



Instrução SELECT

Prof. Ramon Souza

A instrução básica para **recuperar informações de um banco de dados** é a instrução **SELECT**.

É importante ressaltar que esta instrução **não realiza a mesma função que a operação de seleção da álgebra relacional**, mas sim a função da operação de **PROJEÇÃO**.



A sintaxe básica de uma instrução SELECT é da seguinte forma:

```
SELECT coluna1, coluna2, ... FROM nome_da_tabela WHERE condição;
```

É possível realizar uma consulta sem indicar colunas específicas, retornando-se **todas as colunas**, basta usar o *:

```
SELECT * FROM nome_da_tabela WHERE condição;
```

É possível realizar uma instrução SELECT mesmo **sem indicar nenhuma condição** e, portanto, a cláusula WHERE é **opcional**.

```
SELECT coluna1, coluna2, ... FROM nome_da_tabela;
```

SELECT (exemplo)

IDCliente	Nome_Cliente	Nome_Conhecido	Endereco	Cidade	CEP	Pais
1	Alfreds Futterkiste	Maria Anders	Obere Str. 57	Berlin	12209	German y
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

SELECT Nome_Cliente, Cidade FROM Clientes;

Nome_Cliente	Cidade
Alfreds Futterkiste	Berlin
Ana Trujillo Emparedados y helados	México D.F.
Antonio Moreno Taquería	México D.F.

Ao usar **SELECT DISTINCT**, somente serão retornados os **valores diferentes ou distintos**, isto é, não são exibidos valores duplicados.

```
SELECT DISTINCT coluna1, coluna2, ... FROM nome_da_tabela WHERE condição;
```



SELECT DISTINCT (exemplo)

IDCliente	Nome_Cliente	Nome_Conhecido	Endereco	Cidade	CEP	Pais
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

SELECT Pais FROM Clientes;

Pais
Germany
Mexico
Mexico

SELECT DISTINCT Pais FROM Clientes;

Pais
Germany
Mexico

SELECT

- lista_de_atributos
- * retorna todos os atributos.
- **DISTINCT** elimina duplicidades.

