



Estácio

Missão Prática Nível 4 – Mundo 4

Fernanda Canto P. da Costa - 202208379788

Campus de Ipanema

Tirando proveito da nuvem para projetos de software – Turma 22.3 –
4º semestre

Github: <https://github.com/nandacpc/Missao4-Mundo4>

Documentação do Projeto: Banco de dados no Azure SQL para
LogiMove Transportes

Introdução

Este documento descreve a implementação de um banco de dados SQL no Azure para a empresa LogiMove Transportes que armazenará informações sobre motoristas, clientes e pedidos.

Objetivos do Projeto

- Desenvolver um banco de dados robusto e seguro no Azure SQL.
- Garantir que o banco de dados possa escalar conforme a empresa cresce.
- Facilitar a integração com outras plataformas e serviços.

Estrutura do Projeto

1. Configuração do Ambiente Azure
2. Design do Banco de Dados
3. Implementação do Banco de Dados
4. Inserção e Gestão de Dados
5. Execução e Validação de Consultas
6. Operações CRUD Eficientes

1. Configuração do Ambiente Azure

Script: `ambiente_azure.sh`

Este script configura o ambiente necessário no Azure, criando o grupo de recursos, o servidor SQL e o banco de dados. Também configura as regras de firewall para permitir acesso ao servidor.

Caminho: `/azure_setup/ ambiente_azure.sh`

```
# Configuração de variáveis
RESOURCE_GROUP="LogiMoveResourceGroup"
SQL_SERVER_NAME="LogiMoveSQLServer"
SQL_DATABASE_NAME="LogiMoveDatabase"
ADMIN_USERNAME="adminlogimove"
ADMIN_PASSWORD="SenhaSegura123!"
LOCATION="brazilsouth"

# Criação do grupo de recursos
az group create --name $RESOURCE_GROUP --location $LOCATION

# Criação do servidor de banco de dados
az sql server create \
  --name $SQL_SERVER_NAME \
  --resource-group $RESOURCE_GROUP \
  --location $LOCATION \
  --admin-user $ADMIN_USERNAME \
  --admin-password $ADMIN_PASSWORD

# Criação do banco de dados SQL
az sql db create \
  --resource-group $RESOURCE_GROUP \
  --server $SQL_SERVER_NAME \
  --name $SQL_DATABASE_NAME \
```

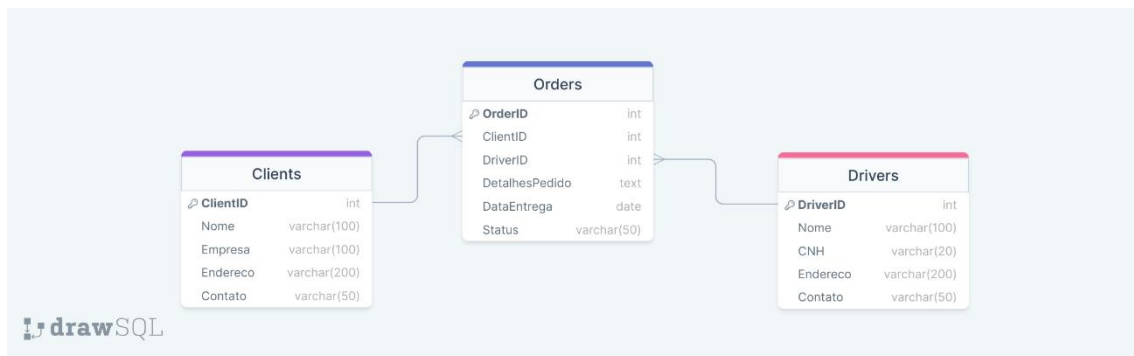
```
--service-objective S0

# Configuração de firewall para permitir acesso ao servidor
az sql server firewall-rule create \
  --resource-group $RESOURCE_GROUP \
  --server $SQL_SERVER_NAME \
  --name AllowMyIP \
  --start-ip-address 0.0.0.0 \
  --end-ip-address 255.255.255.255
```

2. Design do Banco de Dados

- Diagrama ER

O diagrama ER (Entidade-Relacionamento) visualiza as relações entre as tabelas: `Drivers`, `Clients`, e `Orders`.



