

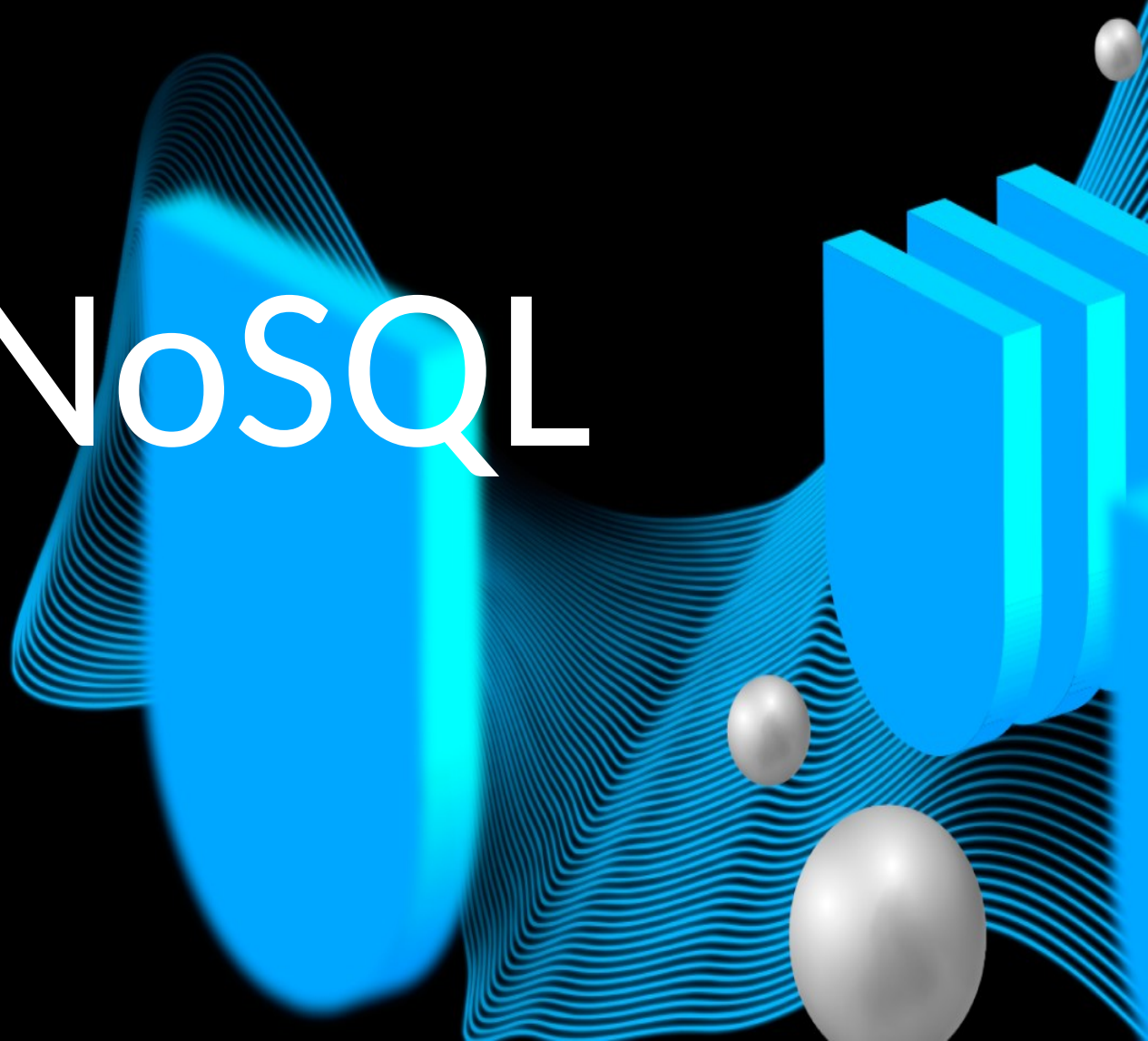


**UNITAU**  
Universidade de Taubaté





# NoSQL



# NoSQL



O termo **NoSQL** foi primeiramente utilizado em 1998 como o nome de um banco de dados relacional de código aberto que não possuía uma interface SQL. Seu autor, Carlo Strozzi, alega que o movimento NoSQL "é completamente distinto do modelo relacional e portanto deveria ser mais apropriadamente chamado "NoREL" ou algo que produzisse o mesmo efeito".

Atualmente, NoSQL ( Not Only SQL ) é o termo genérico para bancos de dados não relacionais que tem como um dos principais focos a escalabilidade.

Com o grande crescimento de aplicações web com um enorme volume de dados e tecnologias de cloud computing, a escalabilidade horizontal tem se tornado uma característica determinante em diversas situações.

Os bancos NoSQL abandonam o modelo relacional, o uso de esquemas pré-definidos e as garantias ACID para obter maior escalabilidade.



# Modelos de Banco de Dados NoSQL



O termo NoSQL define mais uma quebra de modelo do que a criação de um novo modelo. Existem diversos modelos de bancos NoSQL.

