

--Comandos de Banco de Dados

--1.Criação de um Database

Create database Estudos;

--Entrar no banco de dados criado (Não possuí na versão do postgresQL)

Use Estudos;

--2.Criando tabelas no banco de dados

--Create table 'nome da tabela'('Valores da tabela');

--Tabela de Alunos

Create table Alunos(

id_aluno int primary key not null, --Criando uma chave primaria

id_turma int not null, --Transformar em chave estrangeira com o comando "Alter Table"

nome varchar(50) not null,

idade int not null

);

--Tabela de Turmas

Create table Turmas(

id_turma int primary key not null,

num_turma int not null unique

);

--Tabela de Professores

Create table Professor(

id_professor int primary key not null,

id_turma int not null, --Transformar em chave estrangeira

```
id_materia int not null, --Transformar em chave estrangeira
nome varchar(50) not null,
idade int not null,
cpf varchar(11) not null unique, --Valor único para a variável, mas, permitindo
atualiza-lo
);
```

--Tabela de Matéria

```
Create table Materia(
    id_materia int primary key not null,
    materia varchar(40) not null
);
```

--3.Adicionando chaves estrangeiras nas tabelas através do "Alter Table"

--Chave estrangeira "id_turma" na tabela de alunos

Alter table Alunos --Identifica a tabela

add constraint fk_turma_aluno --Adiciona uma restrição na tabela

foreign key (id_turma) --Identifica a FK de "id_turma" da tabela de "Alunos", ela pode ser encontrada na pasta de "restrições" da tabela

references Turmas(id_turma); --Referencia com a chave primaria "id_turma" da tabela "turmas"

--Chave estrangeira "id_turma" na tabela de professores

Alter table Professor

add constraint fk_turma_professor

foreign key(id_turma)

references Turmas(id_turma);

--Chave estrangeira "id_materia" na tabela de professores

Alter table Professor

```
add constraint fk_materia_professor  
foreign key(id_materia)  
references Materia(id_materia);
```

--4.Adicionar registros em uma tabela através do comando "Insert Into"

--Obs.: Por conta das chaves estrangeiras, os registros devem ser adicionados em uma ordem específica!

--Pois não faz sentido você cadastrar um aluno sem ao menos ter uma turma.

--Insert into 'nome da tabela / (valores especificos -> opcional)' values ('valores da tabela separados por vírgula');

--Cadastrando registros na tabela de Turmas

Insert into Turmas values (1, 1); --Turma 1 --O registros são referenciados da seguinte forma: "(id_turma, nu_turma);"

Insert into Turmas values(2,2); --Turma 2

Insert into Turmas values(3,3); --Turma 3

Insert into Turmas values(4,4); --Turma 4 (Não terá ninguém associada a ela)

--Cadastrando registros na tabela de Matérias

Insert into Materia values(1, 'Sistemas Operacionais');

Insert into Materia values(2, 'Redes');

Insert into Materia values(3, 'Lógica de Programação');

Insert into Materia values(4, 'Engenharia de Software');

Insert into Materia values(5, 'Banco de Dados Relacional');

Insert into Materia values(6, 'Front-End Web'); --(Não terá ninguém associada a ela)

--Cadastrando registros na tabela de Alunos

Insert into Alunos values(1, 1, 'Joãozinho', 18); --(id_aluno, id_turma, nome, idade) O ID da turma precisa existir para poder cadastrar 1 aluno

Insert into Alunos values(2, 1, 'Pedrinho', 19);

