Explicação para cada comando SQL fornecido:

1. `SELECT \* FROM nome da tabela;`

- Este comando seleciona todas as colunas de uma tabela específica, retornando todas as linhas.

2. `SELECT \* FROM tabelafornecedores;`

- Similar ao primeiro comando, mas específico para a tabela "tabelafornecedores".

3. `SELECT \* FROM nome da tabela WHERE nome da coluna = dado que desejamos consultar;`

- Seleciona todas as colunas de uma tabela onde um determinado valor em uma coluna específica corresponde a um valor fornecido.

4. `SELECT \* FROM tabelafornecedores WHERE país\_de\_origem = ‘China’;`

- Seleciona todas as colunas da tabela "tabelafornecedores" onde o país de origem é "China".

5. `SELECT DISTINCT nome da coluna FROM nome da tabela;`

- Seleciona valores únicos de uma coluna específica em uma tabela, removendo duplicatas.

6. `SELECT DISTINCT cliente FROM tabelapedidos;`

- Seleciona clientes únicos da tabela "tabelapedidos".

7. `CREATE TABLE tabelaclientes (...)`

- Cria uma nova tabela chamada "tabelaclientes" com as colunas especificadas.

8. `ALTER TABLE tabelaclientes ADD Endereço\_Cliente VARCHAR (250);`

- Adiciona uma nova coluna chamada "Endereço\_Cliente" à tabela "tabelaclientes".

9. `DROP TABLE tabelaclientes;`

- Exclui permanentemente a tabela "tabelaclientes" e todos os seus dados.

10. `CREATE TABLE tabelacategorias (...)`

- Cria uma nova tabela chamada "tabelacategorias" com as colunas especificadas.

11. `CREATE TABLE tabelaprodutos (...)`

- Cria uma nova tabela chamada "tabelaprodutos" com as colunas especificadas, incluindo chaves estrangeiras para referenciar outras tabelas.

12. `CREATE TABLE tabelapedidosgold (...)`

- Cria uma nova tabela chamada "tabelapedidosgold" para pedidos de alto valor com as colunas especificadas, incluindo uma chave estrangeira para referenciar a tabela de clientes.

13. `INSERT INTO tabelapedidosgold (...) SELECT ... FROM tabelapedidos WHERE total\_do\_pedido >= 400;`

- Insere registros na tabela "tabelapedidosgold" selecionando dados da tabela "tabelapedidos" onde o total do pedido é maior ou igual a 400.

14. `SELECT \* FROM tabelapedidos WHERE total\_do\_pedido > 200;`

- Seleciona todos os pedidos da tabela "tabelapedidos" onde o total do pedido é maior que 200.

15. `SELECT \* FROM tabelapedidos WHERE total\_do\_pedido >= 200;`

- Seleciona todos os pedidos da tabela "tabelapedidos" onde o total do pedido é maior ou igual a 200.

f

- Seleciona todos os pedidos da tabela "tabelapedidos" onde o total do pedido é menor que 200.

17. `SELECT \* FROM tabelapedidos WHERE total\_do\_pedido <= 200;`

- Seleciona todos os pedidos da tabela "tabelapedidos" onde o total do pedido é menor ou igual a 200.

18. `SELECT \* FROM tabelapedidos WHERE total\_do\_pedido <> 200;`

- Seleciona todos os pedidos da tabela "tabelapedidos" onde o total do pedido não é igual a 200.

19. `SELECT \* FROM tabelaclientes WHERE nome\_cliente > ‘C’;`

- Seleciona todos os clientes cujos nomes vêm após "C" em ordem alfabética.

20. `SELECT \* FROM tabelapedidos WHERE data\_do\_pedido > ‘2023-09-19’;`

- Seleciona todos os pedidos feitos após 19 de setembro de 2023.

21. `SELECT \* FROM tabelapedidos WHERE total\_do\_pedido >= 200 AND Status = ‘Pendente’;`

- Seleciona pedidos com um total de pedido superior ou igual a 200 e que têm status "Pendente".

