

# LINGUAGEM PARA BANCOS DE DADOS RELACIONAIS - SQL

## Atividade 1

Pergunta: Os comandos INSERT, UPDATE e DELETE fazem parte de qual dos seguintes subconjuntos da linguagem SQL?

Alternativas:

- DDL (Data Definition Language)
- DTL (Data Transaction Language)
- DQL (Data Query Language)
- DCL (Data Control Language)
- DML (Data Manipulation Language) (Correta)

Resumo: Os comandos INSERT, UPDATE e DELETE são utilizados para manipular dados dentro das tabelas, por isso pertencem ao subconjunto DML (Data Manipulation Language).

Pergunta: Que comando do SQL Plus deve ser utilizado para exibir a estrutura de qualquer tabela?

Alternativas:

- GET
- CONNECT
- SPOOL
- LIST
- DESCRIBE (Correta)

Resumo: O comando DESCRIBE exibe a estrutura de uma tabela, mostrando colunas, tipos de dados e restrições, sendo a opção correta para essa finalidade.

Pergunta: Deseja-se armazenar em uma tabela denominada PEDIDO o valor unitário de produtos que variam de 0.01 a 999.99. Qual das seguintes alternativas deveria ser utilizada no Oracle?

Alternativas:

- NUMBER (3,2)
- INT (5,2)
- NUMBER (5,2) (Correta)
- NUMBER (6)
- INT (3,2)

Resumo: O tipo NUMBER(5,2) permite armazenar valores com até 5 dígitos no total, sendo 2 após a vírgula. Isso cobre o intervalo de 0.01 até 999.99, atendendo ao requisito.

Pergunta: Qual dos seguintes comandos deve ser utilizado para exibir os nomes das regras (restrições) de uma tabela denominada PRODUTO?

Alternativas:

- SELECT CONSTRAINT\_NAME FROM USER\_TABLES WHERE TABLE\_NAME = 'PRODUTO';

- SELECT USER\_NAME FROM CONSTRAINTS WHERE TABLE\_NAME = 'PRODUTO';
- SELECT NAME FROM USER\_CONSTRAINTS WHERE TABLE\_NAME = 'PRODUTO';
- SELECT CONSTRAINT\_NAME FROM USER\_CONSTRAINTS WHERE TABLE\_NAME = 'PRODUTO'; (Correta)
- SELECT CONSTRAINT FROM USER\_CONSTRAINTS WHERE TABLE\_NAME = 'PRODUTO';

Resumo: As restricoes de uma tabela estao armazenadas na visao USER\_CONSTRAINTS. O comando correto para listar os nomes das restricoes e SELECT CONSTRAINT\_NAME FROM USER\_CONSTRAINTS WHERE TABLE\_NAME = 'PRODUTO'.

## Atividade 2

Pergunta: Deseja-se eliminar a tabela CLIENTE. Porém, outra tabela, denominada PEDIDO, contém uma chave estrangeira relacionando o CODIGO\_CLIENTE com a chave primaria na tabela CLIENTE. Qual dos seguintes comandos deverá ser utilizado para eliminar a tabela CLIENTE, conforme as condições apresentadas?

Alternativas:

- DELETE CLIENTE CASCADE CONSTRAINTS;
- DROP TABLE CLIENTE CASCADE CONSTRAINTS; (Correta)
- DROP CLIENTE AND CONSTRAINTS;
- DROP TABLE CLIENTE;
- DROP TABLE CLIENTE WITH CONSTRAINTS;

Resumo: O comando correto é DROP TABLE CLIENTE CASCADE CONSTRAINTS; pois permite excluir a tabela CLIENTE mesmo havendo chaves estrangeiras em outras tabelas, removendo também as restrições associadas.

Pergunta: É necessário adicionar em uma tabela criada no SGDB Oracle e denominada ALUNO a coluna EMAIL. Ficou decidido que nessa coluna serão inseridos até 50 caracteres alfa numéricos. Qual dos seguintes comandos deve ser utilizado para adicionar a coluna EMAIL na tabela ALUNO?

Alternativas:

- ADD TABLE ALUNO (EMAIL) VARCHAR2(50);
- ALTER TABLE ALUNO ADD COLUMN EMAIL VARCHAR2(50);
- ALTER TABLE EMAIL ADD ALUNO VARCHAR2(50);
- ALTER TABLE ALUNO ADD EMAIL VARCHAR2(50); (Correta)
- ALTER COLUMN ALUNO ADD EMAIL VARCHAR2(50);

Resumo: O comando ALTER TABLE ALUNO ADD EMAIL VARCHAR2(50); é o correto, pois segue a sintaxe do Oracle para adicionar uma nova coluna a uma tabela existente.

