



• Gabarito •



Utilize este gabarito apenas para conferir as respostas dos exercícios propostos. Elas são meramente ilustrativas e não necessariamente as melhores ou únicas.

CAPÍTULO 1

1. Defina campo, registro, tabela e banco de dados.

Resposta: Banco de dados é o conjunto de registros, registro é o conjunto de campos e campo é a menor unidade de armazenamento.

2. Quem desenvolveu banco de dados relacional?

Resposta: O modelo relacional de banco de dados foi desenvolvido pelo cientista inglês Edgar Frank "Ted" Codd.

3. Qual é o significado da sigla SQL em inglês e em português?

Resposta: SQL significa Structured Query Language, ou seja, Linguagem Estruturada de Consulta.

4. Como um banco de dados relacional armazena seus dados?

Resposta: Armazena os dados em forma de tabela.

5. Qual o significado das siglas SGBD e SGBDR?

Resposta:

- **SGBD** - Sistema de Gerenciamento de Banco de Dados.
- **RDBMS** - Relational Data Base Management System.

6. Quem foi o principal desenvolvedor da linguagem SQL?

Resposta: foi o Ph.D. Donald D. Chamberlin.

7. Quem foi o desenvolvedor do sistema de gerenciamento de banco de dados SEQUEL-XRM?

Resposta: Foi a IBM.

8. Quais são os subconjuntos de palavras-chave (comandos) adotados pelo gerenciador de banco de dados MySQL?

Resposta: A linguagem de consulta estruturada SQL atualmente é composta por seis subconjuntos ou grupos de palavras-chave que formam a estrutura básica das instruções: DML (Data Manipulation Language - Linguagem de Manipulação de Dados), DDL (Data Definition Language - Linguagem de Definição de Dados), DQL (Data Query Language - Linguagem de Pesquisa de Dados), DCL (Data Control Language - Linguagem de Controle de Dados), DAC (Data Administration Commands - Comandos de Administração de Dados) e TCC (Transactions Control Commands - Comandos de Controle de Transações). No caso do gerenciador de banco de dados MySQL esse subconjunto de comandos é dividido do seguinte modo e formado pelos comandos com a possibilidade de acrescentar os subconjuntos SRC (Stored Routines Commands - Comandos de Rotinas Armazenadas) e DTC (Data Type Commands - Comandos de Tipos de Dados).

9. Quem são os desenvolvedores do programa MySQL e em que ano e local ele foi lançado?

Resposta: O MySQL é oriundo da Suécia e as idéias para seu desenvolvimento surgiram em 1979 pelas mãos de um sueco: David Axmark e um finlandês: Michael Widenius, também conhecido como Michael "Monty" Widenius, que trabalhavam juntos como programadores numa pequena empresa denominada TcX.

10. Além da linguagem SQL, quais outras são de quarta geração?

Resposta:

- ADF (Application Development Facility);
- ADRS II (A Departmental Reporting System);
- AS (Application System);
- CSP (Cross System Product Set);
- DMS (Development Management System);
- GIS (Generalized Information System);
- IC/1 (Information Center/1);
- QBE (Query By Example);
- QMF (Query Management Facility);
- TIF (The Information Facility).

CAPÍTULO 2

1. Qual a finalidade da instrução SELECT VERSION()?

Resposta: Apresenta o número da versão do programa em uso.

2. Qual a finalidade da instrução SELECT CURRENT_DATE?

Resposta: Apresenta a data corrente do sistema em que o programa está em execução.

3. Qual a finalidade da instrução SELECT CURRENT_TIME?

Resposta: Exibe a hora corrente do sistema em que o programa está em execução.

4. Qual a finalidade da instrução SELECT CURDATE()?

Resposta: Mostra a data corrente do sistema em que o programa está em execução.

5. Qual a finalidade da instrução SELECT CURTIME()?

Resposta: Apresenta a hora corrente do sistema em que o programa está em execução.

6. Qual a finalidade da instrução SELECT NOW()?

Resposta: Apresenta a data e a hora corrente do sistema em que o programa está em execução.

7. Qual a finalidade do comando help contents?

Resposta: Mostra uma lista de opções das quais se pode obter ajuda.

8. Qual a finalidade do comando help contents?

Resposta: Apresenta uma lista de opções das quais se pode obter ajuda.

CAPÍTULO 3

1. Qual a finalidade da instrução CREATE DATABASE?

Resposta: A finalidade é criar outro banco de dados.

2. Qual a finalidade da instrução CREATE DATABASE IF NOT EXISTS?

Resposta: Criar outro banco de dados caso o banco a ser criado não exista.

3. Qual a finalidade da instrução SHOW DATABASE?

Resposta: A finalidade é mostrar todos os bancos de dados existentes.

