|  |
| --- |
| #SQL - Server |
|  |  |
|  |  |
|  |  |
|  | \*\*ATALHOS\*\* |
|  |  |
|  | \* CTRL + S = Salva o script |
|  | \* CTRL + F = Pesquisa ou substitui informação dentro da consulta ou todas as consutlas abertas |
|  | \* CTRL + R = Remove ou exibe janela de consultaSELECT = Realiza uma consulta na base de dados com retorno. |
|  |  |
|  |  |
|  |  |
|  | ## Tipos de Dados SQL |
|  |  |
|  | | Numeros Exatos | Data e Hora | Cadeia de Caracteres | Unicode | |
|  | | :------------: | :-------------: | :------------------: | :-------: | |
|  | | int | \*\*date\*\* | char | nchar | |
|  | | money | datetimeoffaset | \*\*varchar\*\* | nvarchar | |
|  | | decimal | datetime2 | text | ntext | |
|  | | \*\*numeric\*\* | smalldatetime | Unicode | Binario | |
|  | | bit | datetime | nchar | binary | |
|  | | smallint | time | nvarchar | varbinary | |
|  | | smallmoney | Numeros | ntext | imagem | |
|  | | bigint | \*\*float\*\* | | | |
|  | | tinyint | real | | | |
|  |  |
|  |  |
|  |  |
|  | ##SELECT (SELECIONAR) = Realiza um consulta no banco de dados. |
|  |  |
|  | ```sql |
|  | SELECT \* FROM nome\_da\_tabela |
|  | --O '\*' (asteristico) significa selecionar todas as linhas com base na condição após o 'where' |
|  | ``` |
|  |  |
|  | ```sql |
|  | SELECT \* FROM nome\_da\_tabela WHERE nome\_da\_coluna LIKE = nome\_da\_informação |
|  | --consulta com base em uma condição |
|  | ``` |
|  |  |
|  | ```sql |
|  | SELECT nome\_da\_coluna1, nome\_da\_coluna2 FROM nome\_da\_tabela where nome\_da\_coluna\_condição like 'A' |
|  | --consulta com retorno de 2 tabelas com base em uma condição |
|  | ``` |
|  |  |
|  | ~~~sql |
|  | SELECT \* FROM nome\_da\_tabela where (SELECT nome\_da\_coluna2 FROM nome\_tabela2 where nome\_coluna2 like = 'A' ) |
|  | --consulta com condição usando outro select |
|  | ~~~ |
|  |  |
|  |  |
|  |  |
|  | ## SELECT DISTINCT (SELECIONAR ÚNICO) = Realiza uma consulta com valores nao duplicados, se estiver duplicados retornará apenas um deles. |
|  |  |
|  | ```sql |
|  | SELECT DISTINCT \* FROM nome\_da\_tabela; |
|  | --percorre toda a tabela e retorna valores sem duplicidade |
|  | ``` |
|  |  |
|  | ```sql |
|  | SELECT DISTINCT nome\_da\_coluna1, nome\_da\_coluna2 FROM nome\_da\_tabela |
|  | --percorre toda a tabela e retorna valores sem duplicidade apenas das colunas fornecidas |
|  | ``` |
|  |  |
|  |  |
|  |  |
|  | ## WHERE (AQUI) = É uma cláusula para fazer um consulta em uma coluna específica, com uma condição. |
|  |  |
|  | ```sql |
|  | SELECT \* FROM nome\_da\_tabela |
|  | WHERE nome\_da\_coluna like 'joao' |
|  | --percorre toda a tabela e retorna as linhas que possui o nome joão na coluna 1 |
|  | ``` |
|  |  |
|  | ```sql |
|  | SELECT nome\_da\_tabela FROM nome\_da\_tabela |
|  | WHERE nome\_da\_coluna2 = 1; |
|  | --condição direta sem 'like' |
|  | ``` |
|  |  |
|  |  |
|  |  |
|  | ## AND, OR, NOT, AND NOT( E, OU, E NAO) = São operadores usados em consultas para trazer mais valores, ou usados como condição para trazer esses valores. |
|  |  |
|  | \*\*AND\*\* |
|  |  |
|  | ```sql |
|  | SELECT \* FROM nome\_da\_tabela WHERE condição1 AND condição2 AND condition3 ... |
|  | --retorna os dados com base nas condições estabelecidas |
|  | ``` |
|  |  |
|  | \*\*OR\*\* |
|  |  |
|  | ```sql |
|  | SELECT \* FROM nome\_da\_tabela WHERE condição1 OR condição2 OR condition3 OR ... |
|  | ----retorna os dados com base nas condições estabelecidas, uma após a outra, caso a primeira nao tenha resultado positivo |
|  | ``` |
|  |  |
|  | \*\*NOT\*\* |
|  |  |
|  | ```sql |
|  | SELECT \* FROM nome\_da\_tabela WHERE NOT nome\_da\_coluna1 = 3 |
|  | --retorna os dados, con exceção da linha em que contém a condição |
|  | ``` |
|  |  |
|  | \*\*AND NOT\*\* |
|  |  |
|  | ~~~sql |
|  | SELECT \* FROM nome\_da\_tabela WHERE NOT coluna1 ='dado1' AND NOT coluna2 ='dado2' |
|  | --retorna todos os dados com excelção das duas condições |
|  | ~~~ |
|  |  |
|  |  |
|  |  |
|  | ##ORDER BY( ORDENE POR ) = É usado para retornar a consulta em ordem crescente ou decrescente. |
|  |  |
|  | \*\*Crescente\*\* |
|  |  |
|  | ```sql |
|  | SELECT \* FROM tabela ORDER BY coluna ASC |
|  | ``` |
|  |  |
|  | \*\*Decrescente\*\* |
|  |  |
|  | ```sql |
|  | SELECT \* FROM tabela ORDER BY coluna DESC |
|  | ``` |
|  |  |
|  |  |
|  |  |
|  | ##INSERT INTO ( INSERIR DENTRO!!! ) = Inserir um novo valor na tabela |
|  |  |
|  | ```sql |
|  | INSERT INTO nome\_tabela (coluna1, coluna2) VALUES (valor1, valor2) |
|  | --irá inserir dois valores nas colunas específicadas |
|  | ``` |
|  |  |
|  |  |
|  |  |
|  | ##NULL, NOT NULL ( VAZIO, NAO VAZIO) |
|  |  |
|  | ~~~sql |
|  | SELECT coluna1, coluna2 FROM tabela |
|  | WHERE coluna1 IS NULL; |
|  | --essa consulta está informando que a coluna 1 RETORNARÁ valores nulos |
|  | ~~~ |
|  |  |
|  | ~~~sql |
|  | SELECT coluna1, coluna2 FROM tabela |
|  | WHERE coluna2 IS NOT NULL; |
|  | --essa consulta está informando que a coluna 2 NÃO tratá valores nulos |
|  | ~~~ |
|  |  |
|  |  |
|  |  |
|  | ##UPDATE ( ATUALIZAR!!! ) = Substitui as informações em uma ou mais tabelas |
|  |  |
|  | ```sql |
|  | UPDATE tabela1 |
|  | SET nome\_coluna = 'valor novo' |
|  | WHERE coluna\_condição = 'valor' |
|  | ``` |
|  |  |
|  | \*\*Exemplo\*\* |
|  |  |
|  | \*\*Tabela (funcionarios)\*\* |
|  |  |
|  | | \*\*ID\*\* | \*\*Nome\*\* | \*\*Departamento\*\* | |
|  | | ------ | --------------- | ---------------- | |
|  | | 1 | Pedro Henrique | Compras | |
|  | | 2 | Lucas da Silva | Administrativo | |
|  | | 3 | Luana Gabriela | Comercial | |
|  | | 4 | Renato Lazuna | Administrativo | |
|  | | 5 | Rafaela Barrosa | Administrativo | |
|  | | 6 | Arline da Costa | Comercial | |
|  | | 7 | Giseli Pascoa | Comercial | |
|  | | 8 | Renan Pereira | Compras | |
|  |  |
|  | \*\*Tabela (supervisor)\*\* |
|  |  |
|  | | \*\*Id\_funcionario\*\* | \*\*data\_admissao\*\* | |
|  | | ------------------ | ----------------- | |
|  | | 1 | 2012-05-05 | |
|  | | 2 | 2015-04-01 | |
|  | | 3 | 2015-01-01 | |
|  | | 4 | 2017-01-01 | |
|  | | 5 | 2017-02-01 | |
|  | | 6 | 2017-01-01 | |
|  |  |
|  | ~~~sql |
|  | UPDATE |
|  | funcionario F |
|  | SET |
|  | departamento = 'Supervisão Comercial' |
|  | WHERE id IN |
|  | ( SELECT id\_funcionario |
|  | FROM supervisor S |
|  | WHERE S.id\_funcionario = F.id |
|  | AND F.departamento = 'Comercial') |
|  | --essa query fará com que todos os funcionários do departamento 'comercial' sejam 'supervisores' |
|  | ~~~ |
|  |  |
|  |  |
|  |  |
|  | ##DELETE ( DELETAR!!! ) = Apagar os dados da tabela |
|  |  |
|  | ```sql |
|  | DELETE FROM nome\_tabela WHERE nome\_coluna = 'valor' |
|  | --colocar WHERE logo após DELETE EX: DELETE WHERE |
|  | ``` |
|  |  |
|  |  |
|  |  |
|  | ## SELECT TOP ( SELECIONA OS PRIMERIOS ) = Seleciona uma quantidada específica de linhas da tabela |
|  |  |
|  | ~~~sql |
|  | SELECT top 10 \* FROM nome\_tabela |
|  | --seleciona as primeiras 10 linhas da tabela |
|  | ~~~ |
|  |  |
|  | ~~~sql |
|  | SELECT TOP 15 nome\_coluna FROM nome\_tabela |
|  | --seleciona as primeiras 15 linhas apenas da coluna especificada |
|  | ~~~ |
|  |  |
|  |  |
|  |  |
|  | ##MIN ( MÍNIMO ) = Seleciona o menor valor da condição |
|  |  |
|  | ```sql |
|  | SELECT MIN(nome\_coluna) FROM tabela WHERE condição; |
|  | --consulta mínimo valor com condição |
|  | ``` |
|  |  |
|  | ~~~sql |
|  | select MIN(nome\_coluna) from tabela |
|  | --consulta mínimo valor sem condição |
|  | ~~~ |
|  |  |