22. `SELECT \* FROM tabelapedidos WHERE status = ‘Pendente’ OR status = ‘Processando’;`

- Seleciona pedidos com status "Pendente" ou "Processando".

23. `SELECT \* FROM tabelapedidos WHERE NOT status = ‘Pendente’;`

- Seleciona pedidos que não têm status "Pendente".

24. `SELECT \* FROM tabelapedidos WHERE data\_de\_envio\_estimada BETWEEN ‘2023-08-01’ AND ‘2023-09-01’;`

- Seleciona pedidos com datas de envio estimadas entre 1 de agosto de 2023 e 1 de setembro de 2023.

25. `SELECT \* FROM tabelaprodutos WHERE preco\_de\_compra BETWEEN 200 AND 600 ORDER BY nome\_do\_produto;`

- Seleciona produtos com preços de compra entre 200 e 600 e ordena-os por nome do produto.

26. `SELECT \* FROM tabelaprodutos WHERE preco\_de\_compra BETWEEN 200 AND 600 ORDER BY data\_de\_inclusao;`

- Seleciona produtos com preços de compra entre 200 e 600 e ordena-os por data de inclusão.

27. `SELECT \* FROM tabelaprodutos WHERE preco\_de\_compra BETWEEN 200 AND 600 ORDER BY fornecedor DESC;`

- Seleciona produtos com preços de compra entre 200 e 600 e ordena-os pelo fornecedor em ordem decrescente.

28. `SELECT informacoes\_de\_contato AS email\_cliente FROM tabelaclientes;`

- Seleciona informações de contato da tabela de clientes, renomeando a coluna "informacoes\_de\_contato" como "email\_cliente".

29. `UPDATE tabelapedidos SET status = 'Enviado' WHERE status = 'Processando';`

- Este comando atualiza a coluna "status" da tabela "tabelapedidos", alterando todos os registros com status "Processando" para "Enviado".

30. `UPDATE tabelaclientes SET informacoes\_de\_contato = 'j.santos@email.com', endereço\_cliente = 'Rua dos paralelepípedos, 30' WHERE id\_cliente = 2;`

- Este comando atualiza as colunas "informacoes\_de\_contato" e "endereço\_cliente" da tabela "tabelaclientes" para o cliente com ID 2.

31. `DELETE FROM tabelafornecedores WHERE país\_de\_origem = ‘Turquia’;`

- Este comando exclui todos os fornecedores da tabela "tabelafornecedores" que têm o país de origem como "Turquia".

32. `DELETE FROM tabelafornecedores WHERE id > 35;`

- Este comando exclui fornecedores da tabela "tabelafornecedores" com IDs maiores que 35.

33. `SELECT \* FROM HistoricoEmprego ORDER BY salario DESC LIMIT 5;`

- Este comando seleciona todos os registros da tabela "HistoricoEmprego" e os ordena em ordem decrescente com base no salário. A cláusula `LIMIT 5` garante que apenas os 5 primeiros registros sejam retornados.

34. `SELECT \* FROM HistoricoEmprego WHERE datatermino IS NULL ORDER BY salario DESC LIMIT 5;`

- Este comando seleciona registros da tabela "HistoricoEmprego" onde a data de término (`datatermino`) é nula, ou seja, os registros correspondentes estão atualmente empregados. Os resultados são ordenados em ordem decrescente com base no salário, e apenas os 5 primeiros registros são retornados.

35. `SELECT \* FROM HistoricoEmprego WHERE datatermino IS NOT NULL ORDER BY salario DESC LIMIT 5;`

- Este comando seleciona registros da tabela "HistoricoEmprego" onde a data de término (`datatermino`) não é nula, ou seja, os registros correspondentes não estão mais empregados. Os resultados são ordenados em ordem decrescente com base no salário, e apenas os 5 primeiros registros são retornados.

36. `SELECT \* FROM Treinamento WHERE curso LIKE 'O poder%';`

- Este comando seleciona todos os registros da tabela "Treinamento" onde o nome do curso começa com "O poder".

