

Nome: Albano Roberto Drescher Von Maywitz

Semana 2

Tarefa 3

Dado o estudo de caso de uma empresa de entregas X com sede em Curitiba e uma filial em SP, escolha um SGBD de sua preferência e descreva sobre o suporte deste SGBD para implementar um BDD para esta empresa. O BDD deve ser formado pelas tabelas Empregado e Entrega, que contém informações dos empregados e entregas, respectivamente, de forma que as tabelas sejam fragmentadas e replicadas para aumentar a disponibilidade, confiabilidade e desempenho. A tabela Empregado é replicada nas duas cidades. A tabela Entrega é fragmentada de acordo com o endereço e é replicada de forma assíncrona na outra cidade. Descrever os comandos e configurações necessárias para implementar este estudo de caso.

Resposta:

O PostgreSQL foi escolhido como SGBD devido à sua robustez, escalabilidade, suporte a replicação e fragmentação de dados, bem como ser opensource.

Para começar, é necessário criar as tabelas Empregado (Exemplo):

```
CREATE TABLE empregado (  
  id SERIAL PRIMARY KEY,  
  nome TEXT NOT NULL,  
  sobrenome TEXT NOT NULL,  
  endereco TEXT NOT NULL,  
  cidade TEXT NOT NULL,  
  estado TEXT NOT NULL,  
  salario NUMERIC(10, 2) NOT NULL,  
  data_de_contratacao DATE NOT NULL  
);
```

E Entrega (Exemplo):

```
CREATE TABLE entrega (  
  id SERIAL PRIMARY KEY,  
  descricao TEXT NOT NULL,  
  endereco_de_entrega TEXT NOT NULL,  
  cidade_de_entrega TEXT NOT NULL,  
  estado_de_entrega TEXT NOT NULL,  
  data_de_entrega DATE NOT NULL  
);
```

Para replicar a tabela empregado entre as cidades de Curitiba e São Paulo, podemos utilizar a replicação nativa do PostgreSQL. Primeiro, precisamos definir a configuração de replicação na cidade de Curitiba. Para isso, adicionamos as seguintes linhas ao arquivo postgresql.conf:

```
listen_addresses = '*'  
wal_level = replica  
max_wal_senders = 3
```

Isso permite que o servidor receba conexões de qualquer endereço e habilita o envio de registros de log de gravação (WAL) para a replicação. Também limita o número máximo de envios de WAL para 3.

Em seguida, precisamos definir a configuração de replicação na cidade de São Paulo. Para isso, adicionamos as seguintes linhas ao arquivo postgresql.conf:

```
listen_addresses = '*'  
hot_standby = on
```

Isso permite que o servidor receba conexões de qualquer endereço e habilita a execução em modo somente leitura.

Para iniciar a replicação, precisamos criar um arquivo de configuração de replicação (pg_hba.conf) em ambas as cidades, que define as permissões de acesso aos servidores. Por exemplo, em Curitiba, podemos adicionar a seguinte linha ao final do arquivo pg_hba.conf:

```
host replication replica_ip trust
```

Isso permite que o servidor replica_ip se conecte ao servidor de Curitiba para replicação. Em São Paulo, adicionamos a seguinte linha:

```
host all all curitiba_ip trust
```

Isso permite que o servidor replica_ip se conecte ao servidor de Curitiba para replicação. Em São Paulo, adicionamos a seguinte linha:

```
host all all curitiba_ip trust
```

Isso permite que qualquer usuário do servidor de São Paulo se conecte ao servidor de Curitiba. Para iniciar a replicação, precisamos criar uma slot de replicação no servidor de Curitiba com o seguinte comando SQL:

```
SELECT pg_create_physical_replication_slot('sp_replication_slot');
```

Isso cria uma cópia da base de dados de Curitiba no diretório /path/to/data/dir e inicia a replicação em modo somente leitura. A opção *-S sp_replication_slot* especifica o nome do slot de replicação criado anteriormente.

A fragmentação da tabela "entrega" de acordo com o endereço de entrega pode ser feita utilizando particionamento por chave. Isso envolve criar uma tabela mãe que contém as colunas principais e uma ou mais tabelas filhas que armazenam os dados de acordo com as chaves de partição.

Para criar a tabela mãe, podemos utilizar o seguinte comando:

```
CREATE TABLE entrega_mae (
  id SERIAL PRIMARY KEY,
  descricao TEXT NOT NULL,
  endereco_de_entrega TEXT NOT NULL,
  cidade_de_entrega TEXT NOT NULL,
  estado_de_entrega TEXT NOT NULL,
  data_de_entrega DATE NOT NULL,
  endereco_de_entrega_id INTEGER NOT NULL
);
```

Em seguida, podemos criar as tabelas filhas utilizando o comando `CREATE TABLE ... PARTITION OF`. Por exemplo, para criar uma tabela filha para entregas em Curitiba, podemos utilizar o seguinte comando:

```
CREATE TABLE entrega_curitiba PARTITION OF entrega_mae
  FOR VALUES IN ('Curitiba');
```

Para criar uma tabela filha para entregas em São Paulo, podemos utilizar o seguinte comando:

```
CREATE TABLE entrega_sao_paulo PARTITION OF entrega_mae
  FOR VALUES IN ('São Paulo');
```

Dessa forma, todas as entregas com endereço de entrega em Curitiba serão armazenadas na tabela `entrega_curitiba`, enquanto todas as entregas com endereço de entrega em São Paulo serão armazenadas na tabela `entrega_sao_paulo`.

É importante ressaltar que a configuração de replicação e a fragmentação da tabela podem envolver uma série de etapas adicionais, como a definição de índices e restrições de integridade, a definição de políticas de backup e recuperação, entre outras.