|  |  |
|  |  |
|  | ##MAX ( MÁXIMO ) = Seleciona o maior valor |
|  |  |
|  | ```sql |
|  | SELECT MAX(nome\_da\_coluna) FROM nome\_tabela WHERE condição; |
|  | --retorna o maior valor da coluna com base em uma condição |
|  | ``` |
|  |  |
|  |  |
|  |  |
|  | ##COUNT ( CONTAR ) = Faz a contagem dos valores |
|  |  |
|  | ```sql |
|  | SELECT COUNT(nome\_coluna) FROM nome\_tabela; |
|  | --retorna a quantidade de linhas da tabela |
|  | ``` |
|  |  |
|  | ~~~sql |
|  | select COUNT(coluna\_x) from coluna\_y with(nolock) where coluna\_x = 'valor' |
|  | --a consulta acima conta quantas veses aparece a informação 'valor' da coluna\_x na tabela\_y |
|  | ~~~ |
|  |  |
|  |  |
|  |  |
|  | ##AVG ( MÉDIA ) = Faz uma média dos valores |
|  |  |
|  | ```sql |
|  | SELECT AVG(coluna) FROM tabela |
|  | --retorna o valor médio de uma coluna numérica |
|  | ``` |
|  |  |
|  |  |
|  |  |
|  | ##SUM ( SOMAR ) = Soma todos os valores |
|  |  |
|  | ```sql |
|  | --retorna a soma total de uma coluna numérica |
|  | SELECT SUM(column\_name) FROM table\_name WHERE condition; |
|  | ``` |
|  |  |
|  |  |
|  |  |
|  | ##LIKE ( PARECIDO ) = É usado em uma cláusula "where" como uma condição para retornar um ou mais valores específicos |
|  |  |
|  | ```sql |
|  | SELECT coluna1, coluna2 FROM tabela WHERE coluna1 LIKE 13; |
|  | --retornar a consulta con a condição da coluna ter o valor 13 |
|  |  |
|  | /\*### OUTROS EXEMPLOS ###\*/ |
|  | SELECT column1, column2, ... |
|  | FROM table\_name |
|  | WHERE columnN LIKE condição 1 AND condição 2; |
|  | --com condição OR |
|  | SELECT column1, column2, ... |
|  | FROM table\_name |
|  | WHERE columnN LIKE condição 1 OR condição 2 |
|  | --procura clientes com o nome começando com 'F' |
|  | SELECT \* FROM cadastros |
|  | WHERE CustomerName LIKE 'B%'; |
|  | --procura clientes com o nome terminando com 'F' |
|  | SELECT \* FROM cadastros |
|  | WHERE CustomerName LIKE '%e'; |
|  | --procura clientes que possuam 'ei' em qual quer posição |
|  | SELECT \* FROM cadastros |
|  | WHERE CustomerName LIKE '%sc%'; |
|  | --procura clientes que possuem 'u' na 3º posição, '\_' = um caracter |
|  | SELECT \* FROM cadastros |
|  | WHERE CustomerName LIKE '\_\_u%'; |
|  | --procura cliente que começar com 'a' e termina com 'o' |
|  | SELECT \* FROM cadastros |
|  | WHERE ContactName LIKE 'a%t'; |
|  | --procura cliente que NÃO POSSUI 'a' |
|  | SELECT \* FROM cadastros |
|  | WHERE CustomerName NOT LIKE 'a%'; |
|  | --seleciona todos que iniciam com b,c, ou d |
|  | SELECT \* FROM cadastros |
|  | WHERE City LIKE '[bcd]%'; |
|  |  |
|  |  |
|  | ##IN ( EM ) = Usado como uma condição para uma valor que está 'em' um determinado campo |
|  |  |
|  | ```sql |
|  | SELECT \* FROM tabela WHERE coluna\_país IN ('Brasil', 'Alemanha', 'Japão') |
|  | --Seleciona todos os clientes que são dos paises entre ( ) |
|  | SELECT coluna1, coluna2 FROM tabela WHERE coluna3 in (valor\_condição1, valor\_condição2) |
|  | --seleciona os valores da coluna 1 e coluna 2 com base nos valores que estão na coluna 3 da mesma tabela |
|  | ``` |
|  |  |
|  |  |
|  |  |
|  | ##BETWEEN ( ESTEJA ENTRE ) = Usado como condição para procurar um valor que estar 'entre' dois determinados valores |
|  |  |
|  | ```sql |
|  | SELECT \* FROM Produtos WHERE Price BETWEEN 10 AND 20; |
|  | --seleciona todos os preços que estão entre 10 e 20 |
|  | ``` |
|  |  |
|  | Outro |
|  |  |
|  | ```sql |
|  | --seleciona produtos que estão entre 10 e 20, mas não possuem o ID 1,2,3 |
|  | SELECT \* FROM Produtos |
|  | WHERE Preço BETWEEN 10 AND 20 |
|  | AND CategoriaID NOT IN (1,2,3); |
|  | ``` |
|  |  |
|  | Exemplo com texto |
|  |  |
|  | ```sql |
|  | SELECT \* FROM Produtos |
|  | WHERE NomeProduto BETWEEN 'Carnarvon Tigers' AND 'Mozzarella di Giovanni' |
|  | ORDER BY NomeProduto; |
|  | ``` |
|  |  |
|  |  |
|  |  |
|  | ##ALIAS ( APELIDO TEMPORARIO ) |
|  |  |
|  | ```sql |
|  | --cria um nome temporário para a tabela |
|  | SELECT Coluna1 AS alias\_name |
|  | FROM nome\_tabela; |
|  | ``` |
|  |  |
|  | Exemplo |
|  |  |
|  | ```sql |
|  | SELECT ClienteID AS ID, NomeCLiente AS Cliente |
|  | FROM Clientes; |
|  | ``` |
|  |  |
|  |  |
|  |  |
|  | ## IS NULL, IS NOT NULL ( É VAZIO, NÃO É VAZIO ) = É usado para testar valores nulos dentro da tabela |
|  |  |
|  | Ex (is null): \*\*select\*\* coluna \*\*from\*\* tabela \*\*where\*\* coluna \*\*is\*\* \*\*null\*\*; |
|  |  |
|  | Ex (is not null): \*\*select\*\* coluna \*\*from\*\* tabela \*\*where\*\* coluna \*\*is\*\* \*\*not\*\* \*\*null\*\*; |
|  |  |
|  | ```sql |
|  | SELECT CustomerName, ContactName, Address |
|  | FROM Customers |
|  | WHERE Address IS NULL; |
|  | ``` |
|  |  |
|  | ```sql |
|  | SELECT CustomerName, ContactName, Address |
|  | FROM Customers |
|  | WHERE Address IS NOT NULL; |
|  | ``` |
|  |  |
|  |  |
|  |  |
|  | ##Group BY ( AGRUPE POR ) = Faz o agrupamento de linhas de uma tabela, com base na linha de outra coluna da mesma tabela |
|  |  |
|  | Geralmente usado com uma condição como: |
|  |  |
|  | \* Count |
|  | \* Sum |
|  | \* AVG |
|  |  |
|  | ~~~sql |
|  | SELECT column\_name(s) |
|  | FROM table\_name |
|  | WHERE condition |
|  | GROUP BY column\_name(s) |
|  | ORDER BY column\_name(s); |
|  | ~~~ |
|  |  |
|  | Ex: Conte Produtos da HP tem na coluna A, com base na coluna B |
|  |  |
|  | | Coluna A | Coluna B | |
|  | | -------- | -------- | |
|  | | HP | Notebook | |
|  | | HP | Notebook | |
|  | | HP | Gabinete | |
|  | | HP | Notebook | |
|  | | HP | Notebook | |
|  | | HP | Gabinete | |
|  | | HP | Gabinete | |
|  |  |
|  | ~~~sql |
|  | select count(ColunaA) from table with(nolock) Group By ColunaB |
|  | --5 |
|  | --4 |
|  | ~~~ |
|  |  |
|  |  |
|  |  |
|  | ##HAVING ( HAVENDO ) = Cláusula similar á cláusula 'where', usada após o 'GROUP BY' |
|  |  |
|  | ~~~sql |
|  | SELECT column\_name(s) |
|  | FROM table\_name |
|  | WHERE condition |
|  | GROUP BY column\_name(s) |
|  | HAVING condition |
|  | ORDER BY column\_name(s); |
|  | ~~~ |
|  |  |
|  | EX: A instrução SQL a seguir lista o número de clientes em cada país. Incluir apenas países com mais de 5 clientes: |
|  |  |
|  | | CustomerID | CustomerName | ContactName | Address | City | PostalCode | Country | |
|  | | ---------- | ---------------------------------- | ------------------ | ----------------------------- | ----------- | ---------- | ------- | |
|  | | 1 | Alfreds Futterkiste | Maria Anders | Obere Str. 57 | Berlin | 12209 | Germany | |
|  | | 2 | Ana Trujillo Emparedados y helados | Ana Trujillo | Avda. de la Constitución 2222 | México D.F. | 05021 | Mexico | |
|  | | 3 | Antonio Moreno Taquería | Antonio Moreno | Mataderos 2312 | México D.F. | 05023 | Mexico | |
|  | | 4 | Around the Horn | Thomas Hardy | 120 Hanover Sq. | London | WA1 1DP | UK | |
|  | | 5 | Berglunds snabbköp | Christina Berglund | Berguvsvägen 8 | Luleå | S-958 22 | Sweden | |
|  |  |
|  | Script: |
|  |  |
|  | ~~~sql |
|  | SELECT COUNT(CustomerID), Country |
|  | FROM Customers |
|  | GROUP BY Country |
|  | HAVING COUNT(CustomerID) > 5; |
|  | ~~~ |
|  |  |
|  | --select |
|  | select \* fom tabela |
|  |  |
|  | select \* from pessoa\_trabalho |
|  | select \* from pessoa\_endereco |
|  | select \* from pessoa\_dados |
|  |  |
|  | select \* from pessoa\_trabalho |
|  | select \* from pessoa\_endereco |
|  | ```sql |
|  | SELECT \* FROM nome\_da\_tabela |
|  | --O '\*' (asteristico) significa selecionar todas as linhas com base na condição após o 'where' |