FROM

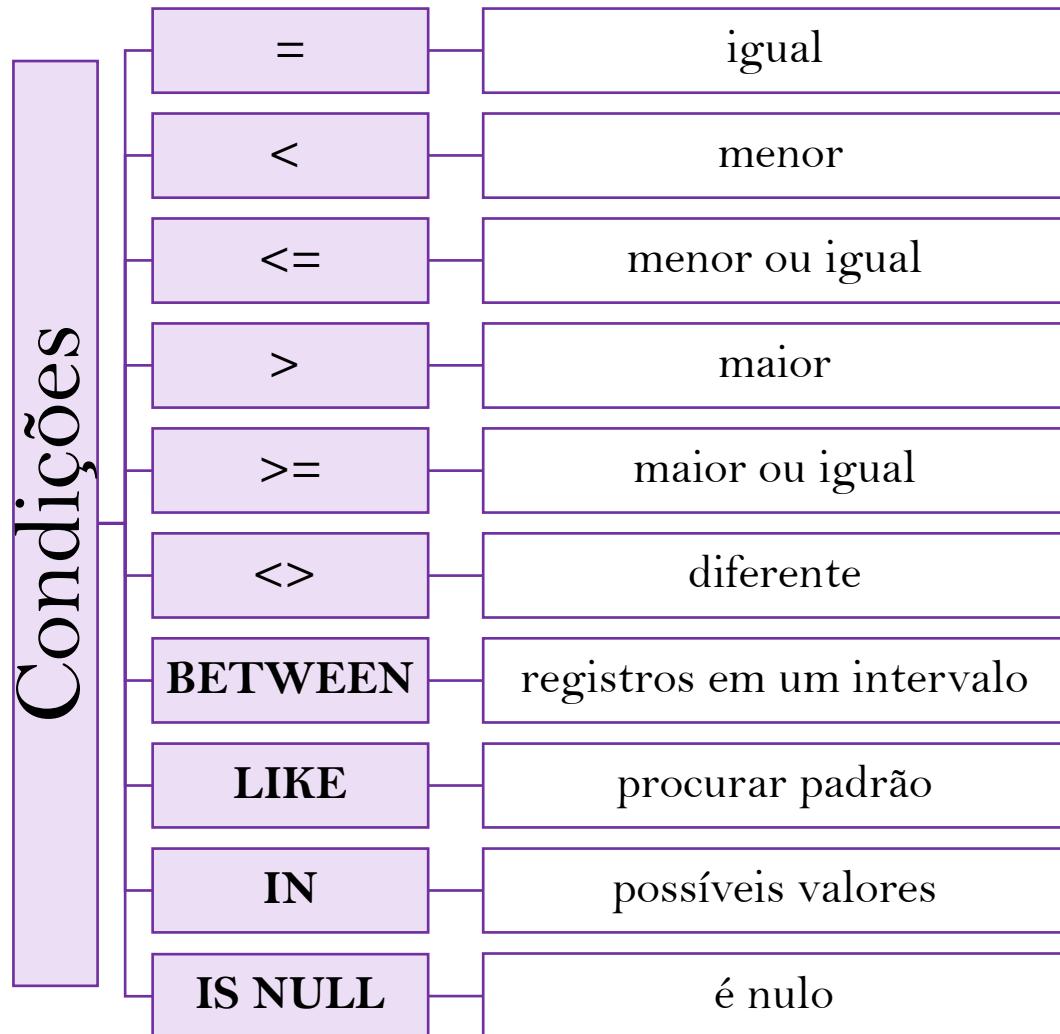
- nome_tabela

WHERE

- condição
- Opcional, mas geralmente usada.

A **condição** é uma expressão condicional (booleana) que **identifica tuplas a serem recuperadas**.





Condições em SELECT (exemplo)

IDCliente	Nome_Cliente	Nome_Conhecido	Endereco	Cidade	CEP	Pais
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	Blondel père et fils	Frédérique Citeaux	24, place Kléber	Strasbourg	67000	France

```
SELECT * FROM Clientes WHERE País='Mexico';
```

IDCliente	Nome_Cliente	Nome_Conhecido	Endereco	Cidade	CEP	Pais
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

O operador **BETWEEN** recupera os **registros que estão em um determinado intervalo**. Os valores podem ser números, texto ou datas e tanto os valores de início como o de fim são incluídos.

```
SELECT coluna1, coluna2, ... FROM nome_da_tabela WHERE coluna BETWEEN valor1 AND valor2;
```

Operador BETWEEN (exemplo)

IDCliente	Nome_Cliente	Nome_Conhecido	Endereco	Cidade	CEP	Pais
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	Blondel père et fils	Frédérique Citeaux	24, place Kléber	Strasbourg	67000	France

SELECT * FROM Clientes WHERE CEP BETWEEN 00000 AND 06000;

IDCliente	Nome_Cliente	Nome_Conhecido	Endereco	Cidade	CEP	Pais
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

O operador **IN** permite especificar **múltiplos valores para uma condição**, isto é, a condição será testada com base na lista de valores indicada

```
SELECT coluna1, coluna2, ... FROM nome_da_tabela WHERE coluna IN (valor1, valor2, ...);
```

Operador IN (exemplo)

IDCliente	Nome_Cliente	Nome_Conhecido	Endereco	Cidade	CEP	Pais
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	Blondel père et fils	Frédérique Citeaux	24, place Kléber	Strasbourg	67000	France

SELECT * FROM Clientes WHERE Cidade IN ('Berlin', 'Strasbourg');

IDCliente	Nome_Cliente	Nome_Conhecido	Endereco	Cidade	CEP	Pais
1	Alfreds Futterkiste	Maria Anders	Obere Str. 57	Berlin	12209	Germany
4	Blondel père et fils	Frédérique Citeaux	24, place Kléber	Strasbourg	67000	France

O operador **LIKE** é utilizado para **procurar um padrão em uma coluna**. Este operador permite a comparação com parte de uma cadeia de caracteres. Este operador é usado em conjunto com dois elementos curinga (wildcard):

- % substitui um número qualquer de 0 ou mais caracteres.
- _ substitui um único caractere.

```
SELECT coluna1, coluna2, ... FROM nome_da_tabela WHERE coluna LIKE padrão;
```

Operador LIKE

Operador LIKE	Procurar padrão em uma coluna
%	Substitui um número qualquer de 0 ou mais caracteres.
_	Substitui um único caractere.
LIKE 'A%	Qualquer string que inicie com A.
LIKE "%A'	Qualquer string que termine com A.
LIKE "%A%"	Qualquer string que tenha A em qualquer posição.
LIKE 'A_'	String de dois caracteres que tenha a primeira letra A e o segundo caractere seja qualquer outro.
LIKE '_A'	String de dois caracteres cujo primeiro caractere seja qualquer um e a última letra seja a letra A.
LIKE '_A_'	String de três caracteres cuja segunda letra seja A, independentemente do primeiro ou do último caractere.
LIKE "%A_"	Qualquer string que tenha a letra A na penúltima posição e a última seja qualquer outro caractere.
LIKE '_A%"	Qualquer string que tenha a letra A na segunda posição e o primeiro caractere seja qualquer outro caractere.
LIKE '___'	Qualquer string com exatamente três caracteres.
LIKE '___%"	Qualquer string com pelo menos três caracteres.
LIKE "%%"	Qualquer string que tenha o caractere "%" em qualquer posição.

