



SQL (QUESTÕES)

Prof. Ramon Souza

(CESPE / CEBRASPE - 2021 - SEED-PR - Professor - Informação e Comunicação) Por meio da linguagem de definição de dados (DDL), são definidas estruturas do banco de dados; por sua vez, uma linguagem de manipulação de dados (DML) é empregada para especificar as recuperações e atualizações do banco de dados. Assinale a opção que apresenta, respectivamente, um comando DDL e um comando DML.

- a) Update e Drop
- b) Alter e Update
- c) Select e Delete
- d) Drop e Alter
- e) Insert e Create

(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.

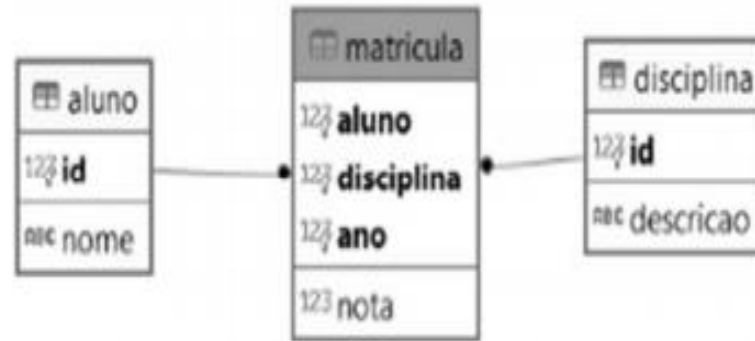
Para selecionar todos os registros da tabela PESSOAS, em que o valor da coluna NOME começa com a letra a, é correto utilizar o comando SQL a seguir.

```
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 / CEBRASPE - 2020 - Ministério da Economia - Tecnologia da Informação - Desenvolvimento de Software)



Tendo como referência o diagrama de entidade relacionamento precedente, julgue o próximo item, a respeito de linguagem de definição de dados e SQL.

A seguinte expressão SQL permite alterar a nota de geografia do aluno de nome Beltrano para 9.5.

```
update matricula set nota=9.5 where aluno ='Beltrano' and disciplina ='Geografia'
```

(CESPE / CEBRASPE - 2020 - Ministério da Economia - Tecnologia da Informação - Desenvolvimento de Software) Acerca de sistemas gerenciadores de banco de dados, julgue o item subsequente.

Em SQL, o comando INSERT INTO é usado para inserir novos dados em um banco de dados MySQL.

(CESPE / CEBRASPE - 2019 – TJ-AM) Julgue o próximo item, relativos a sistema gerenciador de banco de dados (SGBD).

Um SGBD trata do acesso ao banco e pode ser executado independentemente pelo Oracle, MySQL ou PostgreSQL; no entanto, cada SGBD utiliza DML (data manipulation language) e DDL (data definition language) específicas.

(CESPE - 2019 - TJ-AM - Assistente Judiciário - Programador)

```
SELECT P.P#,  
       P.COR,  
       MAX (FP.QDE) AS QDEMAX  
FROM P, FP  
WHERE P.P# = FP.P#  
AND (P.COR = COR ('Azul') OR P.COR = COR ('Rosa'))  
AND FP.QDE > QDE (100)  
GROUP BY P.P#, P.COR  
HAVING SUM (FP.QDE) > QDE (250)
```

Considerando a formulação do algoritmo conceitual da consulta em SQL precedente, julgue o item a seguir.

O resultado da cláusula WHERE é agrupado pelos valores das colunas nomeadas na cláusula GROUP BY P.P#, P.COR. Além disso, o uso da cláusula HAVING elimina os grupos que não satisfazem à expressão booleana $\text{SUM (FP.QDE) > QDE (250)}$.

(CESPE - 2019 - TJ-AM - Assistente Judiciário - Programador)