|  | select \* from pessoa\_trabalho |
|  |  |
|  | SELECT \* FROM nome\_da\_tabela WHERE nome\_da\_coluna LIKE = nome\_da\_informação |
|  | --consulta com base em uma condição |
|  | select \* from pessoa\_trabalho where funcao='advogado' |
|  |  |
|  | SELECT nome\_da\_coluna1, nome\_da\_coluna2 FROM nome\_da\_tabela where nome\_da\_coluna\_condição like 'A' |
|  | --consulta com retorno de 2 tabelas com base em uma condição |
|  | select codigo\_cliente,ramo from pessoa\_trabalho where ramo='financeiro' |
|  |  |
|  | --SELECT \* FROM nome\_da\_tabela where (SELECT nome\_da\_coluna2 FROM nome\_tabela2 where nome\_coluna2 like = 'A' ) |
|  | --consulta com condição usando outro select |
|  | --select \* from pessoa\_trabalho where (select rua from pessoa\_endereco where rua='centro') |
|  |  |
|  | SELECT DISTINCT \* FROM nome\_da\_tabela; |
|  | --percorre toda a tabela e retorna valores sem duplicidade |
|  | Select DISTINCT \* from pessoa\_trabalho; |
|  |  |
|  | SELECT DISTINCT nome\_da\_coluna1, nome\_da\_coluna2 FROM nome\_da\_tabela |
|  | --percorre toda a tabela e retorna valores sem duplicidade apenas das colunas fornecidas |
|  | SELECT DISTINCT ramo,funcao FROM pessoa\_trabalho; |
|  |  |
|  | SELECT \* FROM nome\_da\_tabela |
|  | WHERE nome\_da\_coluna like 'joao' |
|  | --percorre toda a tabela e retorna as linhas que possui o nome joão na coluna 1 |
|  | select \* from pessoa\_dados where nome\_pessoa like 'joao' |
|  |  |
|  | SELECT \* from pessoa\_dados |
|  | SELECT \* from pessoa\_endereco |
|  |  |
|  | select \* from pessoa\_dados |
|  | select codigo\_cliente,nome\_pessoa,cpf\_pessoa from pessoa\_dados where cpf\_pessoa='76585312509' |
|  |  |
|  | SELECT \* FROM nome\_da\_tabela WHERE condição1 AND condição2 AND condition3 ... |
|  | --retorna os dados com base nas condições estabelecidas |
|  | select \* from pessoa\_dados where nome\_pessoa='maria' and cpf\_pessoa='33422311278' and numero\_telefone='91943345566' |
|  |  |
|  | SELECT \* FROM nome\_da\_tabela WHERE condição1 OR condição2 OR condition3 OR ... |
|  | ----retorna os dados com base nas condições estabelecidas, uma após a outra, caso a primeira nao tenha resultado positivo |
|  | select \* from pessoa\_dados WHERE nome\_pessoa='maria' or nome\_pessoa='joao' |
|  |  |
|  | SELECT \* FROM nome\_da\_tabela WHERE NOT nome\_da\_coluna1 = 3 |
|  | --retorna os dados, con exceção da linha em que contém a condição |
|  | select \* from pessoa\_dados WHERE NOT codigo\_cliente = 1024 |
|  |  |
|  | SELECT \* FROM nome\_da\_tabela WHERE NOT coluna1 ='dado1' AND NOT coluna2 ='dado2' |
|  | --retorna todos os dados com excelção das duas condições |
|  | SELECT \* FROM pessoa\_dados WHERE NOT nome\_pessoa='MARIA' AND NOT nome\_pessoa='JOAO' |
|  |  |
|  | SELECT \* FROM tabela ORDER BY coluna ASC |
|  | SELECT \* FROM pessoa\_dados ORDER BY nome\_pessoa ASC |
|  |  |
|  | SELECT \* FROM tabela ORDER BY coluna DESC |
|  | SELECT \* FROM pessoa\_dados ORDER BY nome\_pessoa DESC |
|  |  |
|  | INSERT INTO nome\_tabela (coluna1, coluna2) VALUES (valor1, valor2) |
|  | --irá inserir dois valores nas colunas específicadas |
|  | INSERT INTO pessoa\_dados (codigo\_cliente,nome\_pessoa,cpf\_pessoa,numero\_telefone) VALUES (1019,'PIERRE',' ',' '); |
|  |  |
|  | UPDATE tabela1 |
|  | SET nome\_coluna = 'valor novo' |
|  | WHERE coluna\_condição = 'valor' |
|  |  |
|  | UPDATE pessoa\_dados |
|  | SET numero\_telefone = '91987352394' |
|  | WHERE codigo\_cliente = '1019' |
|  |  |
|  |  |
|  |  |
|  | select \* from Funcionarios |
|  | insert into Funcionarios (Id\_funcionario,Nome\_funcionario,Departamento) values (1,'Pedro Henrique','Compras'); |
|  | insert into funcionarios (Id\_funcionario,Nome\_funcionario,Departamento) values (2,'Lucas da Silva','Administrativo'); |
|  | insert into funcionarios (Id\_funcionario,Nome\_funcionario,Departamento) values (3,'Luana Gabriela','Comercial'); |
|  | insert into funcionarios (Id\_funcionario,Nome\_funcionario,Departamento) values (4,'Renato Lazuna','Administrativo'); |
|  | insert into funcionarios (Id\_funcionario,Nome\_funcionario,Departamento) values (5,'Rafaela Barrosa','Administrativo'); |
|  | insert into funcionarios (Id\_funcionario,Nome\_funcionario,Departamento) values (6,'Arline da Costa','Comercial'); |
|  | insert into funcionarios (Id\_funcionario,Nome\_funcionario,Departamento) values (7,'Gisele Pascoa','Comercial'); |
|  | insert into funcionarios (Id\_funcionario,Nome\_funcionario,Departamento) values (8,'Renan Pereira','Compras'); |
|  |  |
|  | create table Supervisor |
|  | ( |
|  | Id\_funcionario int not null, |
|  | data\_admissao datetime |
|  | ); |
|  | select \* from supervisor |
|  |  |
|  | insert into supervisor (Id\_funcionario,data\_admissao) values (1,'2012-05-05'); |
|  | insert into supervisor (Id\_funcionario,data\_admissao) values (2,'2015-01-01'); |
|  | insert into supervisor (Id\_funcionario,data\_admissao) values (3,'2017-01-01'); |
|  | insert into supervisor (Id\_funcionario,data\_admissao) values (4,'2017-02-01'); |
|  | insert into supervisor (Id\_funcionario,data\_admissao) values (5,'2017-01-01'); |
|  |  |
|  | update supervisor |
|  | set data\_admissao='2015-05-05' |
|  | where Id\_funcionario=1 |
|  |  |
|  | select \* from funcionarios |
|  | UPDATE |
|  | Funcionarios |
|  | SET |
|  | Departamento = 'Supervisores' |
|  | WHERE Id\_funcionario IN |
|  | ( SELECT id\_funcionario |
|  | FROM Funcionarios |
|  | WHERE id\_funcionario = Id\_funcionario |
|  | AND departamento = 'Comercial'); |
|  | --essa query fará com que todos os funcionários do departamento 'comercial' sejam 'supervisores' |
|  |  |
|  |  |
|  | SELECT top 5 \* FROM Funcionarios |
|  | --seleciona as primeiras 10 linhas da tabela |
|  |  |
|  | ##MIN ( MÍNIMO ) = Seleciona o menor valor da condição |
|  |  |
|  | ```sql |
|  | SELECT MIN(codigo\_cliente) FROM pessoa\_endereco where numero=23; |
|  | --consulta mínimo valor com condição |
|  |  |
|  | select \* from pessoa\_endereco |
|  |  |
|  | ```sql |
|  | SELECT COUNT(codigo\_cliente) FROM pessoa\_endereco; |
|  | --retorna a quantidade de linhas da tabela |
|  | ```sql |
|  | SELECT AVG(numero) FROM pessoa\_endereco |
|  | --retorna o valor médio de uma coluna numérica |
|  |  |
|  | select \* from pessoa\_endereco |
|  |  |
|  | SELECT rua, numero,cidade |
|  | FROM pessoa\_endereco |
|  | WHERE rua LIKE 'centro' and cidade='belem'; |
|  | --com condição OR |
|  | SELECT rua, numero |
|  | FROM pessoa\_endereco |
|  | WHERE rua LIKE 'centro' OR cidade='cameta'; |
|  | --procura clientes com o nome começando com 'F' |
|  | SELECT \* FROM pessoa\_dados |
|  | WHERE nome\_pessoa LIKE 'F%'; |
|  | --procura clientes com o nome terminando com 'F' |
|  | SELECT \* FROM pessoa\_dados |
|  | WHERE nome\_pessoa LIKE '%o'; |
|  | --procura clientes que possuam 'ei' em qual quer posição |
|  | SELECT \* FROM pessoa\_dados |
|  | WHERE nome\_pessoa LIKE '%er%'; |
|  | --procura clientes que possuem 'u' na 3º posição, '\_' = um caracter |
|  | SELECT \* FROM pessoa\_dados |
|  | WHERE nome\_pessoa LIKE '\_\_u%'; |
|  | --procura cliente que começar com 'a' e termina com 'o' |
|  | SELECT \* FROM pessoa\_dados |
|  | WHERE nome\_pessoa LIKE 'a%o'; |
|  | --procura cliente que NÃO POSSUI 'a' |
|  | SELECT \* FROM pessoa\_dados |
|  | WHERE nome\_pessoa NOT LIKE 'a%'; |
|  | --seleciona todos que iniciam com b,c, ou d |
|  | SELECT \* FROM pessoa\_dados |
|  | WHERE nome\_pessoa LIKE '[bcd]%'; |