Uma das principais características dos modelos de bancos NoSQL é a utilização de esquemas flexíveis, ao contrário do modelo relacional onde o esquema das tabelas é rígido e todos registros de uma mesma tabela possuem os mesmos atributos.

A quebra da rigidez do esquema do banco de dados facilita a escalabilidade horizontal por não forçar que todos os registros tenham a mesma estrutura.

# Modelos de Banco de Dados NoSQL

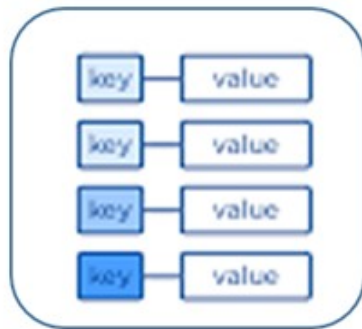


Existem vários modelos de bancos NoSQL e a classificação desses modelos não é um consenso. Alguns exemplos de modelos são:

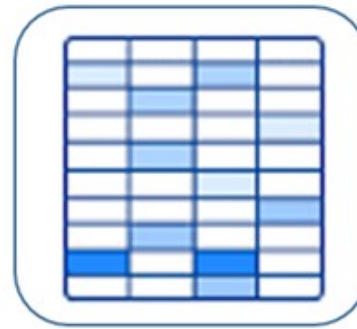
- Modelos Chave-Valor
- Modelos Orientados a Documentos
- Modelos Orientados a Coluna
- Modelos Orientados a Grafos



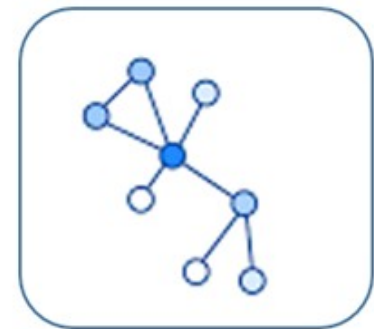
Document  
Store



Key-Value  
Store



Wide-Column  
Store



Graph  
Store

# Modelos Chave-Valor



**Os bancos baseados em chave-valor armazenam um conjunto de pares chave-valor indexados pela chave que permite o acesso ao valor baseado na chave. Esse é um dos modelos mais simples de bancos NoSQL. Exemplos desse modelo são:**

- Dynamo da Amazon**
- Voldemort**
- Riak**



# Modelos Orientados a Documentos



Os bancos orientados a documentos armazenam os dados como documentos, cada documento é associado a uma chave e não possuem uma estrutura definida, sendo possível acrescentar campos distintos a qualquer documento. A pesquisa por um documento pode ser feita pela sua chave ou pelos valores dos seus campos. Alguns exemplos de bancos orientados a documentos:

- CouchDB
- SimpleDB
- MongoDB

# Modelos Orientados a Colunas



( Wide Column Store / Column Families )

Nesse modelo os dados são armazenados por colunas e não por registros. Isso permite uma maior performance ao ler os dados de uma única coluna e pode facilitar a compactação dos dados por agrupar dados de um mesmo tipo. As colunas podem ser agrupadas em famílias de colunas. Alguns exemplos de bancos orientados a coluna:

- Cassandra
- BigTable
- HBase



# Modelos Orientados a Grafos



**Nesse modelo os dados são armazenados em nós de um grafo onde as arestas do grafo representam a associação entre os nós. Alguns exemplos de bancos orientados a grafos:**

- Neo4j
- Infinite Graph
- AllegroGraph

# UnQL



Um dos fatores do grande sucesso do modelo relacional é a linguagem SQL. A utilização dos princípios da álgebra e do cálculo relacional para a criação de uma linguagem simples e padronizada para manipulação dos dados foi essencial para a consolidação dos bancos de dados relacionais.

Apesar de uma das características dos bancos NoSQL ser a não utilização das junções do SQL, que é a base do relacionamento entre tabelas no modelo relacional, uma das metas de alguns desenvolvedores de soluções NoSQL é a criação de uma linguagem de consulta padronizada.

Os desenvolvedores do SQLite e CouchDB desenvolveram a linguagem **UnQL** ( Unstructured Query Language ), que é um superset da linguagem SQL para consulta em coleções.



# Aplicações



**O uso de bancos NoSQL ou de tradicionais bancos relacionais depende das características da aplicação.**

**A necessidade de escalabilidade e a importância da consistência são fatores fundamentais para se decidir por modelos que tem maior enfoque em uma característica ou outra.**

**Além dessas considerações, a forma como os dados serão manipulados também é um fator essencial, pois a forma e frequência com que os dados serão inseridos, atualizados, deletados, consultados e a forma como serão relacionados irão determinar o desempenho de cada modelo de armazenamento de dados na aplicação.**

**Essas características devem ser avaliadas para se determinar se um banco relacional ou se um banco NoSQL e qual modelo de banco NoSQL é o mais adequado para a aplicação.**