```
SELECT P.P#,  
       P.COR,  
       MAX (FP.QDE) AS QDEMAX  
FROM P, FP  
WHERE P.P# = FP.P#  
AND (P.COR = COR ('Azul') OR P.COR = COR ('Rosa'))  
AND FP.QDE > QDE (100)  
GROUP BY P.P#, P.COR  
HAVING SUM (FP.QDE) > QDE (250)
```

Considerando a formulação do algoritmo conceitual da consulta em SQL precedente, julgue o item a seguir.

O uso da cláusula WHERE sobre as linhas que não satisfazem à seguinte expressão booleana

P.P# = FP.P#

AND (P.COR = COR ('Azul') OR P.COR = COR ('Rosa'))

AND FP.QDE > QDE (100)

reduz o resultado da cláusula FROM.

(CESPE - 2019 - TJ-AM - Assistente Judiciário - Programador)

```
SELECT P.P#,  
       P.COR,  
       MAX (FP.QDE) AS QDEMAX  
FROM P, FP  
WHERE P.P# = FP.P#  
AND (P.COR = COR ('Azul') OR P.COR = COR ('Rosa'))  
AND FP.QDE > QDE (100)  
GROUP BY P.P#, P.COR  
HAVING SUM (FP.QDE) > QDE (250)
```

Considerando a formulação do algoritmo conceitual da consulta em SQL precedente, julgue o item a seguir.

A cláusula FROM é avaliada para produzir uma nova tabela, e essa nova tabela não é produzida a partir do produto cartesiano das tabelas P e FP.

(CESPE / CEBRASPE - 2019 - TJ-AM - Analista Judiciário - Analista de Sistemas) A respeito de bancos de dados relacionais, julgue o item a seguir.

Em SQL, o comando `RIGHT OUTER JOIN` exhibe a união entre duas tabelas, apresentando as linhas da segunda tabela que também existem na primeira tabela, descartando-se as demais situações.

(CESPE - 2019 - MPC-PA - Assistente Ministerial de Informática) Considere que, no sistema modelado no diagrama a que se refere o texto 9A1-I, as tarifas de uma companhia aérea para os trechos de voos promocionais sejam de até R\$ 100. Considere, ainda, a aplicação do seguinte comando SQL nesse contexto.

```
select codAeronave, avg(TarifaDoTrecho)
  from TrechoDoVoo
 where TarifaDoTrecho >100
 group by CodAeronave
 having avg(TarifaDoTrecho) >250 ;
```

Nesse caso, o comando SQL em questão mostra apenas as médias de tarifas

- a) maiores que R\$ 250 de todas as aeronaves, de forma agrupada, sem os voos promocionais.
- b) maiores que R\$ 250 de todas as aeronaves, de forma agrupada, com os voos promocionais.
- c) maiores que R\$ 250 de todas as aeronaves, por trecho de voo, sem os voos promocionais.
- d) por aeronave, maiores que R\$ 250 nos trechos em que a companhia aérea participa, com os voos promocionais.
- e) por aeronave, maiores que R\$ 250 nos trechos em que a companhia aérea participa, sem os voos promocionais.

(CESPE - 2018 - Polícia Federal - Perito Criminal Federal - Área 3)



As informações de um departamento e de seus colaboradores devem ser organizadas e armazenadas conforme a estrutura mostrada no diagrama precedente. Para isso, serão utilizados os comandos DDL a seguir.

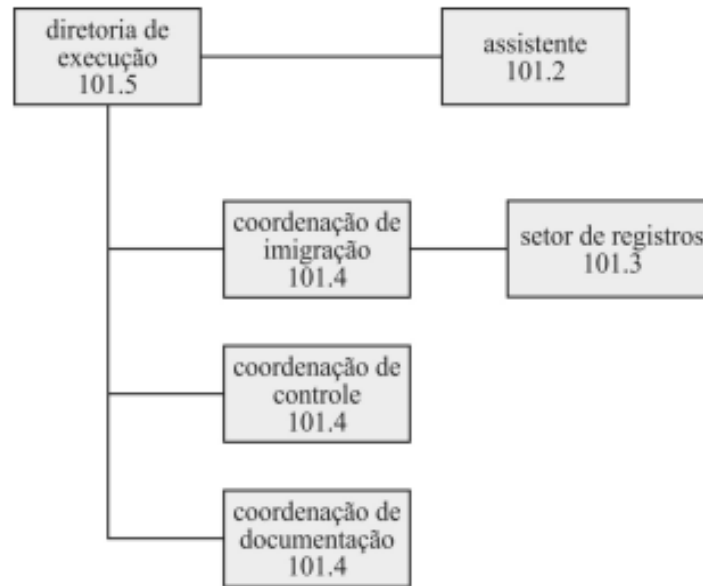
```
create table organograma (  
    id integer primary key,  
    descricao varchar(50),  
    cargo varchar(50),  
    pai integer,  
    constraint fk_organograma foreign key (pai)  
references organograma  
);  
  