|  | ``` |
|  |  |
|  | select \* from cadastros |
|  | ##IN ( EM ) = Usado como uma condição para uma valor que está 'em' um determinado campo |
|  |  |
|  | ```sql |
|  | SELECT \* FROM cadastros WHERE country IN ('France', 'Germany', 'Canada'); |
|  | --Seleciona todos os clientes que são dos paises entre ( ) |
|  |  |
|  | SELECT customername, contactname FROM cadastros WHERE postalcode in ('5021', 'Germany', '67000', 'France'); |
|  | --seleciona os valores da coluna 1 e coluna 2 com base nos valores que estão na coluna 3 da mesma tabela |
|  | ``` |
|  |  |
|  | ##BETWEEN ( ESTEJA ENTRE ) = Usado como condição para procurar um valor que estar 'entre' dois determinados valores |
|  |  |
|  | select \* from produtos |
|  | SELECT \* FROM Produtos WHERE prince BETWEEN 10 AND 20; |
|  | --seleciona todos os preços que estão entre 10 e 20 |
|  | ``` |
|  |  |
|  | select \* from produtos |
|  | Exemplo com texto |
|  |  |
|  | SELECT \* FROM cadastros |
|  | WHERE customername BETWEEN 'Antonio Moreno Taqueria' AND 'Bs Beverages' |
|  | ORDER BY contactname; |
|  |  |
|  | ##ALIAS ( APELIDO TEMPORARIO ) |
|  | select \* from pedidos |
|  | --cria um nome temporário para a tabela |
|  | SELECT employeeId AS teste |
|  | FROM pedidos; |
|  | ``` |
|  |  |
|  | Exemplo |
|  | select \* from cadastros |
|  | SELECT customerID AS ID, contactname AS Cadastro |
|  | FROM cadastros; |
|  | ``` |
|  |  |
|  | Geralmente usado com uma condição como: |
|  |  |
|  | \* Count |
|  | \* Sum |
|  | \* AVG |
|  | select \* from cadastros |
|  | ~~~sql |
|  | SELECT column\_name(s) |
|  | FROM table\_name |
|  | WHERE condition |
|  | GROUP BY column\_name(s) |
|  | ORDER BY column\_name(s); |
|  | ~~~ |
|  |  |
|  | Ex: Conte Produtos da HP tem na coluna A, com base na coluna B |
|  |  |
|  | | Coluna A | Coluna B | |
|  | | -------- | -------- | |
|  | | HP | Notebook | |
|  | | HP | Notebook | |
|  | | HP | Gabinete | |
|  | | HP | Notebook | |
|  | | HP | Notebook | |
|  | | HP | Gabinete | |
|  | | HP | Gabinete | |
|  |  |
|  | ~~~sql |
|  | select count(ColunaA) from table with(nolock) Group By ColunaB |
|  | --5 |
|  | --4 |
|  | ~~~ |
|  |  |
|  |  |
|  |  |
|  | ##HAVING ( HAVENDO ) = Cláusula similar á cláusula 'where', usada após o 'GROUP BY' |
|  |  |
|  | ~~~sql |
|  | SELECT column\_name(s) |
|  | FROM table\_name |
|  | WHERE condition |
|  | GROUP BY column\_name(s) |
|  | HAVING condition |
|  | ORDER BY column\_name(s); |
|  | ~~~ |
|  |  |
|  | EX: A instrução SQL a seguir lista o número de clientes em cada país. Incluir apenas países com mais de 5 clientes: |
|  |  |
|  | Script: |
|  |  |
|  | ~~~sql |
|  | SELECT COUNT Country |
|  | FROM Cadastros |
|  | GROUP BY Country |
|  | HAVING COUNT Country > 3; |
|  |  |
|  |  |
|  |  |
|  | select nome, admissao from cadfun where ADMISSAO between '09/01/1980' and '11/12/1980'; |
|  |  |
|  | select nome, admissao from cadfun where ADMISSAO between '05/01/1980' and '10/12/2022'; |
|  |  |
|  | select nome, admissao from cadfun |
|  | where ADMISSAO between '05/01/1980' and '10/12/2022' order by nome asc; |
|  |  |
|  | select nome, admissao from cadfun |
|  | where ADMISSAO between '05/01/1980' and '10/12/2022' order by nome desc; |
|  |  |
|  | select nome, admissao from cadfun |
|  | where ADMISSAO between '01/10/2016' and '22/10/2016'; |
|  |  |
|  | select nome, admissao from cadfun |
|  | where DEPTO='5'; |
|  |  |
|  | select nome, DEPTO, admissao from cadfun |
|  | where admissao between '05/01/1980' and '10/12/2022' and DEPTO='5'; |
|  |  |
|  | select nome,depto, lower(nome) from cadfun where (depto='3') or (depto='5'); |
|  |  |
|  | select nome, admissao,DEPTO from cadfun |
|  | where ADMISSAO between ('05/01/1900') and ('31/10/2022') and DEPTO='3' or depto='5'; |
|  |  |
|  | select nome, DEPTO, FUNCAO,ADMISSAO from cadfun |
|  | where ADMISSAO between ('01/08/1900') and ('15/10/2022') and FUNCAO='analista'; |
|  |  |
|  |  |
|  |  |
|  | SELECT \* FROM cadfun |
|  |  |
|  | SELECT DEPTO, COUNT(\*) FROM cadfun GROUP BY DEPTO; |
|  |  |
|  | SELECT FUNCAO, COUNT(\*) FROM CADFUN GROUP BY FUNCAO; |
|  |  |
|  | SELECT DEPTO, SUM(SALARIO) FROM cadfun GROUP BY DEPTO; |
|  |  |
|  | SELECT DEPTO, SUM(SALARIO) FROM cadfun WHERE DEPTO IN ('2','4') GROUP BY DEPTO; |
|  |  |
|  | SELECT DEPTO, SUM(SALARIO) FROM cadfun GROUP BY DEPTO HAVING SUM(SALARIO) > 8000; |
|  |  |
|  | SELECT \* FROM morto |
|  | SELECT \* FROM cadfun |
|  |  |
|  | SELECT CODFUN, NOME FROM cadfun UNION SELECT CODFUN, NOME FROM MORTO; |
|  |  |
|  | INSERT INTO MORTO VALUES (21,'EPAMINONDAS DA SILVA','4','PROGRAMADOR',1155.00,'10/10/2016',2); |
|  |  |
|  | SELECT CODFUN, NOME FROM cadfun UNION SELECT CODFUN, NOME FROM MORTO; |
|  | SELECT CODFUN, NOME FROM cadfun UNION ALL SELECT CODFUN, NOME FROM MORTO; |
|  |  |
|  |  |
|  | SELECT NOME, CIDADE, ESTADO FROM cliente; |
|  |  |
|  | INSERT INTO cliente VALUES ('820','MECANICA SAO PAULO', 'R. DO MACUCO, 99','SANTO ANTONIO','ES','29810-020'); |
|  | INSERT INTO cliente VALUES ('170','POSTO BRASIL LTDA', 'AV. IMPERIO, 85','GUAGIRUS','BA','42837-000'); |
|  | INSERT INTO cliente VALUES ('340','TRANSP. CARGA PESADA LTDA', 'AV. DOS AUTONOMISTAS, 1000','OSASCO','SP','06020-010'); |
|  | INSERT INTO cliente VALUES ('100','MICROS INFORMATICA S/A', 'R. DAS PALMEIRAS, 4 - LOJA 2','SAO PAULO','SP','01226-010'); |
|  | INSERT INTO cliente VALUES ('750','IMOBILIARIA PAVAO', 'AV. BRASIL, 105','RIO DO PRADO','MG','39940-111'); |
|  | INSERT INTO cliente VALUES ('860','ASSIS CONTABIL S/C LTDA', 'R. DO MONUMENTO, 550','SANTO AMARO','BA','44200-090'); |
|  | INSERT INTO cliente VALUES ('230','SUPERMERCADO BOTAFOGO', 'R. DA LAGOA, 999','RIO DAS OSTRAS','RJ','28890-540'); |
|  | INSERT INTO cliente VALUES ('150','MARCONDES E IRMAO LTDA', 'R. DO ORATORIO, 66','ROSARIO OESTE','MT','78470-010'); |
|  | INSERT INTO cliente VALUES ('800','DOCEIRA PAO DE ACUCAR', 'R. MARTINS PENA, 200','SANTO ANDRE','SP','09190-700'); |
|  | INSERT INTO cliente VALUES ('990','METALURGICA FOGO E FERRO', 'R. ARTUR BERNANDES, 3500','SANTO ANDRE','SP','09193-410'); |
|  | INSERT INTO cliente VALUES ('500','JOSE DANTAS E FILHOS S/C LTDA', 'AV. IPIRANGA, 1000','LUCRECIA','RN','59805-010'); |
|  | INSERT INTO cliente VALUES ('300','PCTEC - MICROCOMPUTADORES S/A', 'R. CAIS DO PORTO, ARMAZEM 3','SANTOS','SP','11000-005'); |
|  |  |
|  |  |
|  | SELECT \* FROM venda |
|  | SELECT \* FROM cliente |
|  |  |
|  | SELECT venda.DUPLIC, cliente.NOME from cliente, venda |
|  | where cliente.CODCLI = venda.CODCLI; |
|  |  |
|  | SELECT cliente.NOME, venda.DUPLIC, venda.VALOR from cliente, venda |
|  | where cliente.CODCLI = venda.CODCLI ORDER BY cliente.NOME |
|  |  |
|  | select \* from cadfun; |
|  |  |
|  | select AVG(FILHOS) FROM cadfun; |
|  |  |
|  | select AVG(FILHOS) FROM cadfun |
|  | WHERE DEPTO='5'; |
|  |  |
|  | SELECT COUNT(salario) FROM cadfun |
|  | WHERE DEPTO='5'; |
|  |  |
|  | SELECT nome, salario, depto FROM cadfun |
|  | WHERE DEPTO='5'; |
|  |  |
|  | SELECT sum(salario) FROM cadfun |
|  | WHERE funcao='analista'; |
|  |  |
|  | SELECT max(salario) FROM cadfun |
|  | WHERE depto='5'; |
|  |  |
|  | SELECT sum(salario) FROM cadfun |
|  | WHERE funcao='analista'; |
|  |  |
|  | SELECT sum(salario) FROM cadfun |
|  | WHERE funcao='analista' and depto='2'; |
|  |  |
|  | SELECT NOME, admissao from cadfun |
|  | where day(admissao)=10; |
|  |  |
|  | SELECT NOME, admissao from cadfun |
|  | where day(admissao)>5 and day(admissao)<10; |