Pergunta: Uma tabela denominada CLIENTE é composta por duas colunas: CODIGO (NUMBER, tamanho 4) e NOME (VARCHAR2, tamanho 50). Que comando deve ser utilizado para inserir uma nova linha na tabela cliente?

Alternativas:

- INSERT INTO CLIENTE (CODIGO, NOME) VALUES (1001, 'Ana Silva'); (Correta)
- INSERT CLIENTE (CODIGO, NOME) VALUES (1001, 'Ana Silva');
- INSERT CLIENTE (CODIGO, NOME) INTO (1001, 'Ana Silva');
- INSERT INTO CODIGO, NOME (CLIENTE) VALUES (1001, 'Ana Silva');
- INSERT VALUES CLIENTE (CODIGO, NOME) INTO (1001, 'Ana Silva');

Resumo: O comando correto é INSERT INTO CLIENTE (CODIGO, NOME) VALUES (1001, 'Ana Silva'); pois segue a sintaxe padrão do SQL para inserção de registros em tabelas.

Pergunta: A empresa ABC Ltda deseja recompensar todos os funcionários pelos ótimos resultados

obtidos no ano anterior concedendo-lhes um aumento salarial de duzentos reais. Qual dos seguintes comandos atualizara os salarios de todos os funcionarios?

Alternativas:

- ALTER FUNCIONARIO SET SALARIO = SALARIO + 200;
- ALTER FUNCIONARIO SET SALARIO + 200 = SALARIO;
- UPDATE FUNCIONARIO SET SALARIO + 200 = SALARIO;
- UPDATE SALARIO SET FUNCIONARIO = SALARIO + 200;
- UPDATE FUNCIONARIO SET SALARIO = SALARIO + 200; (Correta)

Resumo: O comando UPDATE FUNCIONARIO SET SALARIO = SALARIO + 200; e o correto, pois utiliza a sintaxe adequada do SQL para atualizar todos os registros da coluna SALARIO adicionando 200 ao valor atual.

Pergunta: A empresa ABC Ltda esta retirando de producao todos os produtos da CATEGORIA cujo valor seja igual a 15. Portanto, deverao ser eliminadas todas as linhas da tabela PRODUTO que correspondem a este valor. Que comando devera ser utilizado para isso?

Alternativas:

- UPDATE PRODUTO WHERE CATEGORIA = 15;
- DROP PRODUTO WHERE CATEGORIA = 15;
- DELETE PRODUTO FROM CATEGORIA = 15;
- UPDATE FROM PRODUTO WHERE CATEGORIA = 15;
- DELETE FROM PRODUTO WHERE CATEGORIA = 15; (Correta)

Resumo: O comando DELETE FROM PRODUTO WHERE CATEGORIA = 15; e o correto, pois remove todas as linhas da tabela PRODUTO que atendem a condicao especificada.

### Atividade 3

Pergunta: Qual das seguintes funções apresentará os números de meses entre duas datas?

Alternativas:

- NEXT\_DAY
- SYSDATE
- NEXT\_MONTH
- MONTHS\_BETWEEN (Correta)
- LAST\_MONTH

Resumo: A função MONTHS\_BETWEEN retorna a quantidade de meses entre duas datas, sendo a opção correta para calcular diferenças temporais em meses.

Pergunta: Qual das seguintes consultas apresentará os nomes dos alunos que obtiveram as menores notas?

Alternativas:

- SELECT NOME FROM ALUNO WHERE NOTA = (SELECT AVG(NOTA) FROM ALUNO);
- SELECT NOME FROM ALUNO WHERE NOTA = (SELECT UPPER(NOTA) FROM ALUNO);
- SELECT NOME FROM ALUNO WHERE NOTA = (SELECT LOWER(NOTA) FROM ALUNO);
- SELECT NOME FROM ALUNO WHERE NOTA = (SELECT MIN(NOTA) FROM ALUNO); (Correta)
- SELECT NOME FROM ALUNO WHERE NOTA = (SELECT MAX(NOTA) FROM ALUNO);

Resumo: A função MIN retorna o menor valor de uma coluna. Assim, a consulta com MIN(NOTA) identifica os alunos com as menores notas.

Pergunta: Qual das seguintes consultas retorna os nomes dos funcionários e os nomes dos seus respectivos cargos?

