Nome:\_\_\_\_\_\_\_Elton Correia\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Serie:\_1° Desenvolvimento de sistemas – turma A\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**EXERCÍCIOS DE FIXAÇÃO EM SQL**

1) Os comandos **DTL** são responsáveis por gerenciar diferentes transações ocorridas dentro de um Banco de dados. Ele é dividido em três comandos, quais seriam, justifique sua resposta:

**a)BEGIN TRAN, COMMIT e ROLLBACK.**

– Esses são os comandos de transação em um banco, sendo BEGIN TRAN o comando que inicia a transação de dados, COMMIT o comando que confirma a transação e ROLLBACK o responsável por desfazer essa transação.

b)BEGIN DEVTRAN, DEFINE e BACK.

c)BEGIN LIBRARY, FIND e ROLLBACK.

d)BEGIN, DEFINE LIBRARY e SCROLL.

e)TRAN, FIND e FINISH.

2) Em uma tabela chamada Contribuinte de um banco de dados padrão SQL aberto e em condições ideais há o campo idContribuinte do tipo inteiro e chave primária. Há também o campo nomeContribuinte que é do tipo varchar. Nessa tabela, um Auditor Fiscal deseja alterar o nome do contribuinte de id 1 para 'Marcos Silva'. Para isso, terá que utilizar o comando, justifique sua resposta:

**a)ALTER TABLE Contribuinte SET nomeContribuinte='Marcos Silva' WHERE idContribuinte=1;**

- ALTER TABLE é um comando utilizado para alterar, O SET é utilizado para definir (WHERE) onde está o ‘Marcos Silva’.

b)UPDATE Contribuinte SET nomeContribuinte='Marcos Silva' WHERE idContribuinte=1;

c)UPDATE nomeContribuinte TO 'Marcos Silva' FROM Contribuinte WHERE idContribuinte=1;

d)ALTER TABLE Contribuinte FIELD nomeContribuinte='Marcos Silva' WHERE idContribuinte=1;

e)UPDATE TABLE Contribuinte FIELD nomeContribuinte='Marcos Silva' WHERE idContribuinte=1.

3) A expressão em SQL padrão descrita no quadro abaixo realiza uma consulta em um BANCO DE DADOS que contém, pelo menos, duas tabelas, sendo uma denominada FUNCIONARIO e outra denominada DEPENDENTE, que armazenam informações sobre funcionários de uma empresa e seus dependentes, respectivamente.

A tabela FUNCIONARIO possui pelo menos três atributos: nome\_func, sobrenome\_func e cpf\_func, que armazenam, respectivamente, o nome, o sobrenome e o número de CPF dos funcionários.

A tabela DEPENDENTE possui pelo menos quatro atributos: nome\_depen, sobrenome\_depen e cpf\_depen, que armazenam, respectivamente, o nome, o sobrenome e o número de CPF dos dependentes, e cpf\_func\_resp, que armazena o número do CPF do funcionário responsável pelos dependentes.

SELECT nome\_func, sobrenome\_func  
 FROM FUNCIONARIO  
 WHERE NOT EXISTS (   
SELECT \* FROM DEPENDENTE  
 WHERE cpf\_func = cpf\_func\_resp  
);

Com base nas informações apresentadas, a consulta em SQL padrão acima descrita recupera, justifique sua resposta: :

**~~a)~~o nome e o sobrenome de funcionários que não têm dependentes. –** O comando SELECT irá trazer os valores de nome e sobrenome do funcionário, entretanto o comando NOT EXISTS vai retornar cpf de funcionários que não tem dependente .

b)o nome e o sobrenome de todos os dependentes dos funcionários.

c)o nome e o sobrenome de funcionários que têm dependentes.

d)todos os atributos de funcionários que não têm dependentes.

e)todos os atributos de todos os dependentes dos funcionários.

4) 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, justifique sua resposta:

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.**

- EXISTS retorna com a linha em um valor verdadeiro; EXTRACT extrai campos de data/hora.

e)Todos os itens estão certos.

5) Para calcular o número de produtos com nomes que terminam com a letra B, o Auditor testou os comandos abaixo.