create table colaborador (  
    cpf bigint primary key,  
    nome varchar(50),  
    data_nascimento date  
);  
  
create table colaborador_organograma  
(  
    cpf bigint,  
    cargo integer,  
    data_nomeacao date,  
    data_exoneracao date,  
    constraint pk_colaborador_organograma primary  
key (cpf, cargo),  
    constraint fk_colaborador_organograma_cpf  
foreign key (cpf) references colaborador,  
    constraint fk_colaborador_organograma_cargo  
foreign key (cargo) references organograma (id)  
);
```

Tendo como referência as informações apresentadas, julgue o próximo item.

O comando SQL a seguir permite apagar o conteúdo da tabela colaborador_organograma.

delete from colaborador_organograma;

(CESPE - 2018 - Polícia Federal - Perito Criminal Federal - Área 3)



As informações de um departamento e de seus colaboradores devem ser organizadas e armazenadas conforme a estrutura mostrada no diagrama precedente. Para isso, serão utilizados os comandos DDL a seguir.

```
create table organograma (  
    id integer primary key,  
    descricao varchar(50),  
    cargo varchar(50),  
    pai integer,  
    constraint fk_organograma foreign key (pai)  
references organograma  
);  
  
create table colaborador (  
    cpf bigint primary key,  
    nome varchar(50),  
    data_nascimento date  
);  
  
create table colaborador_organograma  
(  
    cpf bigint,  
    cargo integer,  
    data_nomeacao date,  
    data_exoneracao date,  
    constraint pk_colaborador_organograma primary  
key (cpf, cargo),  
    constraint fk_colaborador_organograma_cpf  
foreign key (cpf) references colaborador,  
    constraint fk_colaborador_organograma_cargo  
foreign key (cargo) references organograma (id)  
);
```

Tendo como referência as informações apresentadas, julgue o próximo item.

A seguir, são apresentadas as expressões SQL corretas para inserir na tabela organograma as informações constantes do diagrama apresentado.


```
insert into organograma (id, descricao, cargo,  
pai) values (1, 'assistente', '101.2', 2);
```

```
insert into organograma (id, descricao, cargo,  
pai) values (2, 'coordenação de imigração',  
'101.4', 1);
```

```
insert into organograma (id, descricao, cargo,  
pai) values (3, 'coordenação de controle',  
'101.4', 1);
```

```
insert into organograma (id, descricao, cargo,  
pai) values (4, 'coordenação de documentação',  
'101.4', 1);
```

```
insert into organograma (id, descricao, cargo,  
pai) values (5, 'setor de registros', '101.3', 3);
```

```
insert into organograma (id, descricao, cargo)  
values (6, 'diretor de execução', '101.5');
```

(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 - TCE-MG - Analista de Controle Externo - Ciência da Computação) Julgue os próximos itens, no que diz respeito a comandos SQL de consulta.

I Subqueries de múltiplas colunas retornam várias colunas em uma mesma linha.

II A função UNION apresenta as linhas que existam simultaneamente em duas ou mais tabelas.

III A função EXISTS garante que o resultado de uma subquery somente seja mostrado se retornar uma ou mais linhas.

IV A função EXTRACT retorna uma das informações de um campo do tipo data (dia, mês, ano, hora, minuto ou segundo).

Assinale a opção correta.

- a) Estão certos apenas os itens I e II.
- b) Estão certos apenas os itens I e IV.
- c) Estão certos apenas os itens II e III.
- d) Estão certos apenas os itens III e IV.
- e) Todos os itens estão certos.



SQL (QUESTÕES)

Prof. Ramon Souza