|  |  |
|  | SELECT NOME, admissao from cadfun |
|  | where day(admissao)>5 and day(admissao)<10 order by NOME; |
|  |  |
|  | SELECT NOME, admissao from cadfun |
|  | where day(admissao)>5 and day(admissao)<10 order by NOME desc; |
|  |  |
|  | SELECT NOME, admissao from cadfun |
|  | where admissao<='20/09/2006'; |
|  |  |
|  | SELECT NOME, admissao, datename(month,ADMISSAO) from cadfun |
|  | where depto='5'; |
|  |  |
|  | select nome, depto, lower(nome) from cadfun |
|  | where depto='3' or depto='5'; |
|  |  |
|  | SELECT NOME, admissao, depto from cadfun |
|  | where day(admissao)>5 and day(admissao)<10 and depto='3' or depto='5'; |
|  |  |
|  | SELECT NOME, admissao, depto from cadfun |
|  | where day(admissao)>30 and day(admissao)<20 and depto='3' or depto='5'; |
|  |  |
|  |  |
|  | --select |
|  | select \* fom tabela |
|  |  |
|  | select \* from pessoa\_trabalho |
|  | select \* from pessoa\_endereco |
|  | select \* from pessoa\_dados |
|  |  |
|  | select \* from pessoa\_trabalho |
|  | select \* from pessoa\_endereco |
|  | ```sql |
|  | SELECT \* FROM nome\_da\_tabela |
|  | --O '\*' (asteristico) significa selecionar todas as linhas com base na condição após o 'where' |
|  | select \* from pessoa\_trabalho |
|  |  |
|  | SELECT \* FROM nome\_da\_tabela WHERE nome\_da\_coluna LIKE = nome\_da\_informação |
|  | --consulta com base em uma condição |
|  | select \* from pessoa\_trabalho where funcao='advogado' |
|  |  |
|  | SELECT nome\_da\_coluna1, nome\_da\_coluna2 FROM nome\_da\_tabela where nome\_da\_coluna\_condição like 'A' |
|  | --consulta com retorno de 2 tabelas com base em uma condição |
|  | select codigo\_cliente,ramo from pessoa\_trabalho where ramo='financeiro' |
|  |  |
|  | --SELECT \* FROM nome\_da\_tabela where (SELECT nome\_da\_coluna2 FROM nome\_tabela2 where nome\_coluna2 like = 'A' ) |
|  | --consulta com condição usando outro select |
|  | --select \* from pessoa\_trabalho where (select rua from pessoa\_endereco where rua='centro') |
|  |  |
|  | SELECT DISTINCT \* FROM nome\_da\_tabela; |
|  | --percorre toda a tabela e retorna valores sem duplicidade |
|  | Select DISTINCT \* from pessoa\_trabalho; |
|  |  |
|  | SELECT DISTINCT nome\_da\_coluna1, nome\_da\_coluna2 FROM nome\_da\_tabela |
|  | --percorre toda a tabela e retorna valores sem duplicidade apenas das colunas fornecidas |
|  | SELECT DISTINCT ramo,funcao FROM pessoa\_trabalho; |
|  |  |
|  | SELECT \* FROM nome\_da\_tabela |
|  | WHERE nome\_da\_coluna like 'joao' |
|  | --percorre toda a tabela e retorna as linhas que possui o nome joão na coluna 1 |
|  | select \* from pessoa\_dados where nome\_pessoa like 'joao' |
|  |  |
|  | SELECT \* from pessoa\_dados |
|  | SELECT \* from pessoa\_endereco |
|  |  |
|  | select \* from pessoa\_dados |
|  | select codigo\_cliente,nome\_pessoa,cpf\_pessoa from pessoa\_dados where cpf\_pessoa='76585312509' |
|  |  |
|  | SELECT \* FROM nome\_da\_tabela WHERE condição1 AND condição2 AND condition3 ... |
|  | --retorna os dados com base nas condições estabelecidas |
|  | select \* from pessoa\_dados where nome\_pessoa='maria' and cpf\_pessoa='33422311278' and numero\_telefone='91943345566' |
|  |  |
|  | SELECT \* FROM nome\_da\_tabela WHERE condição1 OR condição2 OR condition3 OR ... |
|  | ----retorna os dados com base nas condições estabelecidas, uma após a outra, caso a primeira nao tenha resultado positivo |
|  | select \* from pessoa\_dados WHERE nome\_pessoa='maria' or nome\_pessoa='joao' |
|  |  |
|  | SELECT \* FROM nome\_da\_tabela WHERE NOT nome\_da\_coluna1 = 3 |
|  | --retorna os dados, con exceção da linha em que contém a condição |
|  | select \* from pessoa\_dados WHERE NOT codigo\_cliente = 1024 |
|  |  |
|  | SELECT \* FROM nome\_da\_tabela WHERE NOT coluna1 ='dado1' AND NOT coluna2 ='dado2' |
|  | --retorna todos os dados com excelção das duas condições |
|  | SELECT \* FROM pessoa\_dados WHERE NOT nome\_pessoa='MARIA' AND NOT nome\_pessoa='JOAO' |
|  |  |
|  | SELECT \* FROM tabela ORDER BY coluna ASC |
|  | SELECT \* FROM pessoa\_dados ORDER BY nome\_pessoa ASC |
|  |  |
|  | SELECT \* FROM tabela ORDER BY coluna DESC |
|  | SELECT \* FROM pessoa\_dados ORDER BY nome\_pessoa DESC |
|  |  |
|  | INSERT INTO nome\_tabela (coluna1, coluna2) VALUES (valor1, valor2) |
|  | --irá inserir dois valores nas colunas específicadas |
|  | INSERT INTO pessoa\_dados (codigo\_cliente,nome\_pessoa,cpf\_pessoa,numero\_telefone) VALUES (1019,'PIERRE',' ',' '); |
|  |  |
|  | UPDATE tabela1 |
|  | SET nome\_coluna = 'valor novo' |
|  | WHERE coluna\_condição = 'valor' |
|  |  |
|  | UPDATE pessoa\_dados |
|  | SET numero\_telefone = '91987352394' |
|  | WHERE codigo\_cliente = '1019' |
|  |  |
|  |  |
|  |  |
|  | select \* from Funcionarios |
|  | insert into Funcionarios (Id\_funcionario,Nome\_funcionario,Departamento) values (1,'Pedro Henrique','Compras'); |
|  | insert into funcionarios (Id\_funcionario,Nome\_funcionario,Departamento) values (2,'Lucas da Silva','Administrativo'); |
|  | insert into funcionarios (Id\_funcionario,Nome\_funcionario,Departamento) values (3,'Luana Gabriela','Comercial'); |
|  | insert into funcionarios (Id\_funcionario,Nome\_funcionario,Departamento) values (4,'Renato Lazuna','Administrativo'); |
|  | insert into funcionarios (Id\_funcionario,Nome\_funcionario,Departamento) values (5,'Rafaela Barrosa','Administrativo'); |
|  | insert into funcionarios (Id\_funcionario,Nome\_funcionario,Departamento) values (6,'Arline da Costa','Comercial'); |
|  | insert into funcionarios (Id\_funcionario,Nome\_funcionario,Departamento) values (7,'Gisele Pascoa','Comercial'); |
|  | insert into funcionarios (Id\_funcionario,Nome\_funcionario,Departamento) values (8,'Renan Pereira','Compras'); |
|  |  |
|  | create table Supervisor |
|  | ( |
|  | Id\_funcionario int not null, |
|  | data\_admissao datetime |
|  | ); |
|  | select \* from supervisor |
|  |  |
|  | insert into supervisor (Id\_funcionario,data\_admissao) values (1,'2012-05-05'); |
|  | insert into supervisor (Id\_funcionario,data\_admissao) values (2,'2015-01-01'); |
|  | insert into supervisor (Id\_funcionario,data\_admissao) values (3,'2017-01-01'); |
|  | insert into supervisor (Id\_funcionario,data\_admissao) values (4,'2017-02-01'); |
|  | insert into supervisor (Id\_funcionario,data\_admissao) values (5,'2017-01-01'); |
|  |  |
|  | update supervisor |
|  | set data\_admissao='2015-05-05' |
|  | where Id\_funcionario=1 |
|  |  |
|  | select \* from funcionarios |
|  | UPDATE |
|  | Funcionarios |
|  | SET |
|  | Departamento = 'Supervisores' |
|  | WHERE Id\_funcionario IN |
|  | ( SELECT id\_funcionario |
|  | FROM Funcionarios |
|  | WHERE id\_funcionario = Id\_funcionario |
|  | AND departamento = 'Comercial'); |
|  | --essa query fará com que todos os funcionários do departamento 'comercial' sejam 'supervisores' |
|  |  |
|  |  |
|  | SELECT top 5 \* FROM Funcionarios |
|  | --seleciona as primeiras 10 linhas da tabela |
|  |  |
|  | ##MIN ( MÍNIMO ) = Seleciona o menor valor da condição |
|  |  |
|  | ```sql |
|  | SELECT MIN(codigo\_cliente) FROM pessoa\_endereco where numero=23; |
|  | --consulta mínimo valor com condição |
|  |  |
|  | select \* from pessoa\_endereco |
|  |  |
|  | ```sql |
|  | SELECT COUNT(codigo\_cliente) FROM pessoa\_endereco; |
|  | --retorna a quantidade de linhas da tabela |
|  | ```sql |
|  | SELECT AVG(numero) FROM pessoa\_endereco |
|  | --retorna o valor médio de uma coluna numérica |
|  |  |