Operador LIKE (exemplo)

IDCliente	Nome_Cliente	Nome_Conhecido	Endereco	Cidade	CEP	Pais
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	Blondel père et fils	Frédérique Citeaux	24, place Kléber	Strasbourg	67000	France

```
SELECT * FROM Clientes WHERE Nome_Cliente LIKE 'An%';
```

IDCliente	Nome_Cliente	Nome_Conhecido	Endereco	Cidade	CEP	Pais
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

Operador LIKE (exemplo)

IDCliente	Nome_Cliente	Nome_Conhecido	Endereco	Cidade	CEP	Pais
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	Blondel père et fils	Frédérique Citeaux	24, place Kléber	Strasbourg	67000	France

```
SELECT * FROM Clientes WHERE Nome_Cliente LIKE '%Ana%';
```

IDCliente	Nome_Cliente	Nome_Conhecido	Endereco	Cidade	CEP	Pais
2	Ana Trujillo Emparedados y helados	Ana Trujillo	Avda. de la Constitución 2222	México D.F.	05021	Mexico

Operador LIKE (exemplo)

IDCliente	Nome_Cliente	Nome_Conhecido	Endereco	Cidade	CEP	Pais
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	Blondel père et fils	Frédérique Citeaux	24, place Kléber	Strasbourg	67000	France

```
SELECT * FROM Clientes WHERE Nome_Cliente LIKE '_l%';
```

IDCliente	Nome_Cliente	Nome_Conhecido	Endereco	Cidade	CEP	Pais
1	Alfreds Futterkiste	Maria Anders	Obere Str. 57	Berlin	12209	Germany
4	Blondel père et fils	Frédérique Citeaux	24, place Kléber	Strasbourg	67000	France

O operador **IS NULL** **testa se um valor é NULO**, isto é, se o atributo não possui um valor específico.

```
SELECT coluna1, coluna2, ... FROM nome_da_tabela WHERE coluna IS NULL;
```

Caso queira os não nulos, então basta usar **IS NOT NULL**.

```
SELECT coluna1, coluna2, ... FROM nome_da_tabela WHERE coluna IS NOT NULL;
```

Operador IS NULL(exemplo)

IDCliente	Nome_Cliente	Nome_Conhecido	Endereco	Cidade	CEP	Pais
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	Blondel père et fils	Frédérique Citeaux	24, place Kléber	Strasbourg	67000	France
5	Donald Pateta	null	null	null	null	null

```
SELECT * FROM Clientes WHERE Nome_Contato IS NULL;
```

IDCliente	Nome_Cliente	Nome_Conhecido	Endereco	Cidade	CEP	Pais
5	Donald Pateta	null	null	null	null	null

```
SELECT * FROM Clientes WHERE Nome_Contato IS NOT NULL;
```

IDCliente	Nome_Cliente	Nome_Conhecido	Endereco	Cidade	CEP	Pais
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	Blondel père et fils	Frédérique Citeaux	24, place Kléber	Strasbourg	67000	France

A **comparação com NULL não deve ser feita com os operadores lógicos = ou <>**, mas sim **com IS NULL e IS NOT NULL**. Ao comparar qualquer coisa com NULL usando os operadores lógicos comuns, será retornado um resultado desconhecido na comparação (UNKNOWN) e, por isso, não serão retornadas linhas.

Operador IS NULL(exemplo)

IDCliente	Nome_Cliente	Nome_Conhecido	Endereco	Cidade	CEP	Pais
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	Blondel père et fils	Frédérique Citeaux	24, place Kléber	Strasbourg	67000	France
5	Donald Pateta	null	null	null	null	null

```
SELECT * FROM Clientes WHERE CEP = NULL;
```

IDCliente	Nome_Cliente	Nome_Conhecido	Endereco	Cidade	CEP	Pais

Operador IS NULL(exemplo)