Insert into Alunos values(3, 2, 'Mariazinha', 18);

```
Insert into Alunos values(4, 3, 'Evandrinho', 45);
```

```
--Cadastrando registros na tabela de Professores
```

```
Insert into Professor (id_professor, id_turma, id_materia, nome, idade, cpf)  
values(1,1,1,'Alex',59,'12345678910'); --Registrando campos específicos da tabela de  
professores
```

```
Insert into Professor (id_professor, id_turma, id_materia, nome, idade, cpf)  
values(2,1,2,'Eduardo',40,'12345678911');
```

```
Insert into Professor (id_professor, id_turma, id_materia, nome, idade, cpf)  
values(3,1,3,'Leandro',36,'12345678909');
```

```
Insert into Professor (id_professor, id_turma, id_materia, nome, idade, cpf)  
values(4,1,4,'Pedro',32,'10987654321');
```

```
Insert into Professor (id_professor, id_turma, id_materia, nome, idade, cpf)  
values(5,1,5,'Wanderley',57,'12312312312');
```

```
--5.Visualizar tabelas criadas com o comandos básicos de "Select"
```

```
--Select * (Tudo) from 'nome da tabela';
```

```
Select * from alunos;
```

```
Select * from professor;
```

```
Select * from turmas;
```

```
Select * from materia;
```

```
--6.Atualizando registros nas tabelas com o comando "Update"
```

```
--Update 'nome da tabela' set 'nome da coluna' = 'novo valor' where 'condição';
```

```
-- Atualizando a idade do aluno de id '4' na tabela de Alunos
```

```
Update Alunos set idade = 46 where id_aluno = 4;
```

```
Select * from alunos; --Verificar alteração
```

```
--Atualizando o nome da matéria de id '2' na tabela de Matéria
```

```
Update Materia set materia = 'Redes de Computadores' where id_materia = 2;
```

```
Select * from materia; --Verificar alteração
```

--7.Excluindo dados da tabela com o comando "Delete"

```
--Delete from 'nome da tabela' where 'condição';
```

--Adicionando um campo extra para ser excluído na tabela de Alunos

--Pre-exclusão

```
Insert into Alunos values (5,1,'Teste',1);
```

```
Select * from alunos; --Verificar inserção
```

--Pós-exclusão

```
Delete from Alunos where nome='Teste';
```

```
Select * from alunos; --Verificar alteração
```

--Cuidado com: 'Delete from Alunos;' <- Excluí todos os dados da tabela

--8.Alterando informações da tabela (Adicionar, atualizar e remover colunas)

--Adicionar Colunas: Alter table 'nome da tabela' ADD 'nome da coluna' 'tipo da coluna';

--Adicionando uma coluna na tabela de Alunos

```
Alter table Alunos ADD Teste varchar(5) default 'Teste';
```

```
Select * from alunos; --Verificar alteração
```

--Atualizar Colunas (Renomear): Alter table 'nome da tabela' rename column 'nome antigo' to 'nome novo';

--Renomeando o nome da coluna 'teste' criada para 'teste2'

```
Alter table Alunos rename column teste to teste2;
```

Select * from alunos; --Verificar alteração

--Remover Colunas: Alter table 'nome da tabela' drop column 'nome da coluna';

--Removendo a coluna 'teste2' da tabela Alunos

Alter table Alunos drop column teste2;

Select * from alunos; --Verificar alteração

--9.Realizando alguns comandos detalhados com o 'Select'

--Buscar por todos os elementos da tabela Alunos

Select * from Alunos;

--Buscando por informações específicas da tabela de Alunos (nome e idade)

Select nome,idade from Alunos;

--Buscando por pessoas que possuem idade '18' na tabela de Alunos

Select * from Alunos where idade = 18;

--Mesma condição com negação

Select * from Alunos where not idade = 18;

--Buscando por um nome de professor na tabela de Professres

Select * from Professor where nome like 'Alex';

--Mesma condição com negação

Select * from Professor where nome not like 'Alex';

--Buscando informações com concatenação na tabela de Materia

select concat('* ', materia, ' *') Materias from Materia; --'Materias' <-- Colocando apelido na nova coluna concatenada

select concat(id_materia, ' - ', materia) Materias from Materia;

--Buscando o número de caracteres do nome de um aluno na tabela Alunos

Select nome, len(nome) tamanho_nome from Alunos; --Funciona somente no SQL Server

