**Exercícios – Modelo ER / SQL**

Todos as questões abaixo são baseadas no PDF do Modelo ER, portanto é necessário ter esse modelo para poder realizar esse exercício.

1. A tabela TB\_ALUNO possui os registros abaixo.

|  |  |  |
| --- | --- | --- |
| CODALU | NOMEALU | EMAIL |
| 1 | João da Silva | [jsilva@gmail.com](mailto:jsilva@gmail.com) |
| 2 | Maria Pereira | *NULL* |
| 3 | Pedro Pedreira | [pedro@uol.com.br](mailto:pedro@uol.com.br) |
| 4 | Carlos Nascimento | [carlosn@gmail.com](mailto:carlosn@gmail.com) |
| 5 | Felipe Matos | *NULL* |

Faça um SQL que liste somente os alunos que possuem e-mail preenchido.

|  |
| --- |
| [SELECT](http://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/8.0/en/select.html) \* FROM `tb\_alunos` WHERE EMAIL [LIKE](http://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/8.0/en/string-comparison-functions.html#operator_like) "%@%" |

1. Utilizando a mesma tabela da questão 1, faça um SQL que liste somente as pessoas que possuem o e-mail com domínio @gmail.com

|  |
| --- |
| [SELECT](http://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/8.0/en/select.html) \* FROM `tb\_alunos` WHERE EMAIL [LIKE](http://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/8.0/en/string-comparison-functions.html#operator_like) "%@gmail%" |

1. O aluno João da Silva está matriculado nos cursos de Administração e Direito. Um SQL foi utilizado para listar a seguinte tabela:

|  |  |  |
| --- | --- | --- |
| CODALU | NOMEALU | NOMECURSO |
| 1 | João da Silva | Administração |
| 1 | João da Silva | Direito |

Qual foi o SQL utilizado para listar a informação acima? Escreva abaixo.

|  |
| --- |
| [SELECT](http://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/8.0/en/select.html) \* FROM `tb\_alunos` WHERE CODALU = 1 |

1. Os alunos João da Silva e Felipe Matos estão fazendo o curso de Direito. Isso significa que na tabela TB\_CURSO deverão ter dois registros pra esse curso, conforme abaixo:

|  |  |
| --- | --- |
| CODCURSO | NOMECURSO |
| 1 | Direito |
| 2 | Direito |

A afirmação acima é (assinale com X):

|  |  |
| --- | --- |
| X | Verdadeira |
|  | Falsa |

1. A tabela TB\_MATRICULA é utilizada para armazenar cada uma das matrículas dos alunos em cada curso no período letivo. Considerando que existem 5 alunos cadastrados e cada aluno está fazendo 2 cursos, quantos registros haverão na tabela TB\_MATRICULA para um único período letivo.

Assinale a alternativa correta:

|  |  |
| --- | --- |
|  | 2 |
|  | 5 |
| x | 10 |
|  | 20 |

1. Um SQL foi feito para listar todas as matrículas dos alunos que possuem código inferior a 4. O resultado foi a tabela abaixo:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| CODALU | NOMEALU | NOMECURSO | NOMETURMA | DESCPER |
| 1 | João da Silva | Administração | ADM01 | 2020/1 |
| 1 | João da Silva | Direito | DIR03 | 2020/1 |
| 2 | Marcia Pereira | Administração | ADM02 | 2019/2 |
| 3 | Pedro Pedreira | Medicina | MED08 | 2019/2 |

Qual foi o SQL utilizado para listar a informação acima? Escreva abaixo.

|  |
| --- |
| [SELECT](http://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/8.0/en/select.html) \* FROM `tb\_alunos` WHERE CODALU < 4 |

1. Os comandos abaixo representam a inserção de 4 disciplinas na tabela TB\_DISCIPLINA.

INSERT INTO TB\_DISCIPLINA (CODDIS, NOMEDIS, CHDIS) VALUES (1,'Estatística',60)

INSERT INTO TB\_DISCIPLINA (CODDIS, NOMEDIS, CHDIS) VALUES (2,'Cálculo I',60)

INSERT INTO TB\_DISCIPLINA (CODDIS, NOMEDIS, CHDIS) VALUES (2,'Matemática',40)

INSERT INTO TB\_DISCIPLINA (CODDIS, NOMEDIS, CHDIS) VALUES (3,'Contabilidade I',60)

Entretanto, somente 3 disciplinas serão inseridas. Qual disciplina não será inserida? Justifique.

|  |
| --- |
| INSERT INTO TB\_DISCIPLINA (CODDIS, NOMEDIS, CHDIS) VALUES (2,'Matemática',40)  Não seria inserida, pois ,estaria repetindo o CODDIS, visto que ele seria uma chave primaria. |