3. Implementação do Banco de Dados

Script: `banco_de_dados.sql`

Este script cria as tabelas no banco de dados conforme o design definido.

Caminho: `/sql_scripts/banco_de_dados.sql`

```
-- Criação da tabela Drivers
CREATE TABLE Drivers (
  DriverID INT PRIMARY KEY,
  Nome VARCHAR(100),
  CNH VARCHAR(20),
  Endereco VARCHAR(200),
  Contato VARCHAR(50)
);
GO
```

```

-- Criação da tabela Clients
CREATE TABLE Clients (
    ClientID INT PRIMARY KEY,
    Nome VARCHAR(100),
    Empresa VARCHAR(100),
    Endereço VARCHAR(200),
    Contato VARCHAR(50)
);
GO

-- Criação da tabela Orders
CREATE TABLE Orders (
    OrderID INT PRIMARY KEY,
    ClientID INT,
    DriverID INT,
    DetalhesPedido TEXT,
    DataEntrega DATE,
    Status VARCHAR(50),
    FOREIGN KEY (ClientID) REFERENCES Clients(ClientID),
    FOREIGN KEY (DriverID) REFERENCES Drivers(DriverID)
);
GO

```

4. Inserção e Gestão de Dados

Script: `inserir_dados.sql`

Este script insere dados de teste nas tabelas criadas para simular diferentes cenários e casos de uso.

Caminho: `/sql_scripts/inserir_dados.sql`

```

-- Inserção de dados de teste na tabela Drivers
INSERT INTO Drivers (DriverID, Nome, CNH, Endereço, Contato)
VALUES (1, 'João Silva', '123456789', 'Rua A, 123', '91111-1111');
GO

-- Inserção de dados de teste na tabela Clients
INSERT INTO Clients (ClientID, Nome, Empresa, Endereço, Contato)
VALUES (1, 'Maria Santos', 'Empresa X', 'Rua B, 456', '92222-2222');
GO

-- Inserção de dados de teste na tabela Orders
INSERT INTO Orders (OrderID, ClientID, DriverID, DetalhesPedido,
DataEntrega, Status)
VALUES (1, 1, 1, 'Entrega de documentos', '2024-06-01', 'Pendente');
GO

```

5. Execução e Validação de Consultas

Script: `validacao.sql`

Este script executa consultas para validar a inserção de dados e garantir que o banco de dados está funcionando conforme esperado.

Caminho: `/sql_scripts/validacao.sql`

```
-- Consulta de todos os dados na tabela Drivers
SELECT * FROM Drivers;
GO

-- Consulta de todos os dados na tabela Clients
SELECT * FROM Clients;
GO

-- Consulta de todos os dados na tabela Orders
SELECT * FROM Orders;
GO
```

6. Operações CRUD Eficientes

Script: `operacoes_crud.sql`

Este script demonstra operações CRUD - Criar, Ler, Atualizar e Deletar dados.

Caminho: `/sql_scripts/operacoes_crud.sql`

```
-- Atualização de dados na tabela Orders
UPDATE Orders SET Status='Concluído' WHERE OrderID=1;
GO

-- Verificação da atualização
SELECT * FROM Orders WHERE OrderID=1;
GO

-- Exclusão de dados na tabela Orders
DELETE FROM Orders WHERE OrderID=1;
GO

-- Verificar se a tabela Orders está vazia
SELECT COUNT(*) FROM Orders;
GO
```

Conclusão

Este projeto configura um banco de dados no Azure SQL para a LogiMove Transportes, permitindo a migração de um sistema baseado em papel para uma solução digital. Com isso, a empresa poderá melhorar a coordenação e o rastreamento de remessas, aumentando a eficiência e a satisfação do cliente. A implementação foi realizada com sucesso, incluindo a configuração do ambiente Azure, design e criação do banco de dados, inserção e gestão de dados, validação de consultas e demonstração de operações CRUD.