4. Qual a finalidade da instrução USE?

Resposta: Colocar em uso um banco de dados existente.

5. Qual a finalidade da instrução DROP DATABASE?

Resposta: A finalidade é remover um banco de dados existente.

6. Qual a finalidade da instrução CREATE TABLE?

Resposta: Criar uma nova tabela em um banco de dados existente.

7. Qual a finalidade da instrução SHOW TABLE?

Resposta: Apresentar todas as tabelas existentes em um banco de dados.

8. Qual a finalidade da instrução DESCRIBE?

Resposta: Apresentar a estrutura de uma determinada tabela.

9. Qual a finalidade da instrução INSERT INTO?

Resposta: A finalidade é efetuar a entrada de registros em uma tabela.

10. Qual a finalidade da instrução LOAD DATA LOCAL INFILE?

Resposta: Carregar os dados armazenados em um arquivo texto e efetuar a entrada desses dados na tabela em uso.

11. Qual a finalidade da instrução SELECT com FROM?

Resposta: A finalidade é fazer uma consulta e apresentar os registros armazenados em uma determinada tabela.

12. Qual a finalidade da instrução UPDATE com SET?

Resposta: Atualização de registros em uma determinada tabela.

13. Qual a finalidade da instrução DELETE FROM?

Resposta: A finalidade é remover (apagar) registros de uma determinada tabela.

14. Qual a finalidade da instrução ALTER TABLE?

Resposta: Alterar a estrutura de uma determinada tabela.

15. Efetuar a listagem dos nomes e funções de todos os funcionários.

Resposta: SELECT NOME, FUNCAO FROM cadfun;.

16. Faça a listagem dos códigos, nomes e funções de todos os funcionários.

Resposta: `SELECT CODFUN, NOME, FUNCAO FROM cadfun;`

17. Enumere os códigos, nomes, funções e departamentos de todos os funcionários.

Resposta: `SELECT CODFUN, NOME, FUNCAO, DEPTO FROM cadfun;`

18. Quais são os nomes dos funcionários do departamento 5?

Resposta: `SELECT NOME FROM cadfun WHERE DEPTO = '5';`

19. Quais são os nomes e os departamentos dos funcionários que ocupam o cargo de vendedor?

Resposta: `SELECT NOME, DEPTO FROM cadfun WHERE FUNCAO = 'VENDEDOR';`

20. Quais são os salários dos funcionários do departamento 3?

Resposta: `SELECT SALARIO FROM cadfun WHERE DEPTO = '3';`

21. Faça uma listagem do nome e da função de todos os funcionários ordenados de forma descendente pela descrição da função.

Resposta: `SELECT NOME, FUNCAO FROM cadfun ORDER BY FUNCAO DESC;`

22. Elabore a listagem das funções e nomes de todos os funcionários ordenando a coluna de função de forma ascendente e a coluna de nomes de modo descendente.

Resposta: `SELECT FUNCAO, NOME FROM cadfun ORDER BY FUNCAO, NOME DESC;`

23. Faça a listagem de todas as colunas dos registros que tenham outubro como mês de admissão.

Resposta: `SELECT * FROM cadfun WHERE MONTH(ADMISSAO) = 10;`

CAPÍTULO 4

1. Quais operadores aritméticos podem ser utilizados no programa MySQL?

Resposta: Podem ser utilizados: adição (+), subtração (-), multiplicação (*), divisão (/) e resto de divisão (%).

2. Quais símbolos são utilizados para alterar a prioridade de cálculo matemático?

Resposta: Deve-se usar os parênteses.

3. Quais são os operadores de comparação suportados pelo programa MySQL?

Resposta: São: > (maior que), < (menor que), = (igual a), <> ou != (diferente de), >= (maior ou igual a), <= (menor ou igual a) e <=> (igual para NULL).

4. Quais são os operadores lógicos suportados pelo programa MySQL?

Resposta: São: AND ou && (operador de conjunção), OR ou || (operador de disjunção), NOT ou ! (operador de negação) e XOR (operador de disjunção exclusiva).

5. Quais são os operadores auxiliares suportados pelo programa MySQL?

Resposta: São: IS NULL (para verificação de campo ou coluna vazios), BETWEEN (para verificar um valor em uma faixa de valores), IN (para verificar se um valor existe na tabela) e LIKE (para verificar um valor que faz busca por semelhança).

6. Os operadores auxiliares podem ser utilizados com quais comandos SQL?

Resposta: Podem ser usados com SELECT, UPDATE e DELETE.

7. Faça uma simulação para apresentar uma consulta com as colunas (campos) CODFUN, NOME, SALARIO com o salário somado a R\$ 250,00.

