Fundamentos de banco de dados.

O que é são informações?

Estrutura de dados, organização de dados conjunto de dados relacionados entre si que geram valor, que criar sentidos aos dados.

Material do conhecimento.

modelos relacionais

Conceito e melhores práticas com banco de dados PostgreSQL.

Definição

Modelo mais comum, que classifica e organiza as informações em tabelas com linhas e colunas.

As linhas, ou tuplas, são os dados organizados, são os valores das tabelas, e as colunas são os atributos destes dados.

informações relacionais

|  |  |
| --- | --- |
| 12131231312133 | Emengarda Rogerio |
| 85522224668979 | Daniel cleonice |
| telefone | Proprietários |

Transformando em formato de tabela

|  |  |  |  |
| --- | --- | --- | --- |
| 1 | 12131231312133 | 1 | Emengarda Rogerio |
| 2 | 85522224668979 | 2 | Daniel cleonice |
| Telefones | | Proprietários | |

O que é realmente uma tabela?

conjuntos de dados dispostos em colunas e linhas referentes a um objetivo comum.

As colunas são consideradas com "campos da tabela", como atributos da tabela.

As linhas de uma tabela são chamadas também de tuplas, e é onde estão os valores, os dados.

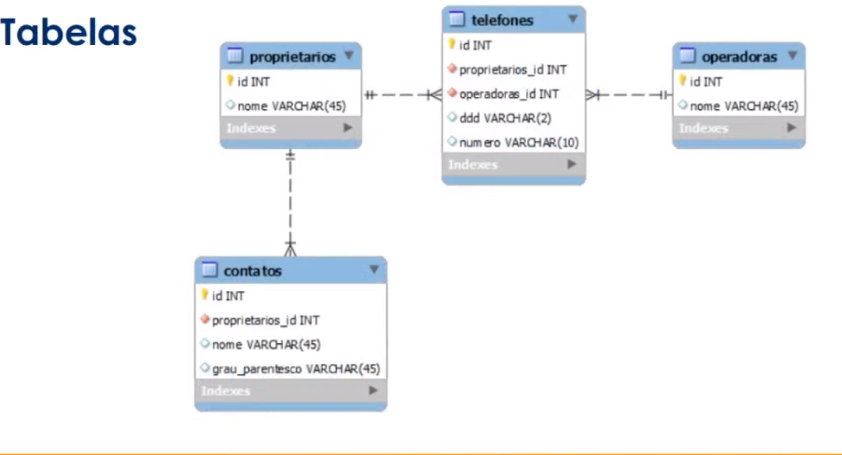
**o que pode ser definido como tabelas**?

* Coisas tangíveis
  + Elementos físicos (carros, produtos, animal)
* Funções
  + Perfis de usuário, status de compra (perfil, Gerente)
* Eventos ou ocorrências
  + Produtos de um pedido, histórico de dados

**A importância das chaves nas nossas tabelas.**

* Chave primaria / Primary Key / PK
  + Conjunto de um ou mais campos que nunca se repetem. Identidade data tabela. São utilizados com índice de referência na criação de relacionamentos entre tabelas.
* Chave estrangeira / Foreign key / FK
  + Valor de referência a uma PK de outra tabela da mesma tabela para criar um relacionamento.

**Modelo relacional**



**Sistema de gerenciamento de banco de dados**

Ou sistemas de gestão de banco de dados.

Chamamos pelo siglas **SGBD**.

Conjunto de programas ou software responsáveis pelo gerenciamento de um banco de dados.

Programas que facilitam a administração de um banco de dados.

O que é o PostgreSQL?

Sistema de gerenciamento de banco de dados objeto relacional.

Teve inicio do departamento de ciência da computação na universidade da Califórnia em Berkeley em 1986.

SGBD Opensource.

Instalação e documentação do PostgreSQL

Site oficial:

<https://www.postgreSQL.org/>

Download com instruções passo a passo:

<https://www.postgreSQL.org/download/>]

Documentação completa:

<https://www.postgreSQL.org/docs/manuals/>

o Arquivo postgreSQl.conf

Arquivo de configurações do PostgreSQL

* Arquivo postgresql.conf
* Arquivo pg\_hba.conf
* Arquivo pg\_ident.conf
* Comandos administrativos

definição dos arquivos.