|  | select \* from pessoa\_endereco |
|  |  |
|  | SELECT rua, numero,cidade |
|  | FROM pessoa\_endereco |
|  | WHERE rua LIKE 'centro' and cidade='belem'; |
|  | --com condição OR |
|  | SELECT rua, numero |
|  | FROM pessoa\_endereco |
|  | WHERE rua LIKE 'centro' OR cidade='cameta'; |
|  | --procura clientes com o nome começando com 'F' |
|  | SELECT \* FROM pessoa\_dados |
|  | WHERE nome\_pessoa LIKE 'F%'; |
|  | --procura clientes com o nome terminando com 'F' |
|  | SELECT \* FROM pessoa\_dados |
|  | WHERE nome\_pessoa LIKE '%o'; |
|  | --procura clientes que possuam 'ei' em qual quer posição |
|  | SELECT \* FROM pessoa\_dados |
|  | WHERE nome\_pessoa LIKE '%er%'; |
|  | --procura clientes que possuem 'u' na 3º posição, '\_' = um caracter |
|  | SELECT \* FROM pessoa\_dados |
|  | WHERE nome\_pessoa LIKE '\_\_u%'; |
|  | --procura cliente que começar com 'a' e termina com 'o' |
|  | SELECT \* FROM pessoa\_dados |
|  | WHERE nome\_pessoa LIKE 'a%o'; |
|  | --procura cliente que NÃO POSSUI 'a' |
|  | SELECT \* FROM pessoa\_dados |
|  | WHERE nome\_pessoa NOT LIKE 'a%'; |
|  | --seleciona todos que iniciam com b,c, ou d |
|  | SELECT \* FROM pessoa\_dados |
|  | WHERE nome\_pessoa LIKE '[bcd]%'; |
|  | ``` |
|  |  |
|  | select \* from cadastros |
|  | ##IN ( EM ) = Usado como uma condição para uma valor que está 'em' um determinado campo |
|  |  |
|  | ```sql |
|  | SELECT \* FROM cadastros WHERE country IN ('France', 'Germany', 'Canada'); |
|  | --Seleciona todos os clientes que são dos paises entre ( ) |
|  |  |
|  | SELECT customername, contactname FROM cadastros WHERE postalcode in ('5021', 'Germany', '67000', 'France'); |
|  | --seleciona os valores da coluna 1 e coluna 2 com base nos valores que estão na coluna 3 da mesma tabela |
|  | ``` |
|  |  |
|  | ##BETWEEN ( ESTEJA ENTRE ) = Usado como condição para procurar um valor que estar 'entre' dois determinados valores |
|  |  |
|  | select \* from produtos |
|  | SELECT \* FROM Produtos WHERE prince BETWEEN 10 AND 20; |
|  | --seleciona todos os preços que estão entre 10 e 20 |
|  | ``` |
|  |  |
|  | select \* from produtos |
|  | Exemplo com texto |
|  |  |
|  | SELECT \* FROM cadastros |
|  | WHERE customername BETWEEN 'Antonio Moreno Taqueria' AND 'Bs Beverages' |
|  | ORDER BY contactname; |
|  |  |
|  | ##ALIAS ( APELIDO TEMPORARIO ) |
|  | select \* from pedidos |
|  | --cria um nome temporário para a tabela |
|  | SELECT employeeId AS teste |
|  | FROM pedidos; |
|  | ``` |
|  |  |
|  | Exemplo |
|  | select \* from cadastros |
|  | SELECT customerID AS ID, contactname AS Cadastro |
|  | FROM cadastros; |
|  | ``` |
|  |  |
|  | Geralmente usado com uma condição como: |
|  |  |
|  | \* Count |
|  | \* Sum |
|  | \* AVG |
|  | select \* from cadastros |
|  | ~~~sql |
|  | SELECT column\_name(s) |
|  | FROM table\_name |
|  | WHERE condition |
|  | GROUP BY column\_name(s) |
|  | ORDER BY column\_name(s); |
|  | ~~~ |
|  |  |
|  | Ex: Conte Produtos da HP tem na coluna A, com base na coluna B |
|  |  |
|  | | Coluna A | Coluna B | |
|  | | -------- | -------- | |
|  | | HP | Notebook | |
|  | | HP | Notebook | |
|  | | HP | Gabinete | |
|  | | HP | Notebook | |
|  | | HP | Notebook | |
|  | | HP | Gabinete | |
|  | | HP | Gabinete | |
|  |  |
|  | ~~~sql |
|  | select count(ColunaA) from table with(nolock) Group By ColunaB |
|  | --5 |
|  | --4 |
|  | ~~~ |
|  |  |
|  |  |
|  |  |
|  | ##HAVING ( HAVENDO ) = Cláusula similar á cláusula 'where', usada após o 'GROUP BY' |
|  |  |
|  | ~~~sql |
|  | SELECT column\_name(s) |
|  | FROM table\_name |
|  | WHERE condition |
|  | GROUP BY column\_name(s) |
|  | HAVING condition |
|  | ORDER BY column\_name(s); |
|  | ~~~ |
|  |  |
|  | EX: A instrução SQL a seguir lista o número de clientes em cada país. Incluir apenas países com mais de 5 clientes: |
|  |  |
|  | Script: |
|  |  |
|  | ~~~sql |
|  | SELECT COUNT Country |
|  | FROM Cadastros |
|  | GROUP BY Country |
|  | HAVING COUNT Country > 5; |
|  |  |
|  |  |
|  |  |
|  | select nome, admissao from cadfun where ADMISSAO between '09/01/1980' and '11/12/1980'; |
|  |  |
|  | select nome, admissao from cadfun where ADMISSAO between '05/01/1980' and '10/12/2022'; |
|  |  |
|  | select nome, admissao from cadfun |
|  | where ADMISSAO between '05/01/1980' and '10/12/2022' order by nome asc; |
|  |  |
|  | select nome, admissao from cadfun |
|  | where ADMISSAO between '05/01/1980' and '10/12/2022' order by nome desc; |
|  |  |
|  | select nome, admissao from cadfun |
|  | where ADMISSAO between '01/10/2016' and '22/10/2016'; |
|  |  |
|  | select nome, admissao from cadfun |
|  | where DEPTO='5'; |
|  |  |
|  | select nome, DEPTO, admissao from cadfun |
|  | where admissao between '05/01/1980' and '10/12/2022' and DEPTO='5'; |
|  |  |
|  | select nome,depto, lower(nome) from cadfun where (depto='3') or (depto='5'); |
|  |  |
|  | select nome, admissao,DEPTO from cadfun |
|  | where ADMISSAO between ('05/01/1900') and ('31/10/2022') and DEPTO='3' or depto='5'; |
|  |  |
|  | select nome, DEPTO, FUNCAO,ADMISSAO from cadfun |
|  | where ADMISSAO between ('01/08/1900') and ('15/10/2022') and FUNCAO='analista'; |
|  |  |
|  |  |
|  |  |
|  | SELECT \* FROM cadfun |
|  |  |
|  | SELECT DEPTO, COUNT(\*) FROM cadfun GROUP BY DEPTO; |
|  |  |
|  | SELECT FUNCAO, COUNT(\*) FROM CADFUN GROUP BY FUNCAO; |
|  |  |
|  | SELECT DEPTO, SUM(SALARIO) FROM cadfun GROUP BY DEPTO; |
|  |  |
|  | SELECT DEPTO, SUM(SALARIO) FROM cadfun WHERE DEPTO IN ('2','4') GROUP BY DEPTO; |
|  |  |
|  | SELECT DEPTO, SUM(SALARIO) FROM cadfun GROUP BY DEPTO HAVING SUM(SALARIO) > 8000; |
|  |  |
|  | SELECT \* FROM morto |
|  | SELECT \* FROM cadfun |
|  |  |
|  | SELECT CODFUN, NOME FROM cadfun UNION SELECT CODFUN, NOME FROM MORTO; |
|  |  |
|  | INSERT INTO MORTO VALUES (21,'EPAMINONDAS DA SILVA','4','PROGRAMADOR',1155.00,'10/10/2016',2); |
|  |  |
|  | SELECT CODFUN, NOME FROM cadfun UNION SELECT CODFUN, NOME FROM MORTO; |
|  | SELECT CODFUN, NOME FROM cadfun UNION ALL SELECT CODFUN, NOME FROM MORTO; |
|  |  |
|  |  |
|  | SELECT NOME, CIDADE, ESTADO FROM cliente; |
|  |  |
|  | INSERT INTO cliente VALUES ('820','MECANICA SAO PAULO', 'R. DO MACUCO, 99','SANTO ANTONIO','ES','29810-020'); |
|  | INSERT INTO cliente VALUES ('170','POSTO BRASIL LTDA', 'AV. IMPERIO, 85','GUAGIRUS','BA','42837-000'); |
|  | INSERT INTO cliente VALUES ('340','TRANSP. CARGA PESADA LTDA', 'AV. DOS AUTONOMISTAS, 1000','OSASCO','SP','06020-010'); |
|  | INSERT INTO cliente VALUES ('100','MICROS INFORMATICA S/A', 'R. DAS PALMEIRAS, 4 - LOJA 2','SAO PAULO','SP','01226-010'); |
|  | INSERT INTO cliente VALUES ('750','IMOBILIARIA PAVAO', 'AV. BRASIL, 105','RIO DO PRADO','MG','39940-111'); |
|  | INSERT INTO cliente VALUES ('860','ASSIS CONTABIL S/C LTDA', 'R. DO MONUMENTO, 550','SANTO AMARO','BA','44200-090'); |
|  | INSERT INTO cliente VALUES ('230','SUPERMERCADO BOTAFOGO', 'R. DA LAGOA, 999','RIO DAS OSTRAS','RJ','28890-540'); |
|  | INSERT INTO cliente VALUES ('150','MARCONDES E IRMAO LTDA', 'R. DO ORATORIO, 66','ROSARIO OESTE','MT','78470-010'); |
|  | INSERT INTO cliente VALUES ('800','DOCEIRA PAO DE ACUCAR', 'R. MARTINS PENA, 200','SANTO ANDRE','SP','09190-700'); |
|  | INSERT INTO cliente VALUES ('990','METALURGICA FOGO E FERRO', 'R. ARTUR BERNANDES, 3500','SANTO ANDRE','SP','09193-410'); |
|  | INSERT INTO cliente VALUES ('500','JOSE DANTAS E FILHOS S/C LTDA', 'AV. IPIRANGA, 1000','LUCRECIA','RN','59805-010'); |