Resposta: SELECT CODFUN, NOME, SALARIO + 250 FROM cadfun;.

8. Faça uma simulação para apresentar uma consulta com as colunas (campos) CODFUN, NOME, SALARIO com o salário deduzido de 7.5%.

Resposta: SELECT CODFUN, NOME, SALARIO – SALARIO * 0.075 FROM cadfun;.

9. Apresente uma consulta de todos os campos (colunas) dos registros cuja função seja igual a analista.

Resposta: SELECT * FROM cadfun WHERE FUNCAO = 'ANALISTA';.

10. Apresente uma consulta de todos os campos (colunas) dos registros cujo salário seja maior ou igual a R\$ 1.700,00.

Resposta: SELECT * FROM cadfun WHERE SALARIO >= 1700;.

11. Faça uma consulta de todos os campos (colunas) dos registros cujo salário seja maior que R\$ 1.700,00.

Resposta: SELECT * FROM cadfun WHERE SALARIO > 1700;.

12. Apresente uma consulta de todos os campos (colunas) dos registros cujo salário seja menor que R\$ 1.700,00.

Resposta: SELECT * FROM cadfun WHERE SALARIO < 1700;.

13. Apresente uma consulta de todos os campos (colunas) dos registros cujo salário seja igual a R\$ 1.700,00.

Resposta: SELECT * FROM cadfun WHERE SALARIO = 1700;.

14. Apresente uma consulta de todos os campos (colunas) dos registros cujo salário seja maior ou igual a R\$ 2.000,00 dos funcionários do departamento 5.

Resposta: SELECT * FROM cadfun WHERE (SALARIO >= 2000) AND (DEPTO = '5');.

15. Apresente uma consulta de todos os campos (colunas) dos registros dos funcionários com função de programador e de analista.

Resposta: SELECT * FROM cadfun WHERE (FUNCAO = 'ANALISTA') OR (FUNCAO = 'PROGRAMADOR');.

16. Apresente uma consulta de todos os campos (colunas) dos registros dos funcionários com função de programador e de analista que ganham acima de R\$ 1.200,00.

Resposta: SELECT * FROM cadfun WHERE (FUNCAO = 'ANALISTA') OR (FUNCAO = 'PROGRAMADOR') AND (SALARIO > 1200);.

17. Apresente uma consulta de todos os campos (colunas) dos registros dos funcionários que não ocupem as funções de programador e de analista.

Resposta: SELECT * FROM cadfun WHERE NOT (FUNCAO = 'ANALISTA') AND NOT (FUNCAO = 'PROGRAMADOR');.

18. Apresente uma consulta de todos os campos (colunas) dos registros dos funcionários que possuem entre 2 e 4 filhos.

Resposta: SELECT * FROM cadfun WHERE FILHOS BETWEEN 2 AND 4;.

19. Apresente uma consulta de todos os campos (colunas) dos registros dos funcionários que possuem entre 2 e 4 filhos que recebem salário abaixo de R\$ 2.000,00.

Resposta: SELECT * FROM cadfun WHERE (FILHOS BETWEEN 2 AND 4) AND (SALARIO < 2000);.

20. Apresente uma consulta de todos os campos (colunas) dos registros dos funcionários que possuem abaixo de 2 e acima de 3 filhos.

Resposta: SELECT * FROM cadfun WHERE FILHOS NOT BETWEEN 2 AND 3;.

21. Apresente uma consulta de todos os campos (colunas) dos registros dos funcionários que possuem abaixo de 2 e acima de 3 filhos, e a consulta somente indique os registros dos funcionários que possuem filhos.

Resposta: `SELECT * FROM cadfun WHERE (FILHOS NOT BETWEEN 2 AND 3) AND (FILHOS <> 0);`.

22. Apresente uma consulta de todos os campos (colunas) dos registros dos funcionários que possuem entre 2 e 3 filhos, utilizando o operador IN.

Resposta: `SELECT * FROM cadfun WHERE FILHOS IN (2, 3);`.

23. Apresente uma consulta de todos os campos (colunas) dos registros dos funcionários que possuem o sobrenome SILVA.

Resposta: `SELECT * FROM cadfun WHERE NOME LIKE '%SILVA';`.

24. Apresente uma consulta de todos os campos (colunas) dos registros dos funcionários cujos nomes comecem com SILV.

Resposta: `SELECT * FROM cadfun WHERE NOME LIKE 'SILV%';`.

25. Quais são os funcionários que possuem o sobrenome SANTOS? Apresente apenas os nomes.

Resposta: `SELECT NOME FROM cadfun WHERE NOME LIKE '%SANTOS';`.

