



SQL (QUESTÕES)

Prof. Ramon Souza

(FCC - 2020 - AL-AP - Analista Legislativo - Desenvolvedor de Sistemas) Um auditor está checando práticas de aplicação de desconto comercial para clientes, executando consultas em SQL (Structured Query Language) nos bancos de dados digitais de uma empresa, para validar se os cálculos e regras de autorização estabelecidas pelo negócio estão sendo cumpridos. É correto afirmar que o uso de SQL em auditoria corresponde à aplicação de

- a) uma linguagem de programação orientada a objetos que funciona integrada ao sistema gerenciador de bancos de dados e permite simular operações.
- b) um software de auditoria especializado que permite, entre outras coisas, a simulação das funções do software auditado e seu banco de dados.
- c) uma linguagem de programação declarativa que funciona integrada ao sistema gerenciador de bancos de dados e permite consultar dados, entre outras operações.
- d) um sistema de segurança computacional que permite coletar dados, analisá-los e realizar medidas protetivas contra práticas incorretas de operação.
- e) um software de aplicação de usuário final, operado com a finalidade de realizar registros de atividades transacionais do dia a dia da empresa.

Questão de Concurso

(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
);
```

Um Estagiário utilizou outro comando SQL, sem erros de sintaxe e com valores válidos, para inserir dados na tabela. O comando correto por ele utilizado foi:

- a) INSERT INTO MANANCIAIS VALUES ('Cantareira', 58.0, , '12-05-2019');
- b) INSERT INTO MANANCIAIS ('Cantareira', 58.0, 78.6, '12-05-2019');
- c) INSERT IN MANANCIAIS VALUES ('Cantareira', 58.0, 78.6, 12-05-2019);
- d) INSERT INTO MANANCIAIS VALUES ('Cantareira', , '78.6', '12-05-2019');
- e) INSERT IN MANANCIAIS ('Cantareira', '58.0', '78.6', '12-05-2019');

(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 - TRF - 3^a REGIÃO - Técnico Judiciário - Informática) Em uma instrução SQL, a cláusula GROUP BY, se utilizada juntamente com funções de grupo, deve estar em uma posição específica como

a) SELECT coluna1 [, coluna2, ...] . função_de_grupo, [função_de_grupo, ...]

FROM tabela

ORDER BY coluna2

GROUP BY coluna4 [, coluna5, ...]

HAVING coluna5 > 10

WHERE condição;

b) SELECT coluna1 [, coluna2, ...] . função_de_grupo, [função_de_grupo, ...]
FROM tabela
WHERE condição
ORDER BY coluna2
GROUP BY coluna4 [, coluna5, ...]
HAVING coluna5 > 10;

c) SELECT coluna1 [, coluna2, ...] . função_de_grupo, [função_de_grupo, ...]
FROM tabela
WHERE condição
GROUP BY coluna4 [, coluna5, ...]
HAVING coluna5 > 10
ORDER BY coluna2;

d) SELECT coluna1 [, coluna2, ...] . função_de_grupo, [função_de_grupo, ...]
FROM tabela
HAVING coluna5 > 10
ORDER BY coluna2
WHERE condição
GROUP BY coluna4 [, coluna5, ...] ;

e) SELECT coluna1 [, coluna2, ...] . função_de_grupo, [função_de_grupo, ...]
FROM tabela
HAVING coluna5 > 10
WHERE condição
ORDER BY coluna2
GROUP BY coluna4 [, coluna5, ...];

(FCC - 2019 - TRF - 3^a REGIÃO - Técnico Judiciário - Informática) Os Sistemas Gerenciadores de Bancos de Dados - SGBDs, antes da virada do século, apresentavam duas linguagens para sua administração e utilização: DDL e DML. Na atualidade, conceitualmente, os autores as distribuem em cinco categorias. O comando GRANT, por exemplo, é categorizado como