* Arquivo onde estão definidas e armazenadas todas as configurações do servidor PostgreSQL.
* Alguns parâmetros só podem ser alterados com uma reinicialização do banco de dados.
* A view\_settings, acessada por dentro do banco de dados, guarda todas as configurações atuais.

Postgresql.conf

Ao acessar a view pg\_settings, e possível visualizar todoas as configurações atuais.

1. SELECT name, setting FROM pg\_setting;

OU e possível usar o comando:

SHOW[parâmetro];

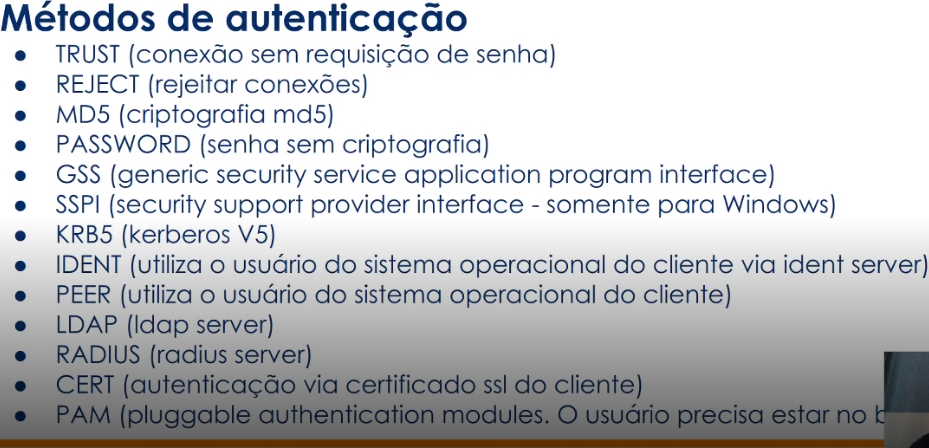
Configurações de conexão

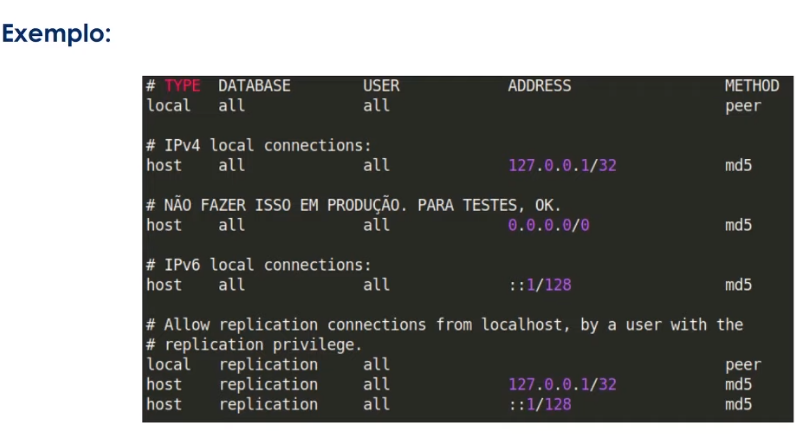
* LISTEN\_ADDRESSES
  + Endereço(s) TCP/IP das interfaces que o servidor Postgresql vai escutar / liberar conexões.
* PORT
  + A porta TPC que o servidor Postgresql vai ouvir. O padrão e 5432
* MAX\_CONNECTIONS
  + Numero máximo de conexões simultâneas no servidor postgresql
* SUPERUSER\_RESERVED\_CONNECTIONS
  + Numero de conexões(slots) reservadas para conexões ao banco de dados de super usuários