26. Qual é o nome e o departamento dos gerentes e dos analistas?

Resposta: `SELECT NOME, DEPTO FROM cadfun WHERE (FUNCAO = 'GERENTE') OR (FUNCAO = 'ANALISTA');`.

27. Faça a lista dos funcionários de códigos 2, 5 e 9. 27. Listar os funcionários de códigos 2, 5 e 9. Apresentar apenas os códigos, nomes e departamentos.

Resposta: `SELECT CODFUN, NOME, DEPTO FROM cadfun WHERE (CODFUN = 2) OR (CODFUN = 5) OR (CODFUN = 9);`.

28. Liste os nomes e departamentos de todos os funcionários que não sejam do departamento 5.

Resposta: `SELECT NOME, DEPTO FROM cadfun WHERE DEPTO <> '5';`.

29. Apresente uma consulta de todos os campos (colunas) dos registros dos funcionários que possuem em qualquer posição de seus nomes a palavra SILVA.

Resposta: `SELECT * FROM cadfun WHERE NOME LIKE '%SILVA%';`.

30. Apresente uma consulta de todos os campos (colunas) dos registros cujo salário seja diferente de R\$ 2.000,00.

Resposta: `SELECT * FROM cadfun WHERE SALARIO <> 2000;`.

CAPÍTULO 5

1. Quais são as categorias de funções apresentadas no capítulo 5?

Resposta: Foram apresentadas funções de agregação, data e hora, numéricas e strings.

2. Qual categoria de função é ideal para cálculos matemáticos?

Resposta: O ideal são as funções numéricas.

3. Qual categoria de função é ideal para operações estatísticas?

Resposta: Deve-se usar as funções de agregação.

4. Quais foram as funções de agregação apresentadas no capítulo 5?

Resposta: Foram apresentadas as funções `AVG()`, `COUNT()`, `MAX()`, `MIN()`, `STD()`, `STDDEV()`, `SUM()` e `VARIANCE()`.

5. Quais foram as funções de data e hora apresentadas no capítulo 5?

Resposta: Foram apresentadas as funções `CURDATE()`, `CURTIME()`, `DATEDIFF()`, `DAY()`, `HOUR()`, `MINUTE()`, `MONTH()`, `MONTHNAME()`, `NOW()`, `SECOND()` e `YEAR()`.

6. Quais foram as funções numéricas apresentadas no capítulo 5?

Resposta: Foram apresentadas as funções `ABS()`, `ACOS()`, `ASIN()`, `ATAN()`, `COS()`, `DEGREES()`, `EXP()`, `LN()`, `MOD()`, `PI()`, `POWER()`, `RADIANS()`, `ROUND()`, `SIN()`, `SQRT()` e `TAN()`.

7. Quais foram as funções strings apresentadas no capítulo 5?

Resposta: Foram apresentadas as funções `ASCII()`, `BIN()`, `CONCAT()`, `HEX()`, `LOWER()`, `LEFT()`, `LENGTH()` e `UPPE()`.

8. Qual é o número médio de filhos por funcionário da empresa?

Resposta: `SELECT AVG(FILHOS) FROM cadfun;`

9. Qual é o número médio de filhos por funcionário da empresa que pertence ao departamento 5?

Resposta: `SELECT AVG(FILHOS) FROM cadfun WHERE DEPTO = '5';`

10. Qual é o valor de salários pagos aos funcionários do departamento 4?

Resposta: `SELECT SUM(SALARIO) FROM cadfun WHERE DEPTO = '4';`.

11. Quantos funcionários estão no departamento 5?

Resposta: `SELECT COUNT(*) FROM cadfun WHERE DEPTO = '5';`.

12. Quantos funcionários do departamento 5 ganham mais de R\$ 2.000,00?

Resposta: `SELECT COUNT(*) FROM cadfun WHERE (DEPTO = '5') AND (SALARIO > 2000);`.

13. Quantos funcionários do departamento 5 ganham R\$ 2.000,00?

Resposta: `SELECT COUNT(*) FROM cadfun WHERE (DEPTO = '5') AND (SALARIO = 2000);`.

14. Quantos funcionários do departamento 4 ganham entre R\$ 1.000,00 e R\$ 1.500,00?

Resposta: `SELECT COUNT(*) FROM cadfun WHERE (DEPTO = '4') AND (SALARIO BETWEEN 1000 AND 1500);`.

15. Quantos funcionários da empresa possuem o sobrenome SANTOS?

Resposta: `SELECT COUNT(*) FROM cadfun WHERE NOME LIKE '%SANTOS';`.

16. Qual é o maior valor de salário pago para o departamento 4?

Resposta: `SELECT MAX(SALARIO) FROM cadfun WHERE DEPTO = '4';`.