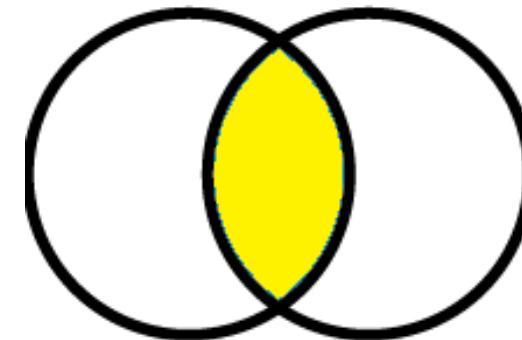
IDCliente	Nome_Cliente	Nome_Conhecido	Endereco	Cidade	CEP	Pais
1	Alfreds Futterkiste	Maria Anders	Obere Str. 57	Berlin	12209	Germany
2	Ana Trujillo Emparedados y helados	Ana Trujillo	Avda. de la Constitución 2222	México D.F.	05021	Mexico
3	Antonio Moreno Taquería	Antonio Moreno	Mataderos 2312	México D.F.	05023	Mexico
4	Blondel père et fils	Frédérique Citeaux	24, place Kléber	Strasbourg	67000	France
5	Donald Pateta	null	null	null	null	null

```
SELECT * FROM Clientes WHERE CEP IS NULL;
```

IDCliente	Nome_Cliente	Nome_Conhecido	Endereco	Cidade	CEP	Pais
5	Donald Pateta	null	null	null	null	null

Exibe os registros em que **todas as condições** são verdadeiras.

```
SELECT coluna1, coluna2, ... FROM nome_da_tabela WHERE condição1 AND  
condição2 AND condição3 ...;
```



Operador AND (exemplo)

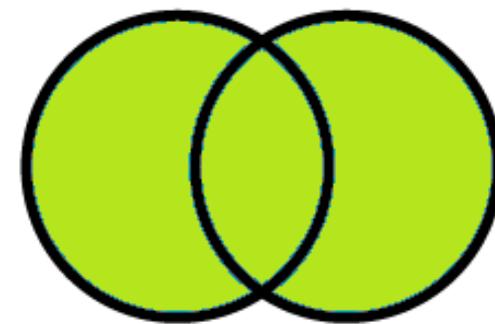
IDCliente	Nome_Cliente	Nome_Conhecido	Endereco	Cidade	CEP	Pais
1	Alfreds Futterkiste	Maria Anders	Obere Str. 57	Berlin	12209	Germany
2	Ana Trujillo Emparedados y helados	Ana Trujillo	Avda. de la Constitución 2222	México D.F.	05021	Mexico
3	Antonio Moreno Taquería	Antonio Moreno	Mataderos 2312	México D.F.	05023	Mexico

```
SELECT * FROM Clientes WHERE Pais='Mexico' AND CEP>=05023;
```

IDCliente	Nome_Cliente	Nome_Conhecido	Endereco	Cidade	CEP	Pais
3	Antonio Moreno Taquería	Antonio Moreno	Mataderos 2312	México D.F.	05023	Mexico

Exibe os registros em que **pelo menos uma condição** é verdadeira.

```
SELECT coluna1, coluna2, ... FROM nome_da_tabela WHERE condição1 OR  
condição2 OR condição3 ...;
```



Operador OR (exemplo)

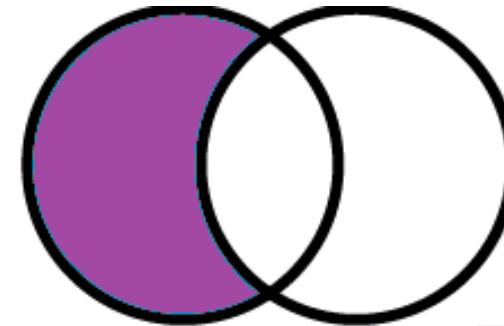
IDCliente	Nome_Cliente	Nome_Conhecido	Endereco	Cidade	CEP	Pais
1	Alfreds Futterkiste	Maria Anders	Obere Str. 57	Berlin	12209	Germany
2	Ana Trujillo Emparedados y helados	Ana Trujillo	Avda. de la Constitución 2222	México D.F.	05021	Mexico
3	Antonio Moreno Taquería	Antonio Moreno	Mataderos 2312	México D.F.	05023	Mexico

```
SELECT * FROM Clientes WHERE Pais='Germany' OR CEP=05021;
```

IDCliente	Nome_Cliente	Nome_Conhecido	Endereco	Cidade	CEP	Pais
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

Exibe os registros que **não satisfazem** uma condição.

```
SELECT coluna1, coluna2, ... FROM nome_da_tabela WHERE NOT condição;
```



Operador NOT (exemplo)

IDCliente	Nome_Cliente	Nome_Conhecido	Endereco	Cidade	CEP	Pais
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

```
SELECT * FROM Clientes WHERE NOT Pais='Mexico';
```

IDCliente	Nome_Cliente	Nome_Conhecido	Endereco	Cidade	CEP	Pais
1	Alfreds Futterkiste	Maria Anders	Obere Str. 57	Berlin	12209	Germany