37. `SELECT \* from Treinamento where curso LIKE '%realizar%';`

- Este comando seleciona todos os registros da tabela "Treinamento" onde o nome do curso contém a palavra "realizar" em qualquer posição.

38. `SELECT \* from Colaboradores where nome LIKE 'Isadora%';`

- Este comando seleciona todos os registros da tabela "Colaboradores" onde o nome do colaborador começa com "Isadora".

39. `SELECT \* FROM HistoricoEmprego WHERE cargo = 'Professor' AND datatermino NOT NULL;`

- Este comando seleciona todos os registros da tabela "HistoricoEmprego" onde o cargo é "Professor" e a data de término não é nula.

40. `SELECT \* FROM HistoricoEmprego WHERE cargo = 'Oftalmologista' OR cargo = 'Dermatologista';`

- Este comando seleciona todos os registros da tabela "HistoricoEmprego" onde o cargo é "Oftalmologista" OU "Dermatologista".

41. `SELECT \* FROM HistoricoEmprego WHERE cargo IN ('Oftalmologista', 'Dermatologista', 'Professor');`

- Este comando seleciona todos os registros da tabela "HistoricoEmprego" onde o cargo é "Oftalmologista", "Dermatologista" ou "Professor".

42. `SELECT \* FROM HistoricoEmprego WHERE cargo NOT IN ('Oftalmologista', 'Dermatologista', 'Professor');`

- Este comando seleciona todos os registros da tabela "HistoricoEmprego" onde o cargo não é "Oftalmologista", "Dermatologista" ou "Professor".

43. `SELECT \* FROM Treinamento WHERE (curso LIKE 'O direito%' AND instituicao = 'da Rocha') OR (curso LIKE 'O conforto%' AND instituicao = 'das Neves');`

- Este comando seleciona todos os registros da tabela "Treinamento" onde o nome do curso começa com "O direito" e a instituição é "da Rocha", OU o nome do curso começa com "O conforto" e a instituição é "das Neves".

1. `SELECT mes, MAX(faturamento\_bruto) FROM faturamento;`

- Esta consulta retorna o mês em que o faturamento bruto atingiu o valor máximo na tabela "faturamento".

2. `SELECT mes, MIN(faturamento\_bruto) FROM faturamento;`

- Esta consulta retorna o mês em que o faturamento bruto atingiu o valor mínimo na tabela "faturamento".

3. `SELECT SUM(numero\_novos\_clientes) AS ‘Novos clientes 2023’ FROM Faturamento WHERE mes LIKE '%2023';`

- Esta consulta retorna a soma do número de novos clientes para todos os meses de 2023.

4. `SELECT AVG(despesas) FROM faturamento;`

- Esta consulta retorna a média das despesas da tabela "faturamento".

5. `SELECT AVG(lucro\_liquido) FROM faturamento;`

- Esta consulta retorna a média do lucro líquido da tabela "faturamento".

6. `SELECT COUNT(\*) FROM HistoricoEmprego WHERE datatermino NOT NULL;`

- Esta consulta conta o número de registros na tabela "HistoricoEmprego" onde a data de término não é nula.

7. `SELECT COUNT(\*) FROM Licencas WHERE tipolicenca = 'férias';`

- Esta consulta conta o número de licenças na tabela "Licencas" onde o tipo de licença é 'férias'.

8. `SELECT parentesco, COUNT(\*) FROM Dependentes GROUP BY parentesco;`

- Esta consulta conta o número de dependentes agrupados por tipo de parentesco na tabela "Dependentes".

9. `SELECT instituicao, COUNT(curso) FROM Treinamento GROUP BY instituicao HAVING COUNT(curso) > 2;`

- Esta consulta conta o número de cursos oferecidos por instituição na tabela "Treinamento" e mostra apenas aquelas instituições que oferecem mais de 2 cursos.