Select nome, length(nome) tamanho_nome from Alunos; --Versão do Postgres

--Método Len como condição para busca

Select nome, length(nome) tamanho_nome from Alunos where length(nome) > 8;

--Colocando apelido nos nome da coluna de Alunos

Select nome Nome_dos_Alunos from Alunos;

--Buscando pelos professores com todos os caracteres em maiúsculas e minúsculas

--Maiúsculas

Select upper(nome) nome_maiusculo from Professor;

--Minúsculas

Select lower(nome) nome_minusculo from Professor;

--Substituindo letras dos nomes do professores da tabela Professor

Select nome, replace(nome, 'o','a') nome_letras_trocada from Professor; -- replace('nome da coluna', 'valor para ser substituido', 'valor substituto');

--Buscar certa quantidade de caracteres de uma coluna varchar da tabela Materia

Select materia, substring(materia,1,7) materia_diminutivo from Materia; --
substring('nome da coluna','valor inicial - 1', 'valor final - 7')

--10.Realizando buscas mais complexas no Select com os métodos de junção: INNER, RIGHT e LEFT JOIN.

--Inner Join (Busca as informações em comuns entre duas tabelas ou mais)

--Buscando o nome e idade da tabela 'Alunos' e juntando com o num_turma da tabela 'Turmas'

Select a.nome, a.idade, t.num_turma --Buscando os campos pelo apelido
from Alunos a --Colocando um apelido para a tabela alunos
inner join turmas t --Realizando a junção com a tabela turmas
on a.id_turma = t.id_turma; --Especificando o que as duas tabelas possuem em comum

--Buscando o nome do professor na tabela 'Professor' e juntando com a matéria da tabela 'Materia'

Select p.nome, m.materia
from Professor p
inner join Materia m
on p.id_materia = m.id_materia;

--Right Join (Realiza a junção das tabelas e busca as informações que não possuem nada em comum da outra tabela - busca a tabela da direita)

--Exemplo: a tabela Turma possui um registro que não está ligado com nenhum Alunos, então, ele exibirá "null" para aquele registro em específico.

Select a.nome, a.idade, t.num_turma
from Alunos a
right join turmas t
on a.id_turma = t.id_turma;

--Left Join (Realiza a junção das tabelas e busca as informações que não possuem nada em comum da outra tabela - busca a tabela da esquerda)

--Exemplo: a tabela Materia possui um registro que não está ligado com nenhum Professor, então, ele exibirá "null" para aquele registro em específico.

Select p.nome, m.materia
from Materia m
left join Professor p
on p.id_materia = m.id_materia;

--11.Métodos View (Visão), Stored Procedure (Procedimento de Armazenamento) e Function (Função)

--View é uma tabela virtual baseada em consultas SQL, muito útil para simplificar consultas complexas:

Create View vw_nome_alunos_maiores as --Estrutura básica de criação da View

Select nome, length(nome) tamanho_nome --Consulta básica para buscar alunos com quantidade de caracteres maiores que 8 no nome

from Alunos

where length(nome) > 8;

--Chamando a View

Select * from vw_nome_alunos_maiores;

--Stored Procedure executam ações no banco (como inserir, atualizar, etc.), mas não retornam valor diretamente como funções.

--Código para o SQL Server

Create procedure inserir_aluno

 @id_aluno int,

 @id_turma int,

 @nome nvarchar(50),

 @idade int

as

begin

 insert into Alunos values (@id_aluno, @id_turma, @nome, @idade);

end;

--Chamando o método do stored procedure

exec inserir_aluno 5,2,'Carlinhos',20;

--Function são semelhantes a procedures, mas retornam um valor e podem ser usadas em SELECT

--Código para o SQL Server

```
create function somar_idade_alunos(@idade int)
```

```
returns int
```

```
as
```

```
begin
```

```
    declare @idade int;
```

```
    set @idade += @idade;
```

```
    return @idade;
```

```
end;
```

```
--Chamando a função Function
```

```
select dbo.somar_idade_alunos(10) as idade;
```