I. SELECT COUNT(nomePro)  
FROM Produto  
WHERE nomePro BEGIN '%B';

II. SELECT COUNT(\*)  
FROM PRODUTO  
WHERE SUBSTR(nomePro, -1)='B';

III. SELECT COUNT(\*)  
FROM Produto  
WHERE nomePro LIKE '%B';

IV. SELECT COUNT(nomePro)  
FROM Produto  
WHERE SUBSTR(nomePro, 0)='B';

Mostrará o resultado desejado o que consta APENAS em, justifique sua resposta:

**a)III** – O comando LIKE busca e compara os dados com a letra especificada.

b)I e IV

c)III e IV

d)I e II

e)II e III

6) Na linguagem SQL, considerando duas tabelas A e B, para se obter o que está na Tabela A e não está na Tabela B, assim como o que está na Tabela B e não está na Tabela A é necessário usar o comando, justifique sua resposta:

**a)outer excluding join.**

– Essa função quando incluída em um comando é utilizado para excluir a intersecção entre o conjunto de duas tabelas ou seja as informações semelhantes.

b)right excluding join.

c)left excluding join.

d)middle excluding join.

7) Considere a sintaxe SQL básica do comando para a criação de gatilhos:

CREATE TRIGGER ON ...

Nesse comando, as opções corretas do

**a)evento disparador são: Insert, Delete e Update. - correta**

b)evento disparador são: Select, Group By e Inner Join.

c)evento disparador são: Commit, Rollback e Savepoint.

d)instante de execução são: First e Last.

e)instante de execução são: One e All.

8) Usando a linguagem SQL, um fiscal escreveu corretamente, em uma consulta, a expressão WHERE Nome-Contribuinte LIKE 'p%o'. Com isso ele obteve nomes de contribuintes como, por exemplo:

**a)Paulo ou Pedro; - resultados possíveis**

b)Augusto e Jordão;

c)Paulo ou Paula;

d)Álvaro ou Augusto;

e)Paula ou Patricio

9) Considere o código SQL abaixo, que gerou a tabela ItemFatura.

CREATE TABLE ItemFatura(  
idItemFatura INT NOT NULL,  
idFatura INT NOT NULL,  
descItemFatura VARCHAR(45),  
valorItemFatura DOUBLE,  
**..I..**  
);

Considerando que a tabela ItemFatura possui chave primária composta pelos campos idItemFatura e idFatura, e que se uma fatura for excluída, automaticamente serão excluídos todos os seus itens, a lacuna I deve ser preenchida corretamente por:

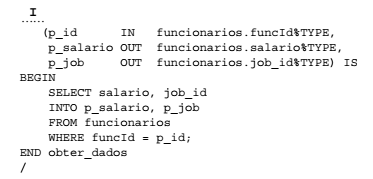
a)PRIMARY KEY (idItemFatura, FOREIGN KEY(idFatura)) REFERENCES Fatura(idFatura) ON DELETE CASCADE;

b)PRIMARY KEY (idItemFatura, idFatura), FOREIGN KEY(idFatura) FROM Fatura(idFatura) WITH DELETE CASCADE;

c)PRIMARY KEY (idItemFatura, idFatura), FOREIGN KEY (idFatura) REFERENCES Fatura(idFatura) ON DELETE CASCADE;

**d)PRIMARY KEY (idItemFatura, idFatura), FOREIGN KEY(idFatura) REFERENCES Fatura(idFatura); – alternativa correta**

e)PRIMARY KEY (idItemFatura, idFatura), FOREIGN KEY(idFatura) FROM Fatura(idFatura) ON DELETE CASCADE.

10) Um programador deseja definir uma query SQL para retornar os valores das colunas salario e job\_id para determinado id de funcionário, de forma que ela seja executada de acordo com os parâmetros que receber. Considerando a existência da tabela e dos campos indicados, no Oracle PL/SQL digitou o bloco de comandos abaixo. Completa adequadamente a lacuna I a instrução: CREATE 

a)FUNCTION obter\_dados PARAMETERS;

b)OR REPLACE VIEW obter\_dados; x

**c)OR REPLACE PROCEDURE obter\_dados**

d)TRIGGER obter\_dados **E**STRUCTURE obter\_dados.