|  | INSERT INTO cliente VALUES ('300','PCTEC - MICROCOMPUTADORES S/A', 'R. CAIS DO PORTO, ARMAZEM 3','SANTOS','SP','11000-005'); |
|  |  |
|  |  |
|  | SELECT \* FROM venda |
|  | SELECT \* FROM cliente |
|  |  |
|  | SELECT venda.DUPLIC, cliente.NOME from cliente, venda |
|  | where cliente.CODCLI = venda.CODCLI; |
|  |  |
|  | SELECT cliente.NOME, venda.DUPLIC, venda.VALOR from cliente, venda |
|  | where cliente.CODCLI = venda.CODCLI ORDER BY cliente.NOME |
|  |  |
|  | select \* from cadfun; |
|  |  |
|  | select AVG(FILHOS) FROM cadfun; |
|  |  |
|  | select AVG(FILHOS) FROM cadfun |
|  | WHERE DEPTO='5'; |
|  |  |
|  | SELECT COUNT(salario) FROM cadfun |
|  | WHERE DEPTO='5'; |
|  |  |
|  | SELECT nome, salario, depto FROM cadfun |
|  | WHERE DEPTO='5'; |
|  |  |
|  | SELECT sum(salario) FROM cadfun |
|  | WHERE funcao='analista'; |
|  |  |
|  | SELECT max(salario) FROM cadfun |
|  | WHERE depto='5'; |
|  |  |
|  | SELECT sum(salario) FROM cadfun |
|  | WHERE funcao='analista'; |
|  |  |
|  | SELECT sum(salario) FROM cadfun |
|  | WHERE funcao='analista' and depto='2'; |
|  |  |
|  | SELECT NOME, admissao from cadfun |
|  | where day(admissao)=10; |
|  |  |
|  | SELECT NOME, admissao from cadfun |
|  | where day(admissao)>5 and day(admissao)<10; |
|  |  |
|  | SELECT NOME, admissao from cadfun |
|  | where day(admissao)>5 and day(admissao)<10 order by NOME; |
|  |  |
|  | SELECT NOME, admissao from cadfun |
|  | where day(admissao)>5 and day(admissao)<10 order by NOME desc; |
|  |  |
|  | SELECT NOME, admissao from cadfun |
|  | where admissao<='20/09/2006'; |
|  |  |
|  | SELECT NOME, admissao, datename(month,ADMISSAO) from cadfun |
|  | where depto='5'; |
|  |  |
|  | select nome, depto, lower(nome) from cadfun |
|  | where depto='3' or depto='5'; |
|  |  |
|  | SELECT NOME, admissao, depto from cadfun |
|  | where day(admissao)>5 and day(admissao)<10 and depto='3' or depto='5'; |
|  |  |
|  |  |
|  |  |
|  | -- junção das tabelas |
|  |  |
|  | select \* from venda |
|  | select \* from cliente |
|  |  |
|  | --consultar as duplicatas referente as tabelas venda e cliente |
|  | select venda.DUPLIC, cliente.NOME from cliente, venda |
|  | where cliente.CODCLI = venda.CODCLI; |
|  |  |
|  | -- ordenar por nome a consulta das tabelas venda e cliente |
|  | select venda.DUPLIC, cliente.NOME from cliente, venda |
|  | where cliente.CODCLI = venda.CODCLI order by cliente.NOME; |
|  |  |
|  | -- inserindo mais dados para exemplificar a junção |
|  | insert into venda values ('235100',1500.00,'06/12/2015','500'); |
|  | insert into venda values ('203052',9008.33,'22/08/2015','550'); |
|  | insert into venda values ('922452',1211.98,'09/03/2016','340'); |
|  | insert into venda values ('999820',3110.22,'05/11/2015','170'); |
|  | insert into venda values ('223345',2112.11,'08/03/2016','230'); |
|  | insert into venda values ('222228',2390.00,'10/07/2016','170'); |
|  | insert into venda values ('111211',3535.00,'15/11/2016','230'); |
|  | insert into venda values ('907754',2056.90,'30/10/2015','340'); |
|  | insert into venda values ('347711',5092.55,'20/11/2016','170'); |
|  | insert into venda values ('209967',9008.33,'10/05/2016','550'); |
|  |  |
|  |  |
|  | SELECT \* FROM cliente |
|  | SELECT \* FROM venda |
|  |  |
|  | -- Consulta que mostra código, o nome a quantidade de titulos e o total da dívida de cada cliente. |
|  | select cliente.NOME, count(\*) AS titulos, SUM(venda.VALOR) AS TOTAL |
|  | FROM cliente, venda where cliente.codcli = venda.codcli |
|  | group by cliente.nome; |
|  |  |
|  | -- Consulta numeros de titulos vencidos agrupados e ordenados pelo nome da tabela cliente |
|  | select cliente.NOME AS CLIENTE, COUNT(\*) AS VENCIDOS |
|  | FROM cliente, venda where cliente.codcli = venda.codcli |
|  | AND VENCTO <='2003-12-31' group by cliente.nome ORDER BY cliente.NOME; |
|  |  |
|  | -- Consulta de duplicata em atrazo anterior a data 31/12/1999, |
|  | -- apresentando nome do cliente o valor da duplicata e dos juros |
|  | select cliente.NOME, venda.VALOR, venda.VALOR \* 0.10 AS JUROS, venda.VALOR \* 1.10 AS TOTAL |
|  | FROM cliente, venda where cliente.codcli = venda.codcli |
|  | AND VENCTO <='1999-12-31' ORDER BY cliente.NOME; |
|  |  |
|  | select cliente.NOME,VENDA.DUPLIC, venda.VALOR, venda.VALOR \* 0.10 AS JUROS, venda.VALOR \* 1.10 AS TOTAL |
|  | FROM cliente, venda where cliente.codcli = venda.codcli |
|  | ORDER BY cliente.NOME; |
|  |  |
|  | select \* from cadfun |
|  |  |
|  | -- Visualização de Dados |
|  | -- Cria uma tabela virtual a partir dos dados da tabela cadfun |
|  |  |
|  | create view visao1 AS SELECT NOME,DEPTO,SALARIO FROM cadfun; |
|  |  |
|  | select \* from visao1 |
|  |  |
|  | -- Visão definida para apresentar os títulos em atraso a partir de 31/12/2005. |
|  |  |
|  | create view visao2 AS SELECT cliente.NOME AS CLIENTE, |
|  | COUNT(\*) AS VENCIDOS FROM CLIENTE, VENDA WHERE cliente.CODCLI = venda.CODCLI |
|  | AND VENCTO <='2005-12-31' GROUP BY cliente.NOME; |
|  |  |
|  | SELECT \* FROM visao2 |
|  |  |
|  | -- Visão3 combinação completa de todos os dados das tabelas cliente e venda. |
|  |  |
|  | create view visao3 AS SELECT cliente.CODCLI, cliente.NOME, venda.DUPLIC, |
|  | venda.VALOR,venda.VENCTO FROM cliente, venda WHERE cliente.CODCLI = venda.CODCLI |
|  |  |
|  | SELECT \* FROM visao3 |
|  |  |
|  | --Indexação de Tabelas |
|  |  |
|  | create index indice1 on cadfun (nome); |
|  |  |
|  | alter table cadfun add CPF char(11); |
|  |  |
|  | update cadfun set CPF = '10020011199' WHERE CODFUN = 1; |
|  | update cadfun set CPF = '10020022299' WHERE CODFUN = 6; |
|  | update cadfun set CPF = '10020033399' WHERE CODFUN = 8; |
|  | update cadfun set CPF = '10020044499' WHERE CODFUN = 10; |
|  | update cadfun set CPF = '10020055599' WHERE CODFUN = 11; |
|  | update cadfun set CPF = '10020066699' WHERE CODFUN = 12; |
|  | update cadfun set CPF = '10020077799' WHERE CODFUN = 20; |
|  | update cadfun set CPF = '10020088899' WHERE CODFUN = 21; |
|  | update cadfun set CPF = '10022211199' WHERE CODFUN = 22; |
|  | update cadfun set CPF = '10022233399' WHERE CODFUN = 24; |
|  |  |
|  |  |
|  |  |
|  | create database curso; |
|  |  |
|  | -- Chave, relacionamento e cardialidade |
|  | -- Criar as tabelas cadpro e caddis |
|  |  |
|  | create table cadpro ( |
|  | CODPRO INT NOT NULL PRIMARY KEY, |
|  | NUMEROPRO VARCHAR(40) NOT NULL, |
|  | CPF CHAR (11) NOT NULL UNIQUE, |
|  | ); |
|  |  |
|  | create table caddis ( |
|  | CODDIS CHAR(6) NOT NULL, |
|  | NUMEDIS VARCHAR(40) NOT NULL, |
|  | CODPROF INT NOT NULL FOREIGN KEY REFERENCES cadpro(CODPRO) |
|  | ON DELETE NO ACTION |
|  | ON UPDATE NO ACTION |
|  | ); |
|  | -- Inserindo dados na tabela cadpro |
|  | INSERT INTO cadpro VALUES(100,'SILVANA SOUZA','11122233399'); |
|  | INSERT INTO cadpro VALUES(120,'RENATO DE ABREU','11122244477'); |
|  | INSERT INTO cadpro VALUES(110,'JOSE PAULO SILVA','11122244488'); |
|  | INSERT INTO cadpro VALUES(130,'PENELOPE DA SILVA','11122255566'); |
|  | INSERT INTO cadpro VALUES(140,'JULIANA DE ALBUQUERQUE','11122266655'); |
|  | INSERT INTO cadpro VALUES(150,'CARLOS MUNHOZ DA SILVA','11122277744'); |
|  |  |
|  | --Inserindo dados na tabela caddis |
|  | INSERT INTO caddis VALUES ('CG-100','MATEMATICA',100); |
|  | INSERT INTO caddis VALUES ('CG-200','PORTUGUES',110); |
|  | INSERT INTO caddis VALUES ('TI-100','ALGORITMOS',120); |
|  | INSERT INTO caddis VALUES ('TI-205','LINGUAGEM DE PROGRAMAÇAO II',130); |
|  | INSERT INTO caddis VALUES ('TI-400','SISTEMAS OPERACIONAIS',140); |