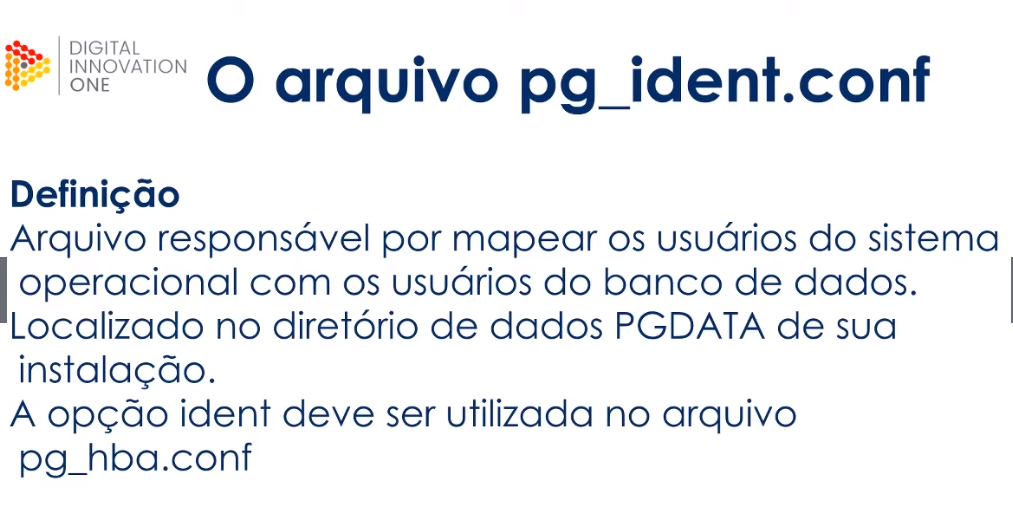
Configurações de autenticação

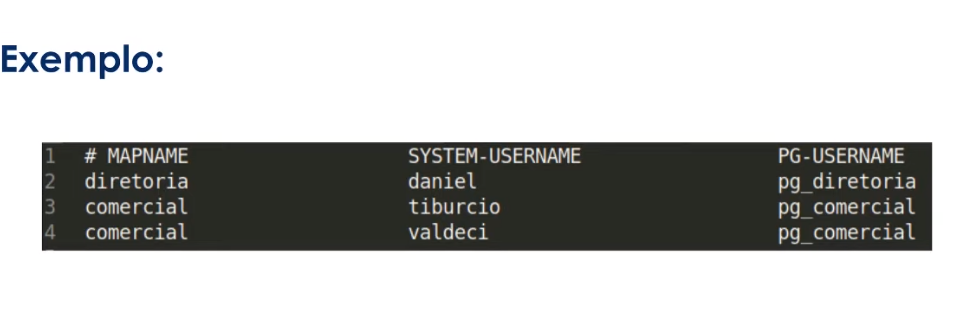
* AUTHENTICATION\_TIMEOUT
  + Tempo máximo em segundas para o cliente conseguir uma conexão com o servidor.
* WOR\_MEM
  + Tamanho da memoria para operações de agrupamento e ordenação(ODER BY, DISTINCT, MERGE JOINS);
* MAINTENHANCE\_WORK\_MEM
  + Tamanho da memoria para operações com VACUUM, INDEX, ALTER TABLE.