AND

- Registros em que todas as condições são verdadeiras.
- **SELECT** coluna1, coluna2, ... **FROM** nome_da_tabela **WHERE** condição1 **AND** condição2 **AND** condição3 ...;



OR

- Registros em que pelo menos uma das condições é verdadeira.
- **SELECT** coluna1, coluna2, ... **FROM** nome_da_tabela **WHERE** condição1 **OR** condição2 **OR** condição3 ...;



NOT

- Registros que não satisfazem uma condição.
- **SELECT** coluna1, coluna2, ... **FROM** nome_da_tabela **WHERE NOT** condição;

(CESPE / CEBRASPE - 2021 - TCE-RJ - Analista de Controle Externo - Especialidade: Tecnologia da Informação) Julgue o item a seguir, acerca dos conceitos de administração de banco de dados.

Para selecionar todos os registros da tabela PESSOAS, em que o valor da coluna NOME começa com a letra a, é correto utilizar o comando SQL a seguir.

`SELECT NOME.* FROM PESSOAS WHERE NOME = '%a%'`

**(CESPE / CEBRASPE - 2020 - Ministério da Economia -
Tecnologia da Informação - Ciência de Dados)**

Julgue o item a seguir, a respeito de conceitos de SQL.

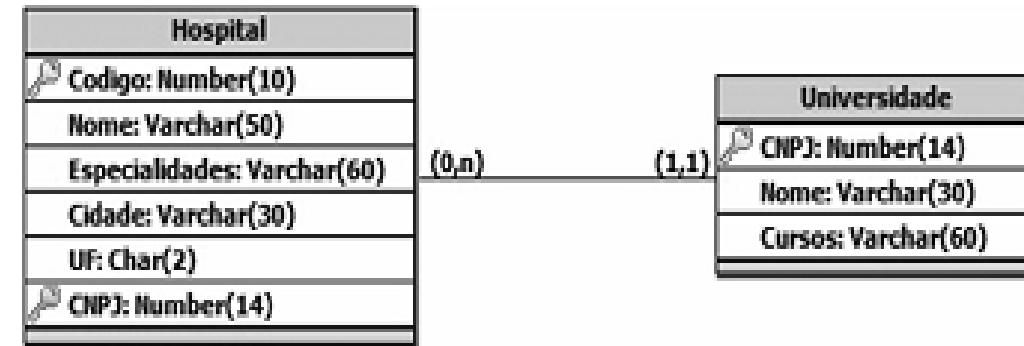
O operador LIKE é usado para pesquisar um padrão especificado em uma coluna da tabela.

(CESPE - 2018 - FUB - Técnico de Tecnologia da Informação)

Julgue o item subsecutivo, a respeito de linguagem de definição e manipulação de dados.

O comando WHERE do SQL corresponde à operação de projeção da álgebra relacional.

(CESPE - 2018 - FUB - Técnico de Tecnologia da Informação)



Tendo como referência o modelo lógico precedente, julgue o item a seguir.

O comando `SELECT nome FROM Hospital WHERE UF='SP';` permite mostrar os nomes dos hospitais localizados na UF (unidade da federação) SP.

(CESPE - 2018 - ABIN - Oficial Técnico de Inteligência - Área 9)

```
SELECT nome
FROM funcionario
WHERE area = 'INTELIGENCIA'
AND endereco LIKE '%BRASILIA,DF%';
```

Tendo como referência o código SQL precedente, julgue o item a seguir.

Em LIKE '%BRASILIA,DF%', o recurso LIKE foi definido de forma incorreta, uma vez que a utilização da vírgula (,), sem a inclusão da palavra-chave ESCAPE, impedirá que o código seja executado.

(CESPE - 2018 - ABIN - Oficial Técnico de Inteligência - Área 9)

```
SELECT nome
FROM funcionario
WHERE area = 'INTELIGENCIA'
AND endereco LIKE '%BRASILIA,DF%';
```

Tendo como referência o código SQL precedente, julgue o item a seguir.

O código em apreço realiza uma consulta que mostra o nome dos funcionários da área de INTELIGENCIA e que têm, como parte do endereço, a cidade de BRASILIA,DF.

(CESPE - 2018 - ABIN - Oficial Técnico de Inteligência - Área 9)

```
SELECT nome
FROM funcionario
WHERE area = 'INTELIGENCIA'
AND endereco LIKE '%BRASILIA,DF%';
```

Tendo como referência o código SQL precedente, julgue o item a seguir.

Na cláusula WHERE, a condição de seleção area = 'INTELIGENCIA' escolhe a tupla de interesse em particular na tabela funcionario, pois area é um atributo de funcionario.