17. Quanto a empresa paga para todos os analistas?

Resposta: `SELECT SUM(SALARIO) FROM cadfun WHERE FUNCAO = 'ANALISTA';`.

18. Quanto a empresa paga para os analistas do departamento 2?

Resposta: `SELECT SUM(SALARIO) FROM cadfun WHERE (FUNCAO = 'ANALISTA') AND (DEPTO = '2');`.

19. Liste os nomes de todos os funcionários admitidos no dia 10 de qualquer mês.

Resposta: `SELECT NOME FROM cadfun WHERE DAY(ADMISSAO) = 10;`.

20. Liste os nomes e as datas de admissão de todos os funcionários admitidos entre o dia 5 e o dia 10 de qualquer mês.

Resposta: `SELECT NOME, ADMISSAO FROM cadfun WHERE (DAY(ADMISSAO) >= 5) AND (DAY(ADMISSAO) <= 10);`.

21. Liste os nomes e as datas de admissão de todos os funcionários admitidos entre o dia 5 e o dia 10 de qualquer mês, ordenados de modo ascendente por data.

Resposta: `SELECT NOME, ADMISSAO FROM cadfun WHERE (DAY(ADMISSAO) >= 5) AND (DAY(ADMISSAO) <= 10) ORDER BY ADMISSAO;`

22. Liste os nomes e as datas de admissão de todos os funcionários admitidos entre o dia 5 e o dia 10 de qualquer mês, ordenados de forma decrescente por data.

Resposta: `SELECT NOME, ADMISSAO FROM cadfun WHERE (DAY(ADMISSAO) >= 5) AND (DAY(ADMISSAO) <= 10) ORDER BY ADMISSAO DESC;`

23. Liste todos os funcionários admitidos antes de 20 de setembro de 2006.

Resposta: `SELECT * FROM cadfun WHERE DATE(ADMISSAO) < '2006-09-20';`

24. Liste os nomes, data de admissão e a descrição por extenso do nome dos meses de admissão de todos os funcionários do departamento 4.

Resposta: `SELECT NOME, ADMISSAO, MONTHNAME(ADMISSAO) FROM cadfun WHERE depto = '4';`

25. Liste os nomes e os departamentos de todos os funcionários em letras minúsculas que pertençam aos departamentos 4 e 5.

Resposta: `SELECT LOWER(NOME), DEPTO FROM cadfun WHERE (DEPTO = '4') OR (DEPTO = '5');`

26. Liste os nomes, departamentos e as datas de admissão de todos os funcionários admitidos entre o dia 5 e o dia 10 de qualquer mês dos departamentos 4 e 5.

Resposta: `SELECT NOME, DEPTO, ADMISSAO FROM cadfun WHERE (DAY(ADMISSAO) >= 5) AND (DAY(ADMISSAO) <= 10) AND (DEPTO = '4') OR (DEPTO = '5');`

27. Liste os nomes, departamentos, funções e as datas de admissão de todos os funcionários admitidos entre o dia 1º e o dia 15 que sejam analistas.

Resposta: `SELECT NOME, DEPTO, FUNCAO, ADMISSAO FROM cadfun WHERE (DAY(ADMISSAO) >= 1) AND (DAY(ADMISSAO) <= 15) AND (FUNCAO = 'ANALISTA');`

28. Liste os nomes, departamentos, funções e as datas de admissão de todos os funcionários admitidos entre os meses de maio e agosto de qualquer ano, que ocupem o cargo de analistas.

Resposta: `SELECT NOME, DEPTO, FUNCAO, ADMISSAO FROM cadfun WHERE (MONTH(ADMISSAO) >= 5) AND (MONTH(ADMISSAO) <= 8) AND (FUNCAO = 'ANALISTA');`

29. Liste os nomes, departamentos, funções e as datas de admissão de todos os funcionários admitidos entre os meses de maio e agosto de qualquer ano que não ocupam os cargos de analistas.

Resposta: `SELECT NOME, DEPTO, FUNCAO, ADMISSAO FROM cadfun WHERE (MONTH(ADMISSAO) >= 5) AND (MONTH(ADMISSAO) <= 8) AND NOT (FUNCAO = 'ANALISTA');`.

29. Quantos funcionários foram admitidos no mês de julho?

Resposta: `SELECT COUNT(*) FROM cadfun WHERE MONTH(ADMISSAO) = 7;`.

CAPÍTULO 6

1. O que é agrupamento de dados?

Resposta: O agrupamento de dados é, segundo STEPHENS & PLEW (2003), "o processo de combinação de colunas com valores repetidos em uma ordem lógica".

2. Qual é a finalidade da cláusula GROUP BY?

Resposta: Permitir a definição e uso do agrupamento de dados.