Configurações do arquivos pg\_hba.conf

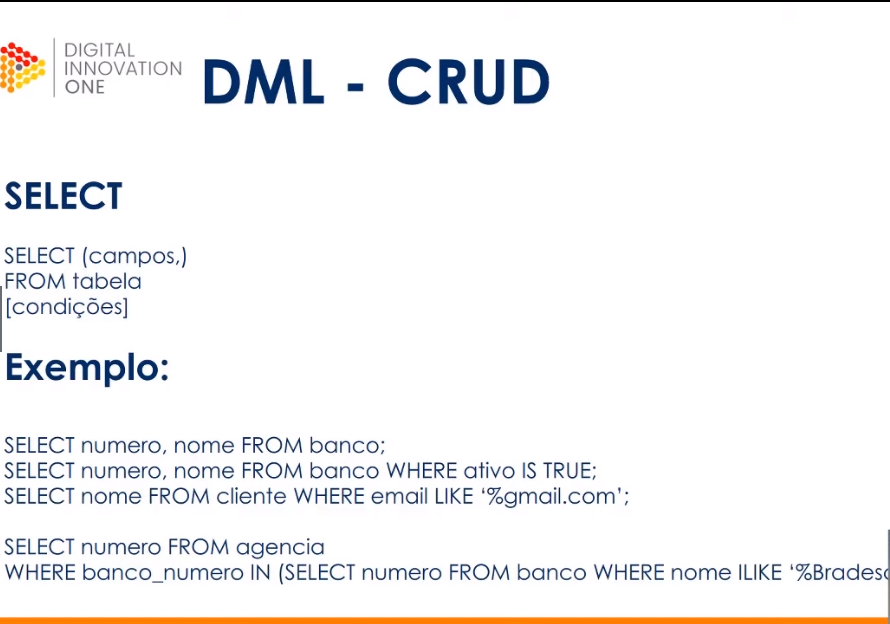








USANDO DML



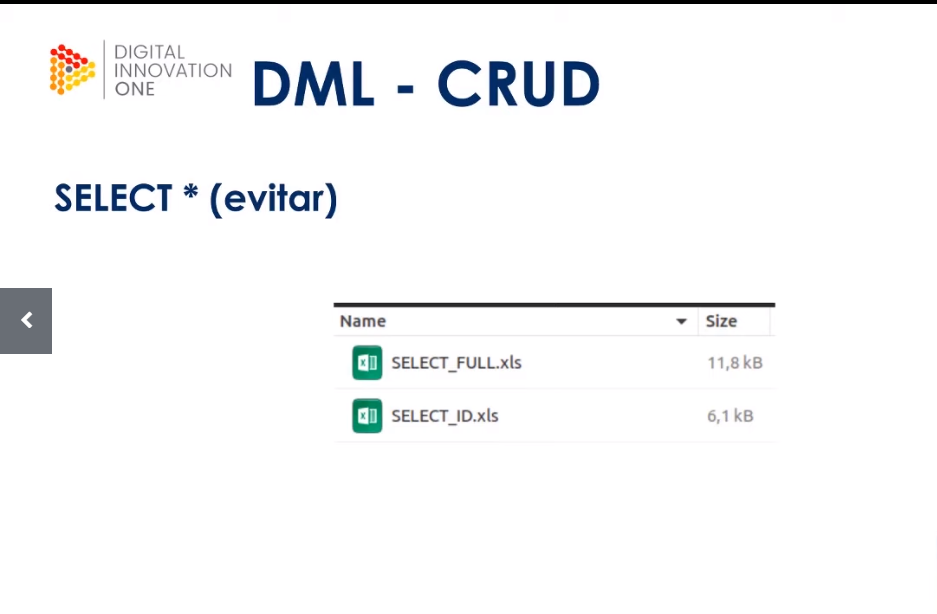
FALANDO UM POUCO DOS OPERADORES



FALANDO DA IDEMPOTÊNCIA



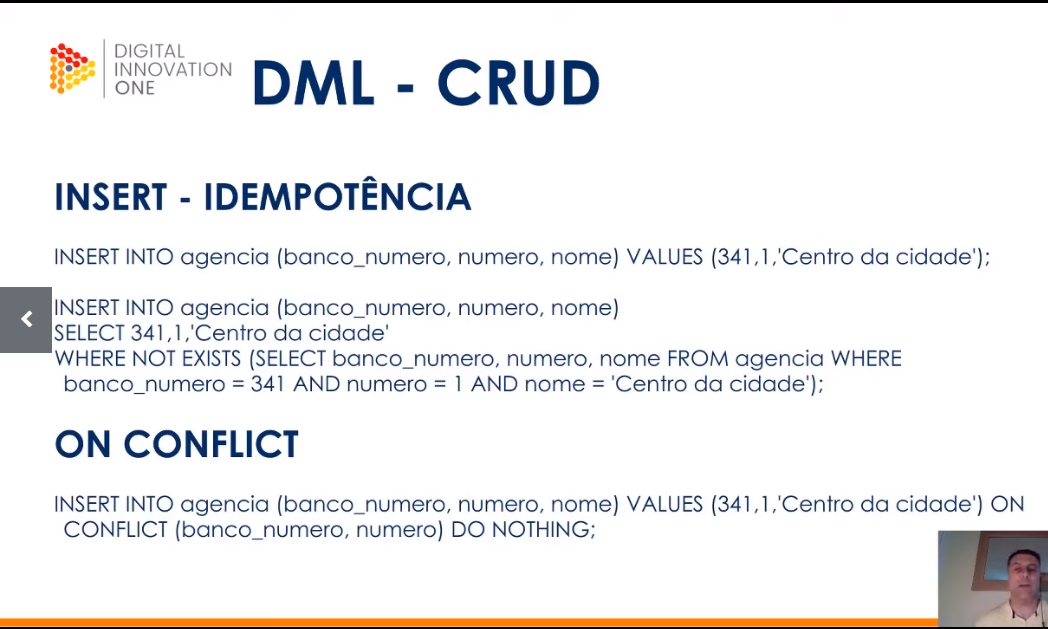
NÃO USAR \* (EVITE USAR, SEMPRE COLOQUE AS COLUNAS PARA REALIZAR O SEU SELECT )