(CESPE - 2018 - ABIN - Oficial Técnico de Inteligência - Área 8)

A respeito de sistemas gerenciadores de banco de dados, julgue o próximo item.

O comando SQL select campo from tabela corresponde a uma operação de projeção da álgebra relacional.

(CESPE - 2016 - FUB - Técnico de Tecnologia da Informação) A respeito das principais instruções da linguagem SQL, julgue o item subsecutivo.

SELECT é uma instrução de controle de banco de dados que permite recuperar o conteúdo de uma ou mais tabelas.

(CESPE - 2016 - FUB - Técnico de Tecnologia da Informação)

A respeito das principais instruções da linguagem SQL, julgue o item subsecutivo.

O operador BETWEEN-AND retornará verdadeiro se o valor da coluna na cláusula WHERE for maior ou igual ao primeiro valor e menor ou igual ao segundo valor.

(CESPE - 2016 - POLÍCIA CIENTÍFICA - PE - Perito Criminal - Ciência da Computação) Na linguagem SQL, quando for necessário obter uma lista e criar uma condição, pode-se utilizar a cláusula

- a) min.
- b) sum.
- c) where.
- d) avg.
- e) max.

(CESPE - 2016 - TRE-PI - Técnico Judiciário - Operação de Computadores)

Assinale a opção correspondente à cláusula SQL que, quando aplicada em uma consulta SELECT, impede a exibição de registros de uma tabela que tenham valores iguais.

- a) DISTINCT
- b) ORDER BY
- c) HAVING
- d) UNIQUE
- e) WHERE

(FCC - 2020 - AL-AP - Analista Legislativo - Desenvolvedor de Banco de Dados) Em um ambiente Oracle, em condições ideais, um Analista precisa criar uma consulta para exibir o nome e o salário dos funcionários cujos números de departamento e salários correspondam a números de departamento e salários de funcionários que ganham comissão. Considerando que os campos usados na instrução existem na tabela colaboradores, a instrução PL/SQL correta que o Analista deve utilizar é:

- a) SELECT nome, salario FROM colaboradores WHERE (salario, idDepartamento) HAVE comissao NOT NULL;
- b) SELECT nome, salario FROM colaboradores WHERE (salario, idDepartamento) = (SELECT salario, idDepartamento FROM colaboradores WHERE comissao IS NOT NULL);
- c) SELECT nome, salario FROM colaboradores WHERE (salario, idDepartamento) IN (SELECT salario, idDepartamento FROM colaboradores WHERE comissao IS NOT NULL);
- d) SELECT nome, salario FROM colaboradores WHERE (comissão!=0 OR comissão IS NOT NULL);
- e) SELECT nome, salario FROM colaboradores WHERE (salario, idDepartamento) IS (SELECT salario, idDepartamento FROM colaboradores WHERE comissao!=0);

(FCC - 2019 - SABESP - Estagiário - Ensino Médio Técnico)

Considere o comando SQL abaixo.

```
CREATE TABLE MANANCIAIS(
    nome_Manancial varchar(50) PRIMARY KEY,
    volume number NOT NULL,
    media_historica number,
    data date
);
```

Considerando que já existem muitos registros na tabela, para selecionar aqueles cujos volumes dos mananciais sejam iguais a 20.0 ou estejam abaixo de 20.0, deve-se utilizar o comando SQL

- a) SELECT ALL RECORDS FROM Mananciais WHERE volume <= 20.0;
- b) SELECT * FROM Mananciais WHERE volume <= 20.0;
- c) SELECT ALL FROM Mananciais WHERE volume <= 20.0;
- d) SELECT *.* FROM Mananciais WHERE volume <= 20.0;
- e) SELECT FROM Mananciais ALL RECORDS WHERE volume <= 20.0;

(FCC - 2019 - METRÔ-SP - Analista Desenvolvimento Gestão Júnior – Ciências da Computação) Considere a expressão SQL, modelo a seguir, em que a numeração não faz parte da expressão, servindo apenas como referência:

I. `SELECT COUNT(column_name)`

II `FROM table_name`

III. `WHERE condition;`

Está correto afirmar que

a) a ordem de escrita deve ser I, III e II.

b) a linha I deve ser alterada para `SELECT COUNT FOR(column_name)`.

c) a ordem de escrita deve ser I, II e III.

d) a linha I deve ser alterada para `SELECT (column_name) COUNT`.

e) a linha I deve ser alterada para `SELECT (column_name) e a linha III para WHERE condition IS COUNT(column_name)`.

