

# Desenvolvimento Web II

## Aula 12 - *Multi Tenancy*

Prof. Fabricio Bizotto

Instituto Federal Catarinense

*[fabricio.bizotto@ifc.edu.br](mailto:fabricio.bizotto@ifc.edu.br)*

Ciência da Computação  
6 de fevereiro de 2024

- 1 Definição
- 2 Analogia com Imóveis
- 3 *Single Tenant vs. Multi Tenant*
- 4 Tipos de *Multi Tenancy*
  - Schema Database
  - One Schema Per Tenant
  - One Database Per Tenant
- 5 *Multi Tenancy* em uma API REST
- 6 Considerações Finais
- 7 QUIZ
- 8 Experimentos

# Multi Tenancy

## Definição

- *Multi Tenancy* é um padrão de arquitetura de software onde uma única instância de software serve a múltiplos clientes, chamados de *tenants*.
- Cada *tenant* é isolado dos demais, e pode ter suas próprias configurações, dados e personalizações.
- O *Multi Tenancy* é comum em sistemas de *Software as a Service* (SaaS), onde um único sistema é utilizado por múltiplos clientes.
- O *Multi Tenancy* pode ser implementado em diferentes níveis de um sistema, como banco de dados, aplicação, interface de usuário, etc.

## Empresas que Utilizam *Multi Tenancy*

- **Salesforce:** um único sistema de CRM utilizado por milhares de empresas.
- **Google Apps:** um único sistema de e-mail, calendário, etc., utilizado por milhões de usuários.
- **Shopify:** um único sistema de e-commerce utilizado por milhares de lojas.
- **Microsoft 365:** um único sistema de produtividade utilizado por milhões de usuários.
- **Dropbox:** um único sistema de armazenamento de arquivos utilizado por milhões de usuários.

# Multi Tenancy

Analogia com Imóveis

- Imagine um prédio de apartamentos.
- Cada apartamento é um *tenant*.
- Cada apartamento tem suas próprias chaves, móveis, decoração, etc.
- Cada apartamento é isolado dos demais.
- O prédio é o sistema de *Multi Tenancy*.

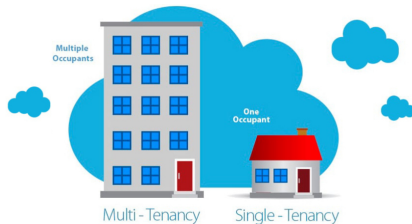


Figura: Single Tenant vs. Multi Tenant

Não é a mesma coisa que *Multi User*?

Não. Todos sistema *Multi Tenant* é *Multi User*, mas nem todo sistema *Multi User* é *Multi Tenant*.

# Multi Tenancy

## Aplicação Multi User

- 1 Existe compartilhamento de dados e recursos entre os usuários.
- 2 O professor pode ver e interagir com os dados dos alunos.
- 3 O aluno pode ver suas próprias notas.
- 4 O coordenador pode gerenciar as disciplinas, alunos, professores.
- 5 Não existe isolamento entre os usuários.
- 6 Não posso vender esse produto para outra escola.

Figura: Aplicação Multi User

## Definição

Aplicação que serve a um único cliente. Cada cliente tem sua própria instância da aplicação. Cada instância é isolada das demais.

## Vantagens

- A base de dados é isolada. Maior segurança.
- A performance é previsível. Não será afetada por outros clientes.
- Facilidade para customizar a aplicação para um cliente específico.
- Oferece maior controle sobre a aplicação.

## Desvantagens

- Cliente precisa arcar com os custos de uma instância completa.
- Exige maior esforço para manter e atualizar as instâncias.

### Definição

Aplicação que serve a múltiplos clientes. Todos os clientes compartilham a mesma instância da aplicação. Cada cliente é isolado dos demais.

### Vantagens

- Redução de custos. A aplicação é compartilhada entre os clientes.
- Facilidade para manter e atualizar a aplicação.
- Facilidade para escalar a aplicação.
- Facilidade para adicionar novos clientes.

### Desvantagens

- Menor controle sobre a aplicação.
- Menor segurança. Um cliente pode afetar a performance de outros.
- Dificuldade para customizar a aplicação para um cliente específico.

Esses são os tipos mais comuns de *Multi Tenancy*:

- **Shared Database:** todos os *tenants* compartilham a mesma base de dados, tabelas, etc. Cada *tenant* tem suas próprias linhas de dados. Essa é a forma mais básica de *Multi Tenancy*.
- **One Schema Per Tenant:** todos os *tenants* compartilham a mesma base de dados, mas cada *tenant* tem seu próprio *schema*. Cada *tenant* tem suas próprias tabelas, índices, etc.
- **One Database Per Tenant:** cada *tenant* tem sua própria base de dados. Cada *tenant* tem suas próprias tabelas, índices, etc. É o mais seguro, mas também o mais caro.



# Multi Tenancy

## Schema Database

- Somente um banco de dados é utilizado.
- Gerenciamento simples.
- É a forma mais comum de *Multi Tenancy*.
- Aumenta os riscos de segurança ao expor dados de um *tenant* para outro.
- Não tem isolamento físico entre os *tenants*. Esse isolamento está na mão do desenvolvedor.