INSERT



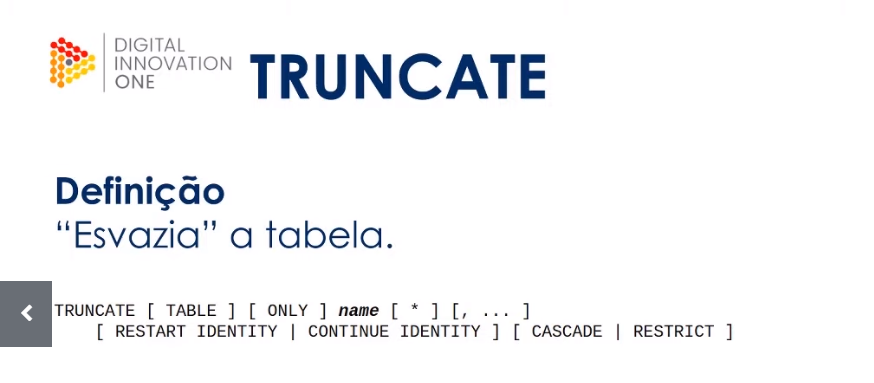
IDEMPOTÊNCIA DE CÓDIGO



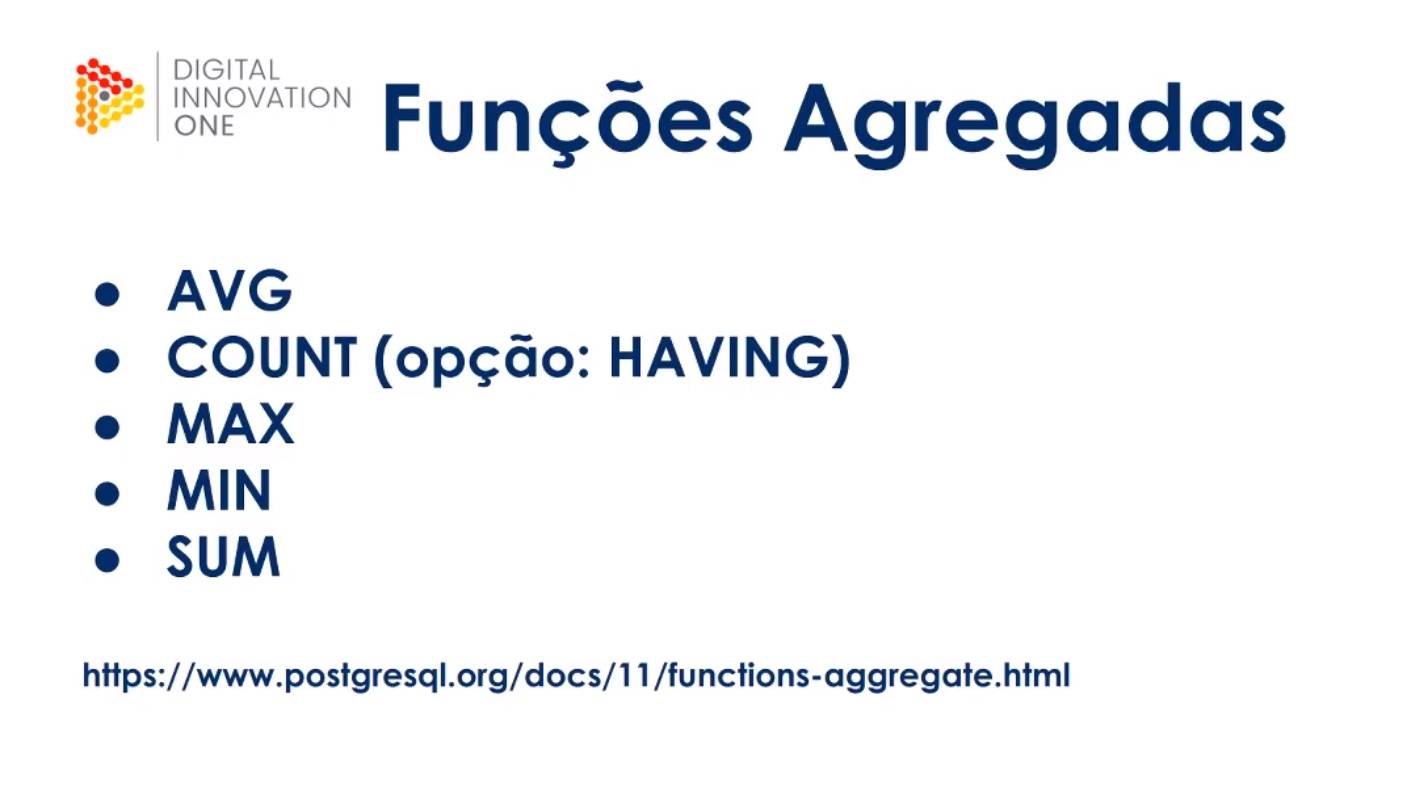
SEMPRE USAR WHERE NESTA CONDIÇÃO



USANDO O TRUNCATE – IRÁ ZERA SUA TABELA, TOME MUITO CUIDADO AO USAR.



VAMOS ESTUDAR FUNCÇOES AGREGADAS



SEGUE ALGUNS SELECT QUE FOI REALIZANDO NO CURSO USANDO PGADMIN

SELECT numero, nome FROM banco;

SELECT banco\_numero, numero, nome FROM agencia;

SELECT numero, nome, emal FROM cliente;

SELECT banco\_numero, agencia\_numero, cliente\_numero FROM cliente\_transacoes;

select \* from information\_schema.columns WHERE table\_name = 'banco';

select column\_name, data\_type FROM information\_schema.columns WHERE table\_name = 'banco';

-- AVG

-- COUNT (HAVING)

-- MAX

-- MIN

-- SUM

SELECT avg(valor) FROM cliente\_transacoes;

SELECT count(numero) FROM cliente;

SELECT count(numero), email FROM cliente WHERE email LIKE '%gmail%' GROUP BY email;

SELECT MAX(NUMERO) from cliente;

SELECT MIN(NUMERO) from cliente;

SELECT MAX(valor) FROM cliente\_transacoes;

SELECT min(valor) FROM cliente\_transacoes;

SELECT MAX(valor), tipo\_transacao\_id FROM cliente\_transacoes GROUP BY tipo\_transacao\_id;

SELECT MIN(valor), tipo\_transacao\_id FROM cliente\_transacoes GROUP BY tipo\_transacao\_id;