(FCC - 2019 - TJ-MA - Técnico Judiciário - Técnico em Informática - Software)

Considere a existência de um banco de dados aberto e em condições ideais, no qual a tabela Processo possui diversos campos, sendo um deles, o campo numero_processo, do tipo cadeia de caracteres (varchar). Para exibir todos os processos cujo número inicie por qualquer caractere seguido de "009.51.01.87348-6", utiliza-se a instrução SQL:

- a) SELECT *.* FROM Processo WHERE numero_processo LIKE '_009.51.01.87348-6';
- b) SELECT * FROM Processo WHERE numero_processo='#009.51.01.87348-6';
- c) SELECT * FROM Processo WHERE numero_processo EQUALS '%009.51.01.87348-6';
- d) SELECT * FROM Processo WHERE numero_processo LIKE '_009.51.01.87348-6';
- e) SELECT *.* FROM Processo WHERE numero_processo LIKE '%009.51.01.87348-6';

(FCC - 2019 - TRF - 4^a REGIÃO - Analista Judiciário - Sistemas de Tecnologia da Informação)

Em uma tabela chamada itemfatura há diversos registros em que constam o mesmo valor no campo idfatura. Para mostrar a quantidade de valores de idfatura diferentes que estão cadastrados na tabela, utiliza-se o comando

- a) SELECT DISTINCT (idfatura) FROM itemfatura;
- b) SELECT * FROM itemfatura WHERE idfatura IS DIFFERENT;
- c) SELECT SUM(DISTINCT idfatura) FROM itemfatura;
- d) SELECT COUNT(DISTINCT idfatura) FROM itemfatura;
- e) SELECT COUNT(DIFFERENT idfatura) FROM itemfatura;

(FCC - 2019 - TRF - 4^a REGIÃO - Técnico Judiciário - Tecnologia da Informação)

Para selecionar na tabela Magistrados todos os registros cadastrados cujo conteúdo do campo cidade inicia-se pela letra 'A', utiliza-se o comando

- a) SELECT * FROM Magistrados WHERE cidade START WITH 'A';
- b) SELECT * FROM Magistrados WHERE cidade LIKE 'A*';
- c) SELECT *.* FROM Magistrados WHERE cidade LIKE 'A_';
- d) SELECT * FROM Magistrados WHERE cidade LIKE 'A%';
- e) SELECT all FROM Magistrados WHERE cidade START WITH 'A';

(FCC - 2019 - TRF - 4^a REGIÃO - Técnico Judiciário - Tecnologia da Informação)

Um Técnico Judiciário necessitou usar a linguagem padrão SQL para recuperar, de uma tabela do banco de dados relacional denominada tabela1,

- I. o menor valor em uma determinada coluna denominada coluna1.
- II. um padrão de valores denominado padrão_desejado em uma outra coluna denominada coluna2.

Para tanto, em duas operações distintas, ele utilizou, respectivamente, as expressões

1.

```
SELECT ....
FROM tabelai
WHERE condição;
```
2.

```
SELECT coluna2
FROM tabelai
WHERE ...;
```

I e II são, correta e respectivamente,

- a) MINVALUE(coluna1) e padrão_desejado %LIKE coluna2
- b) THIN (coluna1) e coluna2 = padrão_desejado
- c) SMALL(coluna1) e padrão_desejado = coluna2
- d) MIN(coluna1) e coluna2 LIKE padrão_desejado
- e) GETSMLL(coluna1) e padrão_desejado % coluna2

(FCC - 2018 - Prefeitura de São Luís - MA - Auditor Fiscal de Tributos I - Tecnologia da Informação (TI))

Um Auditor está executando operações em uma tabela chamada cidadao de um banco de dados aberto e em condições ideais. Para exibir os dados de todas as pessoas que possuem na segunda letra do campo nome a vogal a, deve-se utilizar a instrução SQL:

SELECT * FROM cidadao WHERE

- a) nome = '*a';
- b) nome LIKE '_a%';
- c) nome CONTAINS('a',2);
- d) nome LIKE '*a';
- e) nome HAVE(2,'a');

(FCC - 2017 - TRE-PR - Técnico Judiciário - Programação de Sistemas)

Em um banco de dados aberto e em condições ideais, uma tabela chamada processo possui os registros abaixo.

NumeroProcesso	DataAbertura
1279678-05.2016.1.00.0001	2016-12-26
4598765-03.2017.1.90.0002	2017-01-15
6789764-02.2016.1.80.0003	2016-03-30
9876534-05.2016.1.00.0000	2016-03-20

Ao ser digitado um comando SQL, foi exibido na tela.