Alternativas:

- SELECT FUNCIONARIO, CARGO FROM NOME\_CARGO INNER JOIN NOME\_FUNCIONARIO ON (COD\_CARGO = COD\_FUNCIONARIO);
- SELECT FUNCIONARIO, CARGO FROM NOME\_CARGO INNER JOIN NOME\_FUNCIONARIO ON (CARGO.COD\_CARGO = FUNCIONARIO.COD\_CARGO);
- SELECT FUNCIONARIO, CARGO FROM CARGO INNER JOIN FUNCIONARIO ON (CARGO.COD\_CARGO = FUNCIONARIO.COD\_CARGO);
- SELECT NOME\_FUNCIONARIO, NOME\_CARGO FROM CARGO INNER JOIN FUNCIONARIO ON (CARGO.COD\_CARGO = FUNCIONARIO.COD\_CARGO); (Correta)
- SELECT NOME\_FUNCIONARIO, NOME\_CARGO FROM CARGO INNER JOIN FUNCIONARIO ON (COD\_CARGO = COD\_FUNCIONARIO);

Resumo: A consulta correta utiliza INNER JOIN relacionando CARGO e FUNCIONARIO pela chave COD\_CARGO, retornando NOME\_FUNCIONARIO e NOME\_CARGO.

Pergunta: Qual das seguintes consultas retorna os números dos pedidos emitidos pelo vendedor

cujo nome e 'JOSE SILVA'?

Alternativas:

- SELECT NUMEROPEDIDO FROM PEDIDO INNER JOIN VENDEDOR ON (VENDEDOR.CODVENDEDOR = PEDIDO.CODVENDEDOR) WHERE NOMEVENDEDOR = 'JOSE SILVA'; (Correta)
- SELECT PEDIDO FROM NUMEROPEDIDO INNER JOIN CODVENDEDOR ON (VENDEDOR.CODVENDEDOR = PEDIDO.CODVENDEDOR) WHERE NOMEVENDEDOR = 'JOSE SILVA';
- SELECT PEDIDO FROM NUMEROPEDIDO INNER JOIN CODVENDEDOR ON (CODVENDEDOR = CODVENDEDOR) WHERE NOMEVENDEDOR = 'JOSE SILVA';
- SELECT NUMEROPEDIDO FROM VENDEDOR INNER JOIN VENDEDOR ON (VENDEDOR.CODVENDEDOR = PEDIDO.CODVENDEDOR) WHERE NOMEVENDEDOR = 'JOSE SILVA';
- SELECT NUMEROPEDIDO FROM PEDIDO INNER JOIN VENDEDOR WHERE NOMEVENDEDOR = 'JOSE SILVA';

Resumo: A consulta correta faz INNER JOIN entre PEDIDO e VENDEDOR usando CODVENDEDOR e filtra pelo nome 'JOSE SILVA', retornando os pedidos dele.

Pergunta: Qual das seguintes consultas retorna os nomes dos funcionarios e os nomes dos seus respectivos cargos? (Nota: A consulta devera retornar inclusive os nomes dos funcionarios que ainda nao tem um cargo definido.)

Alternativas:

- SELECT NOMEFUNCIONARIO, NOMECargo FROM CARGO LEFT OUTER JOIN FUNCIONARIO ON (CARGO.CODCARGO = FUNCIONARIO.CODCARGO); (Correta)
- SELECT FUNCIONARIO, CARGO FROM CARGO RIGHT OUTER JOIN FUNCIONARIO ON (CARGO.CODCARGO = FUNCIONARIO.CODCARGO);
- SELECT FUNCIONARIO, CARGO FROM NOMECargo LEFT OUTER JOIN NOMEFUNCIONARIO ON (CODCARGO = CODCARGO);
- SELECT NOMEFUNCIONARIO, NOMECargo FROM CARGO LEFT OUTER JOIN FUNCIONARIO ON (CODCARGO = CODCARGO);
- SELECT NOMEFUNCIONARIO, NOMECargo FROM CARGO RIGHT OUTER JOIN FUNCIONARIO ON (CARGO.CODCARGO = FUNCIONARIO.CODCARGO);

Resumo: A consulta correta utiliza LEFT OUTER JOIN, garantindo que todos os funcionarios aparecam, mesmo aqueles sem cargo associado.