3. Como deve ser utilizada a cláusula GROUP BY?

Resposta: A cláusula GROUP BY deve ser utilizada após a cláusula WHERE e antes da cláusula ORDER BY.

4. O que é união de dados?

Resposta: É a união de mais de uma consulta em uma única por meio de múltiplos comandos SELECT.

5. Quais são os cargos existentes na empresa e quais os funcionários que ocupam cada um dos cargos existentes?

Resposta: `SELECT FUNCAO, COUNT(*) FROM cadfun GROUP BY FUNCAO;`.

6. Quais são os cargos existentes na empresa para o departamento 5 e qual o total de funcionários que ocupam os cargos do departamento 5?

Resposta: `SELECT FUNCAO, DEPTO, COUNT(*) FROM cadfun WHERE DEPTO = '5' GROUP BY FUNCAO;`.

7. Qual a distribuição de filhos por funcionários que tenham acima de 1 filho? Apresente os funcionários que possuem 1, 2, 3 ou 4 filhos.

Resposta: `SELECT FILHOS, COUNT(*) FROM cadfun WHERE FILHOS > 0 GROUP BY FILHOS;`.

8. Liste todos os funcionários das tabelas “cadfun” e “morto” que estejam ou estiveram relacionados com o departamento 5 em ordem alfabética ascendente pelo campo NOME.

Resposta: (SELECT * FROM cadfun WHERE DEPTO = '5') UNION (SELECT * FROM morto WHERE DEPTO = '5') ORDER BY NOME;

9. Liste todos os funcionários das tabelas “cadfun” e “morto” que estejam ou estiveram relacionados com a função de vendedor em ordem alfabética ascendente pelo campo NOME.

Resposta: (SELECT * FROM cadfun WHERE FUNCAO = 'VENDEDOR') UNION (SELECT * FROM morto WHERE FUNCAO = 'VENDEDOR') ORDER BY NOME;

10. Liste a quantidade de funcionários por departamento que tenham como sobrenome SILVA, agrupando a informação por departamento.

Resposta: SELECT DEPTO, COUNT(*) FROM cadfun WHERE NOME LIKE '%SILVA' GROUP BY DEPTO;

CAPÍTULO 7

1. O que é junção de tabelas?

Resposta: Uma junção de tabela possibilita, por meio de um campo comum entre as tabelas existentes, gerar consultas como se as tabelas relacionadas (múltiplas tabelas) na consulta fossem uma só.

2. O que é visualização de tabelas?

Resposta: Uma visualização é uma tabela virtual baseada nos dados de uma tabela real. A partir dos dados de uma única tabela real é possível ter várias visões para facilitar as operações de consultas.

3. Quantas tabelas são necessárias, no mínimo, para estabelecer uma relação de junção?

Resposta: Para determinar o relacionamento entre tabelas por meio de junções, é necessário ter no mínimo duas tabelas que possuam algum campo em comum.

4. Liste por junção os nomes dos clientes, os estados dos clientes, os números das duplicatas e seus respectivos valores para todos os clientes residentes no estado de SP.

Resposta: SELECT cliente.NOME, cliente.ESTADO, venda.DUPLIC, venda.VALOR FROM cliente, venda WHERE (cliente.CODCLI = venda.CODCLI) AND (cliente.ESTADO = 'SP') ORDER BY cliente.NOME;

5. Liste por junção os títulos em carteira do cliente MICROS INFORMATICA S/A.

Resposta: SELECT cliente.NOME, venda.DUPLIC, venda.VALOR FROM cliente, venda WHERE cliente.CODCLI = venda.CODCLI AND cliente.NOME = 'MICROS INFORMATICA S/A';

6. Liste por junção os nomes de todos os clientes que possuam títulos com vencimento no mês de abril de qualquer ano.

Resposta: `SELECT cliente.NOME FROM cliente, venda WHERE (cliente.CODCLI = venda.CODCLI) AND (MONTH(venda.VENCTO) = 4);`

7. Apresente por junção a quantidade de títulos por cliente para qualquer data de vencimento. Faça o agrupamento do nome de cliente.