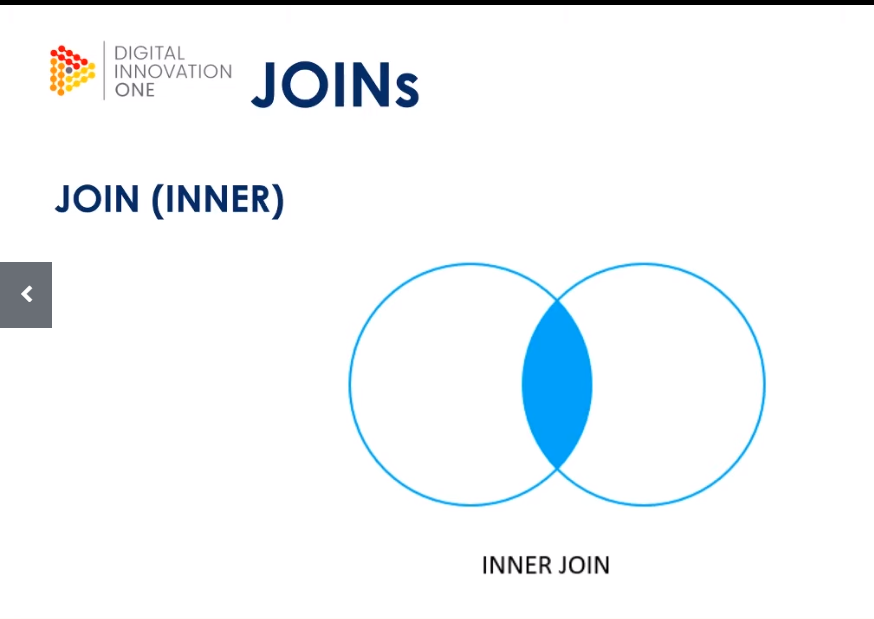
SELECT COUNT(ID), tipo\_transacao\_id FROM cliente\_transacoes GROUP BY tipo\_transacao\_id HAVING COUNT(1) > 150;

SELECT SUM(valor) FROM cliente\_transacoes ;

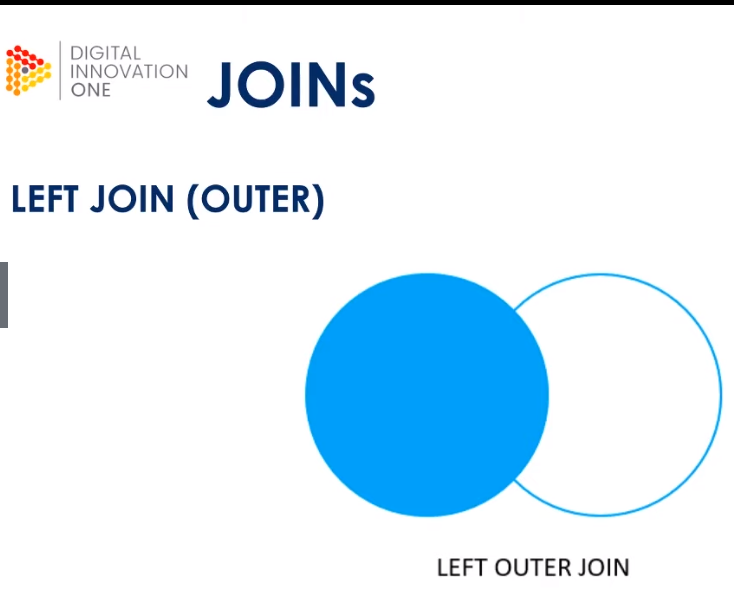
SELECT SUM(valor), tipo\_transacao\_id FROM cliente\_transacoes GROUP BY tipo\_transacao\_id ORDER BY tipo\_transacao\_id DESC;