NumeroProcesso	DataAbertura
6789764-02.2016.1.80.0003	2016-03-30
9876534-05.2016.1.00.0000	2016-03-20

(FCC - 2017 - TRE-PR - Técnico Judiciário - Programação de Sistemas)

O comando digitado foi:

- a) SELECT * FROM processo WHERE DataAbertura BETWEEN '2016-03-01' AND '2016-03-30';
- b) SELECT NumeroProcesso, DataAbertura FROM processo WHERE DataAbertura BETWEEN ('2016-03-01','2016-03-30');
- c) SELECT * FROM processo WHERE DataAbertura>='2016-03-01' OR DataAbertura<='2016-03-30';
- d) SELECT NumeroProcesso, DataAbertura FROM processo WHERE DataAbertura IN ('2016-03-20' AND '2016-03-30');
- e) SELECT * FROM processo WHERE DataAbertura>'2016-03-20' AND DataAbertura<'2016-03-30';

(FCC - 2017 - DPE-RS - Técnico - Informática)

Solicitou-se a um Técnico selecionar em um banco de dados padrão SQL todos os registros da tabela processo cujo conteúdo do campo cidade_origem não contenha os valores Brasília ou Campinas. Para isso, utilizou a instrução SELECT * FROM processo

- a) WHEN cidade_origem NOT ('Brasília' OR 'Campinas');
- b) WHERE cidade_origem NOT IN ('Brasília','Campinas');
- c) LIKE cidade_origem IS NOT ('Brasília','Campinas');
- d) WHERE cidade_origem NOT IN ('Brasília' OR 'Campinas');
- e) WHERE cidade_origem IS NOT ('Brasília';'Campinas');

(FCC - 2017 - TRF - 5^a REGIÃO - Analista Judiciário - Informática Desenvolvimento)

Para localizar os juízes cujos nomes começam pela letra p e têm pelo menos 3 caracteres de comprimento (incluindo p) utiliza-se a instrução SELECT * FROM juizes WHERE nome

- a) LIKE 'p%_%';
- b) BEGINS WITH 'p' AND LENGTH>=3;
- c) HAS 'p' AND LENGTH>=3;
- d) LIKE 'p%3%*';
- e) HAS'p%3%*'

(FCC - 2017 - TRF - 5^a REGIÃO - Analista Judiciário - Informática Desenvolvimento)

Para localizar os juízes cujos nomes começam pela letra p e têm pelo menos 3 caracteres de comprimento (incluindo p) utiliza-se a instrução SELECT * FROM juizes WHERE nome

- a) LIKE 'p%_%';
- b) BEGINS WITH 'p' AND LENGTH>=3;
- c) HAS 'p' AND LENGTH>=3;
- d) LIKE 'p%3%*';
- e) HAS'p%3%*'

(FGV - 2018 - MPE-AL - Analista do Ministério Público - Administrador de Banco de dados) Considere o comando SQL a seguir.

```
select * from teste where nota <> null
```

Sabendo-se que a tabela teste tem 100 registros, dos quais 15 estão com a coluna nota ainda não preenchida (null), o comando acima, executado no MS SQL Server ou no Oracle, retorna

- a) um erro, porque o literal null não pode ser usado diretamente numa comparação.
- b) zero linhas.
- c) quinze linhas.
- d) oitenta e cinco linhas.
- e) cem linhas.

(FGV - 2017 - IBGE - Analista Censitário - Geoprocessamento) Considere um banco de dados com duas tabelas, R e S, cujas instâncias são exibidas a seguir.

A linguagem mais comum para elaboração de consultas em bancos de dados é a SQL. Ao elaborar uma consulta nessa linguagem, emprega-se a cláusula WHERE quando se deseja:

- a) especificar a tabela onde será realizada a consulta;
- b) especificar o diretório onde os dados estão armazenados;
- c) especificar o endereço IP onde os dados estão armazenados;
- d) especificar condições a que as instâncias selecionadas devem atender;
- e) extrair a geometria do objeto selecionado na consulta.

(FGV - 2017 - SEPOG - RO - Analista em Tecnologia da Informação e Comunicação) Observe as tabelas a seguir:

Tabela: animais	
familia	nome
mamifero	cachorro
mamifero	leao
peixe	linguado
passaro	aquia
reptil	cobra
passaro	pelicano
peixe	linguado

Tabela: tipos	
familia	tipo_sangue
mamifero	quente
peixe	frio
passaro	quente

Assinale a opção que indica o número de linhas retornadas pela consulta SQL a seguir.

SELECT DISTINCT nome FROM animais a, tipos t WHERE a.familia = t.família

- a) 0
- b) 1
- c) 3
- d) 4
- e) 5



Instrução SELECT

Prof. Ramon Souza