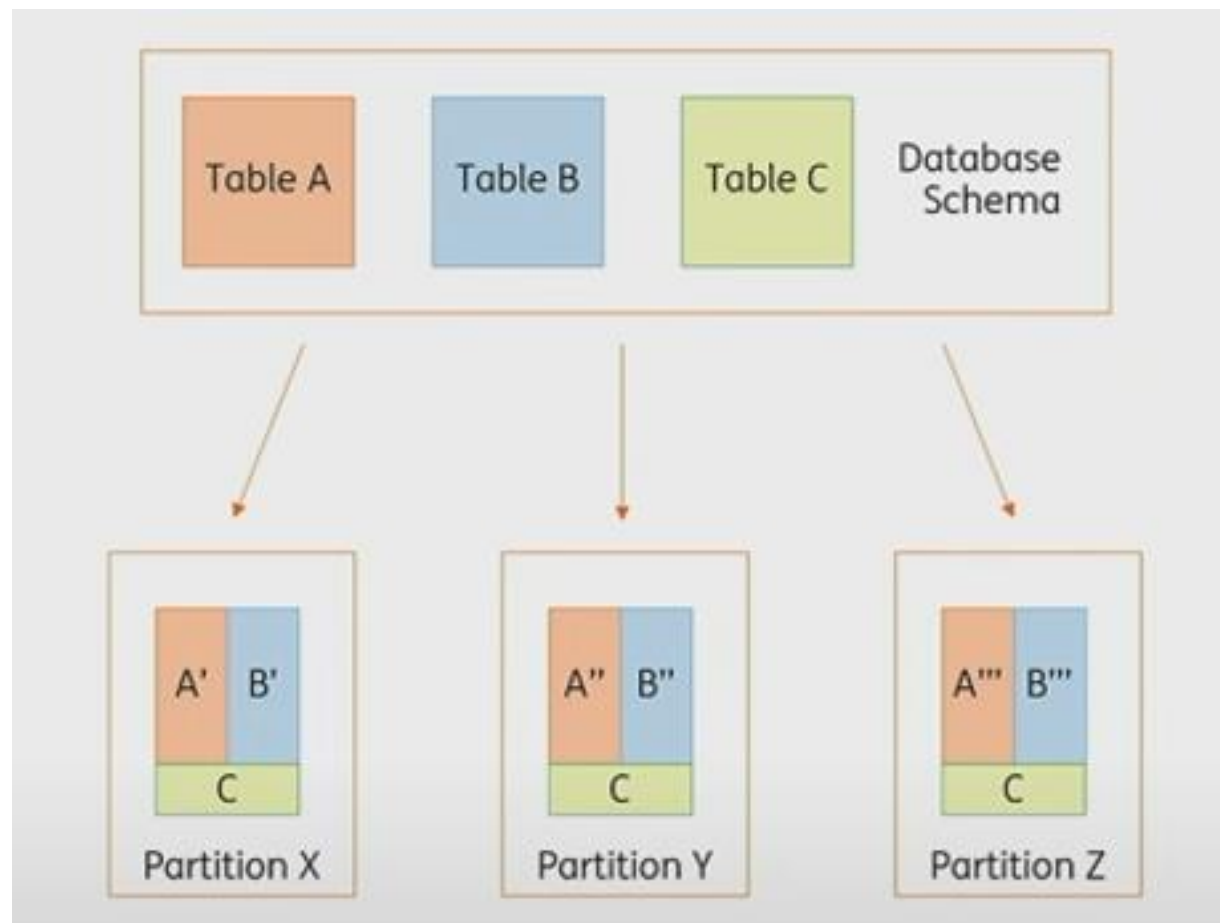


# Comparação SQL x NoSQL





# NewSQL





## NewSQL

**VOLTDDB**



**Cockroach DB**

Google  
Cloud  
Spanner







## NewSQL

**VOLTD**DB



**Cockroach** DB

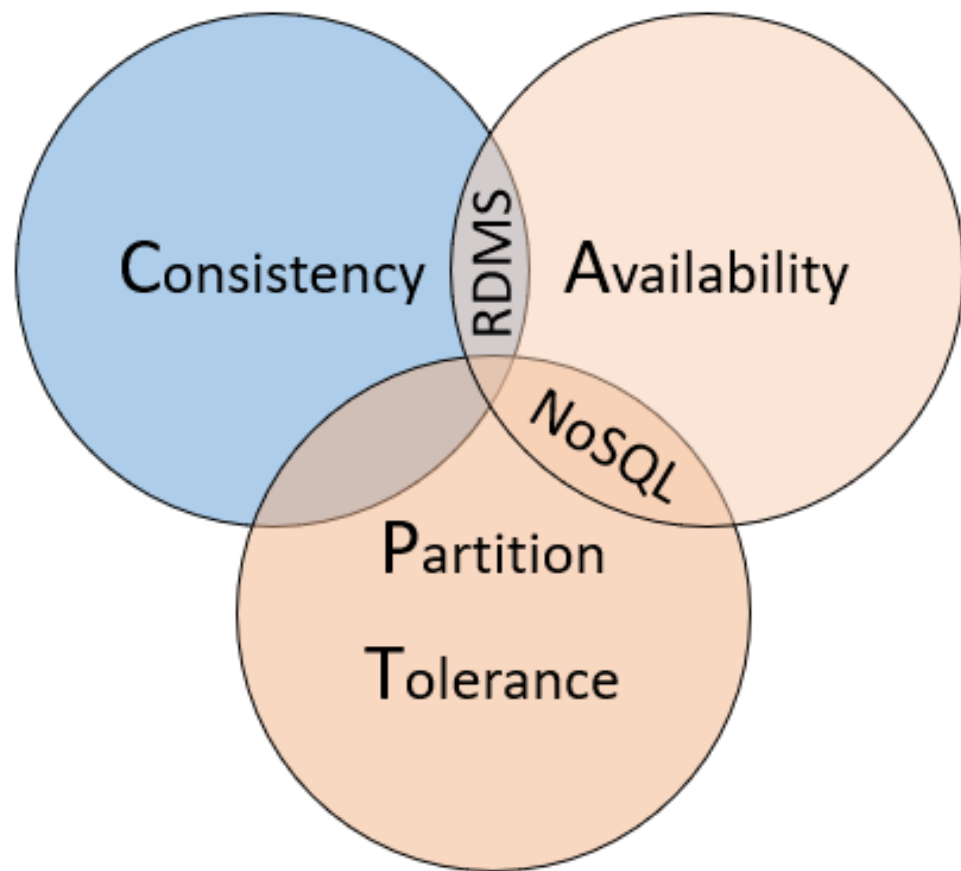
Google  
Cloud  
Spanner



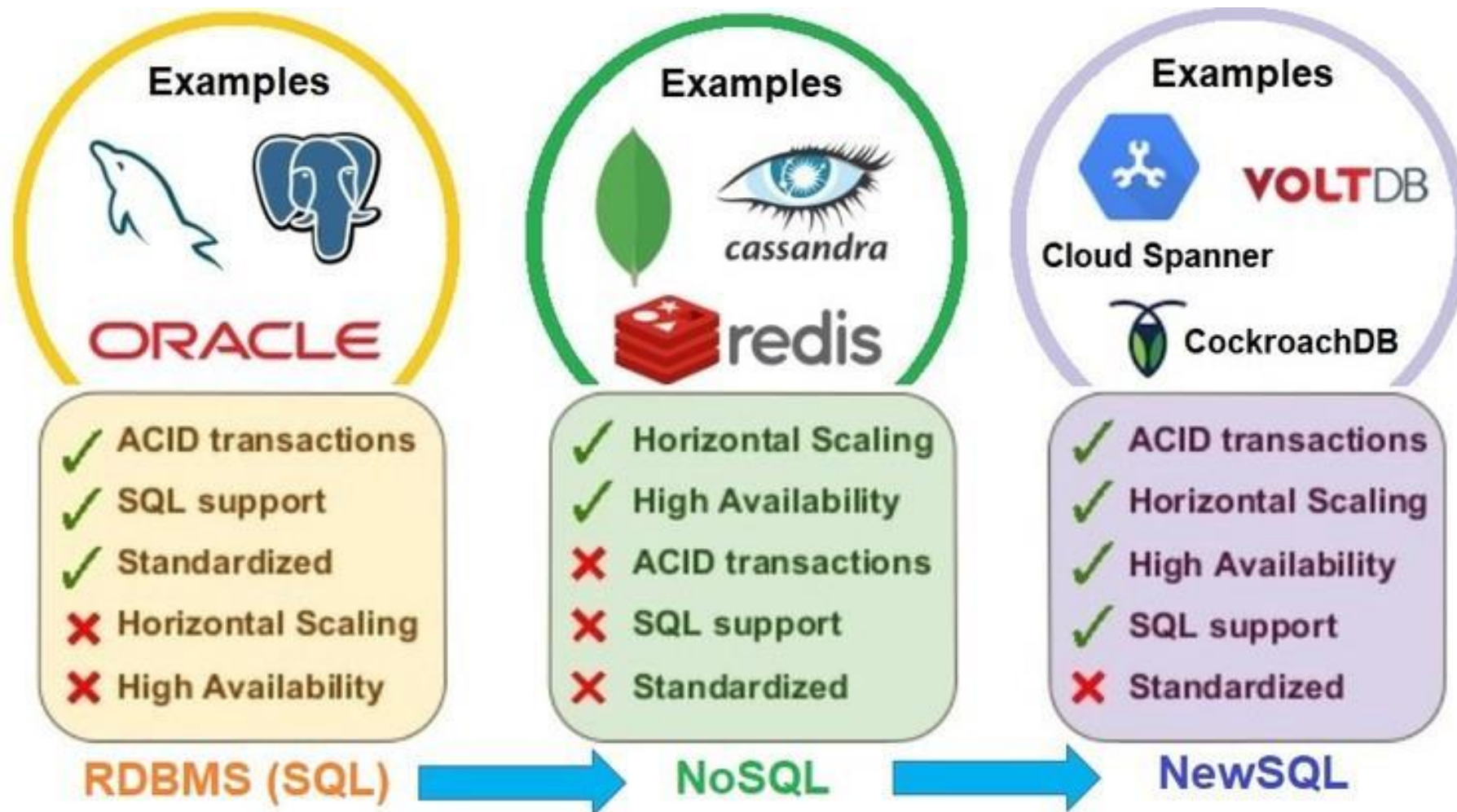
# Teorema CAP

O teorema afirma que os sistemas de dados distribuídos oferecerão uma compensação entre consistência, disponibilidade e tolerância à partição. E que qualquer banco de dados só pode garantir *duas* das três propriedades:

- *Consistência.* Cada nó no cluster responde com os dados mais recentes, mesmo se o sistema precisar bloquear a solicitação até que todas as réplicas sejam atualizadas. Se você consultar um "sistema consistente" para um item atualmente atualizado, aguardará essa resposta até que todas as réplicas sejam atualizadas com êxito. No entanto, você receberá os dados mais atuais.
- *Disponibilidade.* Cada nó retorna uma resposta imediata, mesmo que essa resposta não seja os dados mais recentes. Se você consultar um "sistema disponível" para um item que está atualizando, obterá a melhor resposta possível que o serviço pode fornecer nesse momento.