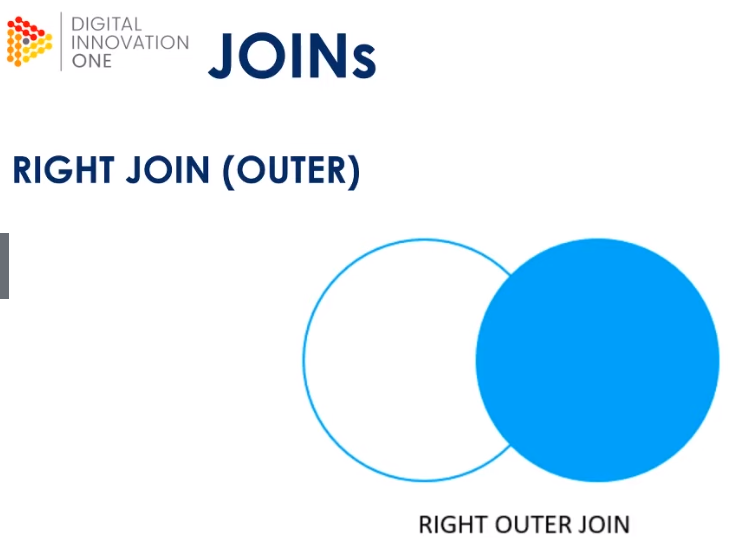




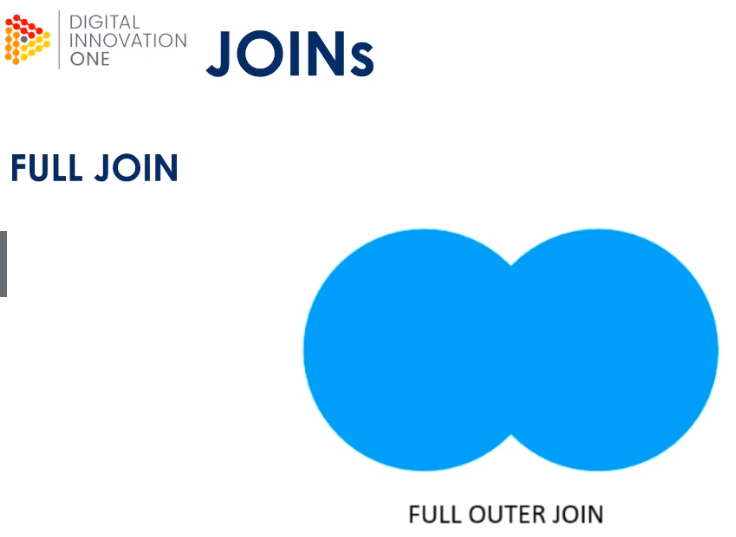














with clientes\_e\_transacoes AS (

SELECT cliente.nome as cliente\_nome,

tipo\_transacao.nome AS tipo\_transacao\_nome,

cliente\_transacoes.valor AS tipo\_transacao\_valor

FROM cliente\_transacoes

JOIN cliente ON cliente.numero = cliente\_transacoes.cliente\_numero

JOIN tipo\_transacao ON tipo\_transacao.id = cliente\_transacoes.tipo\_transacao\_id

)

SELECT cliente\_nome, tipo\_transacao\_nome,tipo\_transacao\_valor

FROM clientes\_e\_transacoes;

with clientes\_e\_transacoes AS (

SELECT cliente.nome as cliente\_nome,

tipo\_transacao.nome AS tipo\_transacao\_nome,

cliente\_transacoes.valor AS tipo\_transacao\_valor

FROM cliente\_transacoes

JOIN cliente ON cliente.numero = cliente\_transacoes.cliente\_numero

JOIN tipo\_transacao ON tipo\_transacao.id = cliente\_transacoes.tipo\_transacao\_id

JOIN banco ON banco.numero = cliente\_transacoes.banco\_numero AND banco.nome ILIKE '%Itaú%'

)

SELECT cliente\_nome, tipo\_transacao\_nome,tipo\_transacao\_valor

FROM clientes\_e\_transacoes;

CRIANDO VIEWS

CREATE OR REPLACE view vw\_bancos AS (

SELECT numero, nome, ativo

FROM banco

);

SELECT numero, nome, ativo

FROM vw\_bancos;

CREATE OR REPLACE VIEW vw\_banco\_2(

banco\_numero, banco\_nome, banco\_ativo)

AS

(

SELECT numero, nome, ativo

FROM banco

);

SELECT banco\_numero, banco\_nome, banco\_ativo

FROM vw\_banco\_2;

INSERT INTO vw\_banco\_2(banco\_numero, banco\_nome, banco\_ativo)

values(51, 'Banco Boa Ideia' , true)

SELECT banco\_numero, banco\_nome, banco\_ativo FROM vw\_banco\_2 WHERE banco\_numero = 51

SELECT numero, nome, ativo FROM banco WHERE numero = 51;

UPDATE vw\_banco\_2 SET banco\_ativo = FALSE WHERE banco\_numero = 51;

DELETE FROM vw\_banco\_2 WHERE banco\_numero =51;

SELECT numero, nome, ativo FROM banco ORDER BY NUMERO;

UPDATE banco SET ativo = false WHERE numero = 0;

BEGIN;

UPDATE banco SET ativo = true WHERE numero = 0;

COMMIT;

USANDO FUNÇÕES