10. `SELECT cargo, COUNT(\*) qtd FROM HistoricoEmprego GROUP BY cargo HAVING qtd >=2;`

- Esta consulta conta o número de registros agrupados por cargo na tabela "HistoricoEmprego" e mostra apenas aqueles cargos que têm 2 ou mais registros.

Aqui estão as explicações para cada uma das consultas SQL fornecidas:

11. `SELECT COUNT(\*), LENGTH(cpf) qtd FROM Colaboradores WHERE qtd = 11;`

- Esta consulta conta o número de registros na tabela "Colaboradores" cujo comprimento do CPF é exatamente 11 dígitos.

12. `SELECT ('A pessoa colaboradora ' || nome || ' de CPF ' || cpf || ' possui o seguinte endereço: ' || endereco) as texto FROM Colaboradores;`

- Esta consulta concatena o nome, CPF e endereço de cada colaborador em uma única string para todos os registros da tabela "Colaboradores".

13. `SELECT UPPER('A pessoa colaboradora ' || nome || ' de CPF ' || cpf || ' possui o seguinte endereço: ' || endereco) as texto from Colaboradores;`

- Esta consulta converte o resultado da consulta anterior para letras maiúsculas.

14. `SELECT LOWER('A pessoa colaboradora ' || nome || ' de CPF ' || cpf || ' possui o seguinte endereço: ' || endereco) as texto from Colaboradores;`

- Esta consulta converte o resultado da consulta original para letras minúsculas.

15. `SELECT id\_colaborador, STRFTIME('%Y/%m', datainicio) FROM Licencas;`

- Esta consulta retorna o ID do colaborador e o ano/mês (no formato 'AAAA/MM') da data de início das licenças da tabela "Licencas".

16. `SELECT id\_colaborador, JULIANDAY (datatermino) - JULIANDAY (datacontratacao) FROM HistoricoEmprego WHERE datatermino IS NOT NULL;`

- Esta consulta calcula o tempo em dias entre a data de término e a data de contratação para os registros da tabela "HistoricoEmprego" onde a data de término não é nula.

17. `SELECT AVG(faturamento\_bruto), ROUND (AVG(faturamento\_bruto),2) FROM faturamento;`

- Esta consulta retorna a média do faturamento bruto da tabela "faturamento" e a mesma média arredondada para 2 casas decimais.

18. `SELECT CEIL(faturamento\_bruto), CEIL(despesas) FROM faturamento;`

- Esta consulta retorna o menor inteiro maior ou igual ao faturamento bruto e às despesas para cada registro na tabela "faturamento".

19. `SELECT FLOOR(faturamento\_bruto), FLOOR(despesas) FROM faturamento;`

- Esta consulta retorna o maior inteiro menor ou igual ao faturamento bruto e às despesas para cada registro na tabela "faturamento".

20. `SELECT (' O faturamento bruto médio foi ' || CAST(ROUND (AVG(faturamento\_bruto),2) AS TEXT)) FROM faturamento;`

- Esta consulta retorna uma frase concatenada que descreve o faturamento bruto médio da tabela "faturamento" arredondado para 2 casas decimais. A função `CAST` é usada para converter o valor numérico em texto para concatenar com a string.

21. `SELECT id\_colaborador, cargo, salario,

CASE

WHEN salario < 3000 THEN 'Baixo'

WHEN salario BETWEEN 3000 AND 6000 THEN 'Médio'

ELSE 'Alto'

END AS categoria\_salario

FROM HistoricoEmprego;`

- Esta consulta seleciona o ID do colaborador, o cargo, o salário e categoriza os salários com base em faixas definidas usando a declaração `CASE`. Os salários menores que 3000 são classificados como "Baixo", os salários entre 3000 e 6000 são classificados como "Médio" e os salários maiores que 6000 são classificados como "Alto".

22. `ALTER TABLE HistoricoEmprego RENAME TO CargosColaboradores;`

- Este comando altera o nome da tabela "HistoricoEmprego" para "CargosColaboradores". Isso é útil para refletir mais precisamente o conteúdo da tabela, uma vez que ela agora contém informações sobre cargos de colaboradores.