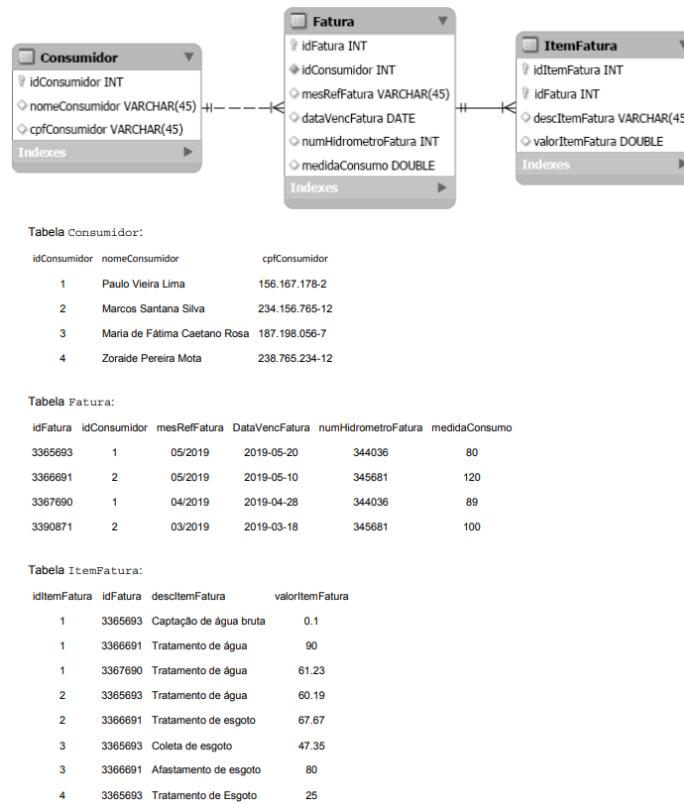
- a) DML - Data Manipulation Language.
- b) DTL - Data Transaction Language.
- c) DQL - Data Query Language.
- d) DDL - Data Definition Language.
- e) DCL - Data Control Language.

(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 - SANASA Campinas - Analista de Tecnologia da Informação - Suporte de DBA-Banco de Dados) Atenção: Para responder à questão, considere os dados abaixo.



Considere que não há nenhum registro cadastrado além dos mostrados nas tabelas acima.

Um Analista de Suporte de Banco de Dados digitou um comando SQL que exibiu o número de faturas que cada consumidor possuía cadastradas no banco de dados, conforme mostrado abaixo.

nomeConsumidor	numeroFaturas
Paulo Vieira Lima	2
Marcos Santana Silva	2
Maria de Fátima Caetano Rosa	0
Zoraide Pereira Mota	0

O comando correto utilizado para exibir os dados foi

- a) SELECT nomeConsumidor FROM Consumidor c JOIN Fatura f (COUNT(f.idFatura)) AS numeroFaturas ON (f.idConsumidor = c.idConsumidor);
- b) SELECT nomeConsumidor, (SELECT COUNT(idFatura) FROM Fatura WHERE idConsumidor = c.idConsumidor) FROM Consumidor c;
- c) SELECT nomeConsumidor FROM Consumidor c JOIN Fatura f ON (SELECT COUNT(f.idFatura) FROM Fatura f WHERE f.idConsumidor = c.idConsumidor) AS numeroFaturas;
- d) SELECT nomeConsumidor, (SELECT f.idFatura FROM Fatura f WHERE f.idConsumidor = c.idConsumidor) as numeroFaturas FROM Consumidor c;
- e) SELECT nomeConsumidor, (SELECT COUNT(f.idFatura) FROM Fatura f WHERE f.idConsumidor = c.idConsumidor) as numeroFaturas FROM Consumidor c;

(FCC - 2019 - SANASA Campinas - Analista de Tecnologia da Informação - Suporte de DBA-Banco de Dados) Atenção: Para responder à questão, considere os dados abaixo.

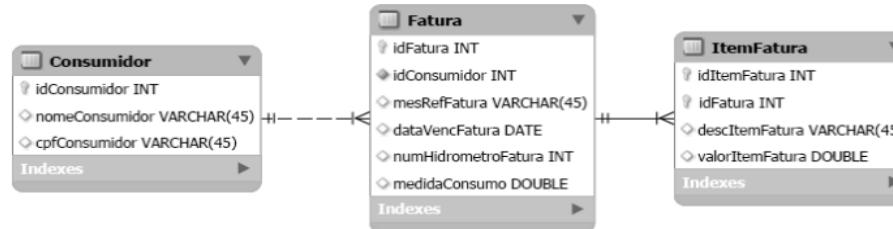


Tabela Consumidor:

	idConsumidor	nomeConsumidor	cpfConsumidor
1	Paulo Vieira Lima	156.167.178-2	
2	Marcos Santana Silva	234.156.765-12	
3	Maria de Fátima Caetano Rosa	187.198.056-7	
4	Zoraide Pereira Mota	238.765.234-12	

Tabela Fatura:

	idFatura	idConsumidor	mesRefFatura	DataVencFatura	numHidrometroFatura	medidaConsumo
3365693	1	05/2019	2019-05-20	344036	80	
3366691	2	05/2019	2019-05-10	345681	120	
3367690	1	04/2019	2019-04-28	344036	89	
3390871	2	03/2019	2019-03-18	345681	100	

Tabela ItemFatura:

	idItemFatura	idFatura	descItemFatura	valorItemFatura
1	3365693	Captação de água bruta	0.1	
1	3366691	Tratamento de água	90	
1	3367690	Tratamento de água	61.23	
2	3365693	Tratamento de água	60.19	
2	3366691	Tratamento de esgoto	67.67	
3	3365693	Coleta de esgoto	47.35	
3	3366691	Afastamento de esgoto	80	
4	3365693	Tratamento de Esgoto	25	

Considere que não há nenhum registro cadastrado além dos mostrados nas tabelas acima.

Para excluir as faturas do consumidor 2 utiliza-se o comando

- a) `DROP FROM Fatura WHERE idConsumidor=2;`
- b) `DELETE FROM Fatura WHERE idConsumidor=2 ON DELETE CASCADE;`
- c) `DROP RECORD WHERE idConsumidor=2 FROM Fatura;`
- d) `DELETE ALL idConsumidor=2 FROM Fatura;`
- e) `DELETE FROM Fatura WHERE idConsumidor=2;`



SQL (QUESTÕES)

Prof. Ramon Souza