|  | INSERT INTO caddis VALUES ('CG-110','FISICA',100); |
|  | INSERT INTO caddis VALUES ('CG-300','FILOSOFIA',110); |
|  | INSERT INTO caddis VALUES ('TI-200','LINGUAGEM DE PROGRAMACAO I',120); |
|  | INSERT INTO caddis VALUES ('TI-300','BANCO DE DADOS',130); |
|  | INSERT INTO caddis VALUES ('TI-500','PROGRAMACAO WEB',140); |
|  | INSERT INTO caddis VALUES ('CG-100','MATEMÁTICA',150); |
|  | INSERT INTO caddis VALUES ('CG-300','FILOSOFIA',150); |
|  |  |
|  |  |
|  | use curso |
|  |  |
|  | SELECT \* FROM caddis; |
|  | SELECT \* FROM cadpro; |
|  |  |
|  | -- Consulta apresentara a relação de disciplina ofertadas e os professores que as ministram |
|  | SELECT caddis.CODDIS, caddis.NUMEDIS, cadpro.NOMEPRO |
|  | FROM caddis, cadpro WHERE caddis.CODPROF = cadpro.CODPROP; |
|  |  |
|  | -- Procedimentos Armazenados |
|  | create procedure sp\_saudacao AS |
|  | DECLARE @Mensagem CHAR(11) |
|  | SET @Mensagem = 'Alo Mundo' |
|  | PRINT @Mensagem |
|  |  |
|  | -- Execute o comando abaixo |
|  | exec sp\_saudacao; |
|  |  |
|  | -- Outro exemplo |
|  | create procedure sp\_dados AS |
|  | SELECT NOME, SALARIO FROM cadfun |
|  | --Execute o comando abaixo |
|  | exec sp\_dados |
|  |  |
|  |  |
|  | create procedure sp\_decisao2 @valor INT AS |
|  | DECLARE @Resto INTEGER |
|  | DECLARE @Mensagem1 VARCHAR(50) |
|  | DECLARE @Mensagem2 VARCHAR(50) |
|  | SET @Resto = @valor % 3 |
|  | SET @Mensagem1 = 'e um valor divisivel por 3' |
|  | SET @Mensagem2 = 'nao e um valor divisivel por 3' |
|  | IF (@Resto = 0) |
|  | BEGIN |
|  | PRINT @Valor |
|  | PRINT @Mensagem1 |
|  | END |
|  | ELSE |
|  | BEGIN |
|  | PRINT @Valor |
|  | PRINT @Mensagem2 |
|  | END |
|  | ; |
|  |  |
|  | EXEC sp\_decisao2 4; |
|  | EXEC sp\_decisao2 9; |
|  |  |
|  | -- Teste de programaçao |
|  |  |
|  | select \* from venda |
|  | select \* from cliente |
|  |  |
|  | select cliente.nome, count(\*) as total from cliente,venda |
|  | where cliente.CODCLI = venda.CODCLI |
|  | group by cliente.NOME order by cliente.NOME asc; |
|  |  |
|  | select cliente.nome, count(\*) as TITULOS, sum(venda.valor) as TOTAL from cliente, venda |
|  | where cliente.CODCLI = venda.CODCLI |
|  | group by cliente.NOME; |
|  |  |
|  | use virtualdc |
|  |  |
|  | select cliente.NOME as cliente, count(\*) as vencidos from cliente,venda |
|  | where cliente.CODCLI = venda.CODCLI and VENCTO <= '2003-12-31' |
|  | group by cliente.NOME order by cliente.NOME; |
|  |  |
|  | select cliente.NOME, venda.VALOR, venda.VALOR \* 0.10 as JUROS, |
|  | venda.VALOR \*1.10 as TOTAL, VENCTO, DUPLIC,CIDADE from cliente, venda where cliente.CODCLI = venda.CODCLI |
|  | and VENCTO <='1999-12-31' order by cliente.NOME |
|  |  |
|  | --Visualização de dados |
|  | CREATE VIEW visao1 as Select NOME, DEPTO, SALARIO from cadfun; |
|  |  |
|  | select \* from visao1 |
|  |  |
|  | create view visao2 as Select cliente.NOME as cliente, count(\*) as VENCIDOS |
|  | FROM cliente, venda WHERE cliente.CODCLI = venda.CODCLI |
|  | AND VENCTO <= '2005-12-31' GROUP BY cliente.NOME |
|  |  |
|  | select \* from visao2 |
|  |  |
|  | create view visao3 as Select cliente.CODCLI, cliente.NOME, venda.DUPLIC, |
|  | venda.VALOR,venda.VENCTO from cliente, venda where cliente.CODCLI=venda.CODCLI |
|  |  |
|  | select \* from visao3 |
|  |  |
|  | select \* from caddis; |
|  | select \* from cadpro; |
|  |  |
|  | Select caddis.CODDIS, caddis.NUMEDIS, cadpro.NOMEPRO from caddis, cadpro |
|  | where caddis.CODPROF = cadpro.CODPROP order by NOMEPRO; |
|  |  |
|  | -- Sub-Rotinas |
|  |  |
|  | use virtualdc |
|  |  |
|  | create procedure sp\_saudacao as |
|  | Declare @Mensagem char(20) |
|  | Set @Mensagem = 'Alô, Mundo' |
|  | Print @Mensagem |
|  |  |
|  | exec sp\_saudacao; |
|  |  |
|  | Create procedure teste as |
|  | Declare @Mensagem char (25) |
|  | Set @Mensagem = 'Vamos Focar no Estudo' |
|  | Print @Mensagem |
|  |  |
|  | -- Dois paramêtro de entrada e um parâmetro de saida |
|  | CREATE PROCEDURE sp\_calculo @A INT, @B INT, @S INT OUTPUT AS SET @S = @A + @B |
|  | DECLARE @SAIDA INT; |
|  | EXEC sp\_calculo 5, 8, @saida output; |
|  | PRINT @SAIDA; |
|  |  |
|  | -- Recebe dois valores reais com os parâmetros e apresenta o resultado da adição dos valores |
|  | -- caso a soma seja maior ou igual a 10. Caso não seja satisfeita não apresenta nada |
|  |  |
|  | Create Procedure sp\_decisao1 @A FLOAT, @B FLOAT AS |
|  | DECLARE @X FLOAT |
|  | SET @X = @A + @B |
|  | IF (@X >= 10) |
|  | PRINT @X |
|  |  |
|  | EXEC sp\_decisao1 4.4,5.5; |
|  | EXEC sp\_decisao1 5.0, 6.0; |
|  |  |
|  | -- Recebe a entrada de um parâmetro numérico inteiro e exibe uma mensagem informando se o |
|  | -- numero é ou não divisível por 3. |
|  |  |
|  | Create Procedure sp\_decisao2 @valor INT AS |
|  | DECLARE @Resto INTEGER |
|  | DECLARE @Mensagem1 VARCHAR(50) |
|  | DECLARE @Mensagem2 VARCHAR(50) |
|  | SET @Resto = @valor % 3 |
|  | SET @Mensagem1 = 'é um valor divisivel por 3' |
|  | SET @Mensagem2 = 'não é um valor divisivel por 3' |
|  | IF (@Resto = 0) |
|  | BEGIN |
|  | PRINT @Valor |
|  | PRINT @Mensagem1 |
|  | END |
|  | ELSE |
|  | BEGIN |
|  | PRINT @Valor |
|  | PRINT @Mensagem2 |
|  |  |
|  | EXEC sp\_decisao2 4; |
|  | EXEC sp\_decisao2 9; |
|  |  |
|  |  |
|  | Use virtualdc |
|  | -- Mostra o valor Fatorial de um numero inteiro qualquer usando o comando WHILE (LAÇO) |
|  | Create Procedure sp\_fat @valor INT AS |
|  | Declare @Fator INT, @I INT |
|  | SET @Fator = 1 |
|  | SET @I = 1 |
|  | WHILE (@I <= @valor) |
|  | begin |
|  | set @Fator = @Fator \* @I |
|  | set @I = @I + 1 |
|  | END |
|  | PRINT @Fator |
|  |  |
|  | exec sp\_fat 5; |
|  | exec sp\_fat 6; |
|  |  |
|  | SELECT \* FROM morto; |
|  | SELECT \* FROM cadfun; |
|  | -- Insere o registro no arquivo morto e deleta o registro inserido da tabela cadfun |
|  | Create Procedure sp\_demite @codigo int as |
|  | insert into morto |
|  | select codfun, nome, depto, funcao, salario, admissao, filhos, cpf from cadfun |
|  | where CODFUN = @codigo |
|  | delete from cadfun where CODFUN=@codigo |
|  |  |
|  | exec sp\_demite 23; |
|  |  |
|  | use virtualdc; |
|  | -- Gatilhos ou Disparadores |
|  | CREATE TABLE auditoria ( |
|  | USUARIO VARCHAR(40), |
|  | ACAO VARCHAR(9), |
|  | DATA DATE, |
|  | CODIGO INT |
|  | ); |
|  |  |
|  | -- Gatilho para Inserir dados na tabela cadfun |
|  | CREATE TRIGGER tr\_auditor1 ON cadfun FOR INSERT AS |
|  | INSERT INTO auditoria SELECT |
|  | SUSER\_SNAME(), |
|  | 'Cadastrou', |
|  | GETDATE(), |
|  | CODFUN |
|  | FROM inserted; |
|  |  |
|  | -- Vamos inserir um registro. |
|  |  |
|  | INSERT INTO cadfun VALUES (60, 'MARINALVA DA SILVA','3','PROGRAMADOR', 1200.00,'19/10/2016',0, |
|  | '50022255599'); |
|  |  |
|  | SELECT \* FROM cadfun; |
|  | SELECT \* FROM auditoria; |
|  |  |
|  | CREATE TRIGGER tr\_auditor2 ON cadfun FOR UPDATE AS |
|  | INSERT INTO auditoria SELECT |
|  | SUSER\_SNAME(), |
|  | 'Alterou', |
|  | GETDATE(), |
|  | CODFUN |
|  | FROM inserted; |
|  |  |
|  | ---- Fazer uma alteração no Registro do funcionarios antonio |
|  |  |
|  | UPDATE cadfun SET FUNCAO = 'SUPERINTENDENTE' |
|  | WHERE NOME = 'ANTONIO DA SILVA'; |
|  |  |
|  | UPDATE cadfun SET SALARIO = SALARIO \* 1.05; |
|  |  |
|  | SELECT \* FROM auditoria; |
|  | -- Alterar o codigo do funcionario Antonio dos santos |
|  | UPDATE cadfun SET CODFUN = 13 WHERE CODFUN = 12; |
|  |  |
|  |  |
|  | SELECT \* FROM auditoria; |
|  |  |
|  | pausa da apostila de SQL |