## Atividade 4

Pergunta: Um Analista de RH precisa emitir um relatório com todos os funcionários que ganham auxílio alimentação. O relatório deverá apresentar somente as colunas nome, departamento e aux\_alimentacao organizando os dados por nome de forma ascendente. Considerando que o relatório será emitido com um comando SQL em um banco de dados Oracle, que possui uma tabela chamada funcionario, a instrução SQL que deverá ser utilizada é:

Alternativas:

- SELECT nome, departamento, aux\_alimentacao FROM funcionario WHERE aux\_alimentacao IS NOT NULL ORDER BY nome DESC;
- SELECT funcionario, departamento, aux\_alimentacao FROM funcionario WHERE aux\_alimentacao IS NOT NULL ORDER BY nome ASC;
- SELECT nome, departamento, aux\_alimentacao FROM funcionario WHERE aux\_alimentacao IS NOT NULL ORDER BY nome ASC; (Correta)
- SELECT nome, departamento, aux\_alimentacao FROM funcionario WHERE aux\_alimentacao IS NULL ORDER BY nome ASC;
- SELECT nome, departamento, aux\_alimentacao FROM funcionario WHERE aux\_alimentacao IS NOT EMPTY ORDER BY nome ASC;

Resumo: A consulta correta seleciona apenas as colunas desejadas, filtra os funcionários que possuem auxílio alimentação (IS NOT NULL) e organiza os resultados em ordem ascendente pelo nome.

Pergunta: Considere a tabela criada abaixo. Qual das seguintes consultas retorna os nomes dos empregados e os nomes dos seus respectivos gerentes?

Alternativas:

- SELECT E.EMPREGADO NOME\_EMP, G.GERENTE NOME\_EMP FROM EMPREGADO E, GERENTE G WHERE E.COD\_GER = G.COD\_EMP;
- SELECT NOME\_EMP EMPREGADO, NOME\_EMP GERENTE FROM EMPREGADO, EMPREGADO WHERE COD\_GER = COD\_EMP;
- SELECT E.NOME\_EMP EMPREGADO, G.NOME\_GER GERENTE FROM EMPREGADO E, GERENTE G WHERE E.COD\_GER = G.COD\_GER; (Correta)
- SELECT NOME\_EMP.EMPREGADO, NOME\_EMP.GERENTE FROM EMPREGADO WHERE E.COD\_GER = G.COD\_GER;
- SELECT E.NOME\_EMP EMPREGADO, G.NOME\_EMP GERENTE FROM EMPREGADO E, GERENTE G;

Resumo: A consulta correta faz o join entre EMPREGADO e GERENTE utilizando a chave estrangeira COD\_GER, retornando o nome do empregado e o nome do gerente correspondente.

Pergunta: Considere as tabelas criadas abaixo. Qual das seguintes consultas retorna os nomes dos clientes das duas filiais (A e B)?

Alternativas:

- SELECT NOMECLIENTE FROM FILIAL\_A MINUS SELECT NOMECLIENTE FROM FILIAL\_B;
- SELECT NOMECLIENTE FROM FILIAL\_A FULL OUTER JOIN NOMECLIENTE FROM FILIAL\_B;
- SELECT NOMECLIENTE FROM FILIAL\_A LEFT OUTER JOIN NOMECLIENTE FROM FILIAL\_B;
- SELECT NOMECLIENTE FROM FILIAL\_A UNION SELECT NOMECLIENTE FROM FILIAL\_B;

(Correta)

- SELECT NOMECLIENTE FROM FILIAL\_A INNER JOIN NOMECLIENTE FROM FILIAL\_B;

Resumo: A operacao UNION combina os registros de ambas as tabelas, retornando todos os clientes das duas filiais sem duplicatas.

Pergunta: Considere as tabelas criadas abaixo. Qual das seguintes consultas retorna os nomes dos clientes comuns as duas filiais (A e B)?

Alternativas:

- SELECT NOMECLIENTE FROM FILIAL\_A FULL OUTER JOIN NOMECLIENTE FROM FILIAL\_B;
- SELECT NOMECLIENTE FROM FILIAL\_A LEFT OUTER JOIN NOMECLIENTE FROM FILIAL\_B;
- SELECT NOMECLIENTE FROM FILIAL\_A MINUS SELECT NOMECLIENTE FROM FILIAL\_B;
- SELECT NOMECLIENTE FROM FILIAL\_A INTERSECT SELECT NOMECLIENTE FROM FILIAL\_B;

(Correta)

- SELECT NOMECLIENTE FROM FILIAL\_A UNION SELECT NOMECLIENTE FROM FILIAL\_B;

Resumo: A operacao INTERSECT retorna apenas os registros que estao presentes em ambas as tabelas, ou seja, os clientes comuns as duas filiais.

Pergunta: Que clausula deve ser adicionada para impedir operacoes DML sobre uma VIEW, restringindo-a, portanto, apenas a operacoes de leitura?

Alternativas:

- WITH SELECT ONLY
- WITH WRITE ONLY
- WITH READ AND WRITE
- WITH READ AND WRITE ONLY
- WITH READ ONLY (Correta)

Resumo: A clausula WITH READ ONLY garante que a VIEW seja apenas de leitura, impedindo insercoes, atualizacoes ou exclusoes.