- *Tolerância a Partição.* Garante que o sistema continue operando mesmo que um nó de dados replicado falhe ou perca a conectividade com outros nós de dados replicados.



## Diferenças







#GCPSketchnotes  
07.10.2021

@PVERGADIA THECLOUDGIRL.DEV

# Which Database should I use?



## RELATIONAL



### Cloud SQL

Managed MySQL,  
PostgreSQL,  
SQL Server



### Cloud Spanner

Cloud-native with  
large scale,  
consistency,  
99.999% availability



### Bare Metal

Lift and shift  
Oracle workloads  
to Google Cloud

#### Good For:

General purpose  
SQL DB

RDBMS+ scale,  
HA, HTAP

RDBMS+ scale,  
HA, HTAP

#### Use Case:



Web  
frameworks



ERP



CRM



Ecommerce  
and web



SaaS  
application



Gaming



Global financial  
ledger



Supply chain/  
inventory  
management



Legacy  
applications



Data center  
retirement

## NON-RELATIONAL (NO SQL)

### DOCUMENT



### Firestore

Cloud Native, serverless,  
NoSQL document database,  
backend-as-a-service,  
global strong consistency,  
99.999% SLA

### KEY VALUE



### Cloud Bigtable

Cloud-native NoSQL  
wide-column store  
for large scale,  
low-latency workloads

#### Good For:

Large scale, complex  
hierarchical data

Heavy read + write, events

#### Use Case:



Mobile/web/  
IoT applications



Real-time sync



Offline sync



Personalized apps



Personalization



Adtech



Recommendation  
engines



Fraud detection

## IN MEMORY



### Memory Store

Fully managed Redis and  
Memcached for sub-millisecond  
data access

#### Good For:

In-memory and Key-value store

#### Use Case:



Caching



Gaming



Leaderboard



Social chat or  
news feed



Session store



Personalization



Adtech





## Vale a pena conferir:

<https://learn.microsoft.com/pt-br/dotnet/architecture/cloud-native/relational-vs-nosql-data>

<https://cloud.google.com/blog/topics/developers-practitioners/your-google-cloud-database-options-explained>

<https://aws.amazon.com/pt/startups/start-building/how-to-choose-a-database/>

<https://www.youtube.com/watch?v=czbExIRmyKo>