Resposta: `SELECT cliente.NOME, COUNT(*) FROM cliente, venda WHERE cliente.CODCLI = venda.CODCLI GROUP BY cliente.NOME;`

8. Apresente por junção a quantidade de títulos por cliente agrupados por data de vencimento.

Resposta: `SELECT cliente.NOME, venda.VENCTO, COUNT(*) FROM cliente, venda WHERE cliente.CODCLI = venda.CODCLI GROUP BY venda.VENCTO;`

9. Apresente por junção a quantidade de títulos agrupados por ordem de ano para qualquer vencimento.

Resposta: `SELECT YEAR(venda.VENCTO), COUNT(*) FROM cliente, venda WHERE cliente.CODCLI = venda.CODCLI GROUP BY YEAR(venda.VENCTO);`

10. Apresente por junção a quantidade de títulos agrupados por ordem de mês e ano para qualquer vencimento.

Resposta: `SELECT MONTH(venda.VENCTO), YEAR(venda.VENCTO), COUNT(*) FROM cliente, venda WHERE cliente.CODCLI = venda.CODCLI GROUP BY YEAR(venda.VENCTO), MONTH(venda.VENCTO);`

11. Apresente por junção os nomes, os valores e as datas de vencimento dos títulos em atraso entre 1º de janeiro de 2000 e 31 de dezembro de 2003. Ordenar sem agrupamento a consulta por nome de cliente.

Resposta: `SELECT cliente.NOME, venda.VALOR, venda.VENCTO FROM cliente, venda WHERE cliente.CODCLI = venda.CODCLI AND VENCTO BETWEEN '2000-01-01' AND '2003-12-31' ORDER BY cliente.NOME;`

12. Apresente por junção os nomes e a quantidade de títulos em atraso por cliente que estejam vencidos entre 1º de janeiro de 2000 e 31 de dezembro de 2003. Agrupe por nome de cliente.

Resposta: `SELECT cliente.CODCLI, cliente.NOME, COUNT(*) FROM cliente, venda WHERE cliente.CODCLI = venda.CODCLI AND VENCTO BETWEEN '2000-01-01' AND '2003-12-31' GROUP BY cliente.NOME;`

13. Apresente por junção os nomes e a quantidade de títulos em atraso e o valor total dos títulos por cliente que estejam vencidos entre 1º de janeiro de 2000 e 31 de dezembro de 2003. Agrupe por nome de cliente.

Resposta: SELECT cliente.CODCLI, cliente.NOME, COUNT(*), SUM(venda.VALOR) FROM cliente, venda WHERE cliente.CODCLI = venda.CODCLI AND VENCTO BETWEEN '2000-01-01' AND '2003-12-31' GROUP BY cliente.NOME;.

14. Apresente por junção os nomes e a quantidade de títulos em atraso e o valor total dos títulos por cliente que estejam vencidos entre as datas de 1º de janeiro de 2000 e 31 de dezembro de 2003 dos clientes residentes na cidade de SANTOS. Agrupe por nome de cliente.

Resposta: SELECT cliente.CODCLI, cliente.NOME, COUNT(*), SUM(venda.VALOR) FROM cliente, venda WHERE cliente.CODCLI = venda.CODCLI AND VENCTO BETWEEN '2000-01-01' AND '2003-12-31' AND cliente.CIDADE = 'SANTOS' GROUP BY cliente.NOME;.

15. Apresente por junção os nomes, as cidades, os estados e o valor total dos títulos em atraso por cliente que estejam vencidos entre as datas de 1º de janeiro de 2000 e 31 de dezembro de 2003 dos clientes residentes nas cidades de SANTOS, SANTO AMARO e OSASCO. Agrupe por nome de cliente.

Resposta: SELECT cliente.NOME, cliente.CIDADE, cliente.ESTADO, COUNT(*), SUM(venda.VALOR) FROM cliente, venda WHERE cliente.CODCLI = venda.CODCLI AND VENCTO BETWEEN '2000-01-01' AND '2003-12-31' AND cliente.CIDADE IN ('SANTOS', 'SANTO AMARO', 'OSASCO') GROUP BY cliente.NOME;.

CAPÍTULO 8

1. O que é índice de uma tabela?

Resposta: O índice é uma estrutura de arquivo complementar que armazena o campo chave de uma determinada tabela e o local onde aquele campo é encontrado na tabela. Segundo STEPHENS & PLEW (2003), "um índice em um banco de dados é muito semelhante ao índice na parte de trás de um livro", ou seja, semelhante ao índice remissivo. No índice remissivo é possível localizar uma palavra-chave e a página em que se encontra o assunto relacionado àquela palavra-chave selecionada.

2. Como podem ser classificados os índices de uma tabela?

Resposta: Os índices de uma tabela podem ser classificados em simples, exclusivo e composto.

3. Descreva índice simples de tabela.

Resposta: O índice simples baseia-se na definição do valor de apenas uma coluna (campo).

4. Descreva índice exclusivo de tabela.

Resposta: O índice exclusivo baseia-se na definição de um valor que não pode ser repetido dentro da tabela.

5. Descreva índice composto de tabela.

Resposta: O índice composto baseia-se em valores de mais de uma coluna.