**Select count (\*) from Sheep where tenant\_id=1**

Types of a multi-tenant database

### 1. Shared database

ID	Sheep	Date of Birth	Tenant ID
1	Shaun	2015	1
2	Dolly	2018	1
3	Woolly	2019	3
4	Puffy	2016	3
5	Fluffy	2015	3
6	Cheese	2018	6
7	Swirling	2019	6

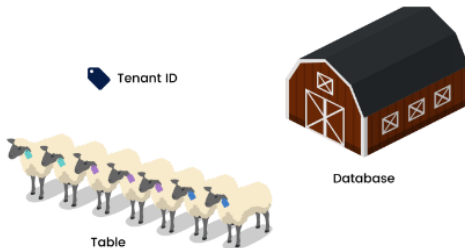


Figura: Schema Database

# Multi Tenancy

## One Schema Per Tenant

- Cada *tenant* tem seu próprio *schema*.
- Os dados de cada *tenant* estão isolados, mas compartilham a mesma base de dados.
- Reduz os riscos de segurança pois não precisa usar a cláusula `WHERE` para filtrar os dados entre os *tenants*.
- **Risco de consultar o *schema* errado.**

**Select count (\*) from schema\_tenant\_3.sheep**

Types of a multi-tenant database

2. One schema per tenant

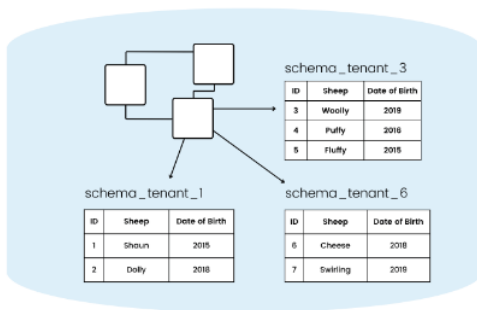
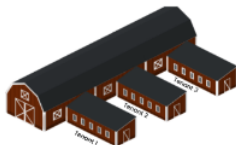


Figura: One Schema Per Tenant

# Multi Tenancy

## One Database Per Tenant

- Cada *tenant* tem seu próprio banco de dados.
- Normalmente é usado para isolar um cliente grande.
- Maior segurança e isolamento entre os *tenants*.
- Sem complexidade para filtrar os dados.
- **Maior custo de manutenção e infraestrutura.**
- **Alterações no banco de dados precisam ser replicadas para todos os *tenants*.**

`db_tenant_3_SELECT_COUNT(*) de_sheep`

Types of a multi-tenant database

### 3. One database per tenant



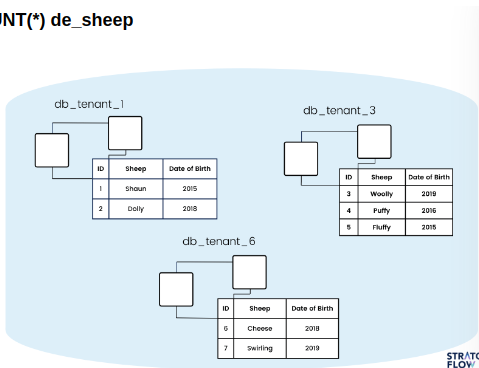
Tenant 1



Tenant 2



Tenant 3



STRATO  
FLOW

Figura: One Database Per Tenant

# Multi Tenancy

Multi Tenancy em uma API REST

- Identificação do *tenant* baseada em parâmetros de consulta  
GET /api/v1/tasks?tenant=tenant\_id
- Identificação do *tenant* baseada em subdomínio  
GET https://tenant1.myapp.com/api/v1/tasks
- Identificação do *tenant* baseada na URL  
GET https://myapp.com/tenant1/api/v1/tasks

- O *Multi Tenancy* é uma forma de compartilhar recursos entre múltiplos clientes.
- Existem diferentes formas de implementar o *Multi Tenancy*, cada uma com suas vantagens e desvantagens.
- O *Multi Tenancy* é comum em sistemas de *Software as a Service* (SaaS).
- O *Multi Tenancy* é uma forma de reduzir custos e facilitar a manutenção e atualização de sistemas.
- O *Multi Tenancy* é uma forma de escalar sistemas para atender a um grande número de clientes.

Vamos praticar um pouco o que vimos até agora?

### QUIZ - Multi Tenancy

# Experimentos

Vamos praticar?

Nesta atividade, o aluno deve desenvolver uma aplicação web simples utilizando a arquitetura multi tenancy. O objetivo é criar uma plataforma de lista de tarefas compartilhada, onde diferentes usuários, ou inquilinos, podem criar e gerenciar suas próprias listas de tarefas dentro da mesma instância da aplicação, garantindo o isolamento e a segurança dos dados de cada inquilino. Os participantes devem planejar e projetar a aplicação, implementar a lógica da aplicação com recursos como autenticação de usuário, gerenciamento de tarefas e compartilhamento de listas.

A entrega no incluirá o código-fonte da aplicação (GitHub) e documentação de suporte (README.md).