6. Qual é a sintaxe do comando de definição de um índice de tabela?

Resposta: CREATE [UNIQUE] INDEX <índice> ON <tabela (campos)>;

7. Qual é sintaxe do comando de remoção de um índice de tabela?

Resposta: DROP INDEX <índice> ON <tabela>;

8. Onde um índice deve ser evitado?

Resposta: Um índice deve ser evitado em tabelas pequenas ou em colunas (campos) que tenham um grande número de valores NULL. Deve-se também evitar o uso de índices em colunas que sofrem alto impacto de atualização, pois degrada a velocidade de acesso aos dados.

9. Quais são as classificações de usuários permitidas no programa MySQL?

Resposta: O usuário do programa MySQL pode estar classificado segundo quatro características de acordo com a documentação do programa:

- **Nível global** - este nível de privilégio permite o acesso do usuário a todos os bancos de dados em um determinado servidor. Os privilégios desta categoria são armazenados na tabela mysql.user;
- **Nível de bancos de dados** - permite o acesso do usuário apenas às tabelas de um determinado banco de dados. Os privilégios desta categoria são armazenados nas tabelas mysql.db e mysql.host.
- **Nível de tabelas** - permite o acesso do usuário apenas às colunas de uma determinada tabela. Os privilégios desta categoria são armazenados na tabela mysql.tables_priv.
- **Nível de colunas** - este nível de privilégio permite o acesso do usuário apenas a uma única coluna em uma determinada tabela. Os privilégios desta categoria são armazenados na tabela mysql.columns_priv.

10. Qual é o comando que revoga a definição de privilégios de um determinado usuário?

Resposta: Usa-se o comando REVOKE.

11. Qual é a finalidade do comando GRANT?

Resposta: Sua finalidade é cadastrar usuários no programa MySQL.

12. Qual é a finalidade do comando SHOW GRANTS FOR?

Resposta: Apresentar as informações de privilégios de um determinado usuário.

13. Qual é o motivo de utilizar um campo (coluna) de uma tabela como chave primária?

Resposta: O campo de chave primária é usado para evitar a duplicidade de registros.

14. O que é chave candidata?

Resposta: É o campo que armazena um determinado valor que poderia ser considerado campo de chave primária. Normalmente se utiliza um campo de chave candidata para armazenar valores como CPF e CNPJ, que são exclusivos para cada pessoa física ou jurídica e não podem ser repetidos para mais de uma entidade de registro.

15. O que é chave estrangeira?

Resposta: A chave estrangeira dá a possibilidade de estabelecer o relacionamento lógico de uma determinada tabela (em que se tem definido um campo chave) com o campo de chave primária de outra tabela.

16. Quais formas de cardinalidade podem ser utilizadas com o programa MySQL?

Resposta: Existem três formas de relacionamentos (cardinalidade) que podem ser utilizadas, sendo:

- Relacionamento de 1 para 1;
- Relacionamento de 1 para N ou de N para 1 quando for o caso;
- Relacionamento de N para N.

17. O que é cardinalidade 1:1?

Resposta: O relacionamento de 1 para 1 (um para um) representado pela cardinalidade (1:1) é a relação de um campo chave de um registro de uma determinada tabela vinculado com um campo chave de um outro registro de outra tabela.

18. O que é cardinalidade 1:N ou N:1?

Resposta: O relacionamento de 1 para N (um para muitos) representado pela cardinalidade (1:N) é a relação de um campo chave de um registro de uma determinada tabela vinculado com o campo chave de muitos registros de outra tabela.

19. O que é cardinalidade N:N?

Resposta: O relacionamento de N para N (muitos para muitos) representado pela cardinalidade (N:N) é a relação de um campo chave de muitos registros de uma determinada tabela vinculado com o campo chave de muitos registros de outra tabela.

20. É correto afirmar que a chave estrangeira no programa MySQL só pode ser usada com tabelas do tipo MyISAM? Justifique sua resposta.

Resposta: É incorreto fazer tal afirmação, pois o formato de tabelas padrão do MySQL é o MyISAM e não dá suporte direto ao uso de chaves estrangeiras. Para usar tal recurso é necessário usar o formato InnoDB que é assumido automaticamente quando da definição de chave estrangeira.

21. Qual cláusula se usa para definir chave estrangeira numa tabela MySQL?

Resposta: Usa-se a cláusula FOREIGN KEY.

22. O uso de chave primária x chave estrangeira é a mesma coisa que usar junções de tabelas?

Resposta: Os dois conceitos são muito diferentes. A relação chave primária x chave estrangeira é realizada por vínculos explicitamente definidos entre os campos das tabelas envolvidas na relação. Já uma junção é uma relação estabelecida de forma lógica no momento de uma determinada consulta.