1. Considerando que a tabela TB\_ALUNO possui 4 registros conforme abaixo:

|  |  |
| --- | --- |
| CODALU | NOMEALU |
| 1 | João da Silva |
| 2 | Maria Pereira |
| 3 | Pedro Pedreira |
| 4 | Carlos Nascimento |

E considerando que o aluno 3 não tem nenhuma matrícula em curso, como fazer para listar a tabela abaixo:

|  |  |  |
| --- | --- | --- |
| CODALU | NOMEALU | NOMECURSO |
| 1 | João da Silva | Administração |
| 2 | Maria Pereira | Direito |
| 3 | Pedro Pedreira | *NULL* |
| 4 | Carlos Nascimento | Enfermagem |

Escreva abaixo o SQL para listar a tabela acima.

|  |
| --- |
| SELECT `CODALU`, `NOMEALU`, `NOMECURSO` FROM `tb\_alunos` |

1. A tabela TB\_MATRICULA\_DISCIPLINA armazena as disciplinas que o aluno está matriculado. Essa tabela possui os seguintes registros:

|  |  |  |
| --- | --- | --- |
| CODMD | CODMAT | CODDIS |
| 1 | 1 | 1 |
| 2 | 1 | 1 |
| 3 | 1 | 2 |
| 4 | 2 | 2 |
| 5 | 2 | 1 |
| 6 | 3 | 3 |

Caso seja necessário listar uma informação mais sintética, é possível totalizar os registros de forma agrupada. O resultado abaixo demonstra de forma agrupada quantas matriculas existem em cada disciplina.

|  |  |
| --- | --- |
| CODDIS | TOTAL |
| 1 | 3 |
| 2 | 2 |
| 3 | 1 |

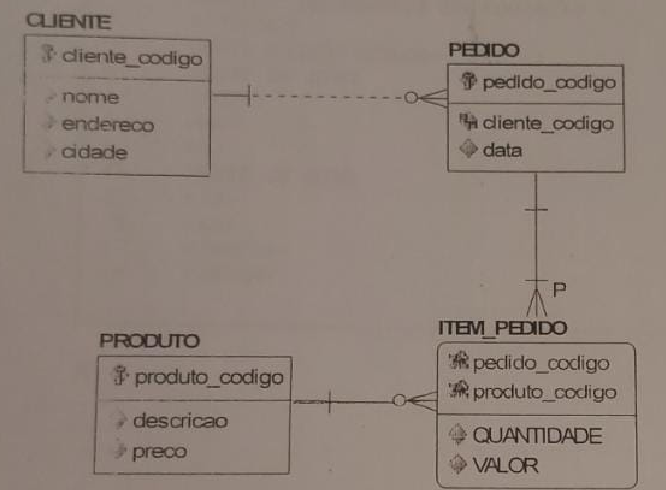
Qual o SQL que foi utilizado para listar a informação acima de forma agrupada?

|  |
| --- |
| SELECT CODDIS, SUM(CODMAT) As TOTAL FROM TB\_MATRICULA\_DISCIPLINA GROUP BY CODDIS; |

1. Caso você precisar excluir um registro de aluno da tabela TB\_ALUNO e esse aluno já tiver matrículas e disciplinas matriculadas, você deverá excluir os registros em uma ordem hierárquica. Qual a ordem correta para excluir um aluno que já possui registros de disciplinas matriculadas:

Assinale a alternativa correta:

|  |  |
| --- | --- |
|  | TB\_DISCIPLINA **>** TB\_MATRICULA\_DISCIPLINA **>** TB\_MATRICULA **>** TB\_ALUNO\_CURSO **>** TB\_ALUNO |
|  | TB\_MATRICULA **>** TB\_MATRICULA\_DISCIPLINA **>** TB\_ALUNO\_CURSO **>** TB\_ALUNO |
| X | TB\_MATRICULA\_DISCIPLINA **>** TB\_MATRICULA **>** TB\_ALUNO\_CURSO **>** TB\_ALUNO |
|  | TB\_ALUNO\_CURSO **>** TB\_MATRICULA **>** TB\_MATRICULA\_DISCIPLINA **>** TB\_DISCIPLINA **>** TB\_ALUNO |
|  | TB\_ALUNO **>** TB\_ALUNO\_CURSO **>** TB\_MATRICULA **>** TB\_MATRICULA\_DISCIPLINA |



1. Relação de clientes com código, nome, endereço. Ordenados por cidade e nome;

|  |
| --- |
| SELECT `nome`, `cidade` FROM `CLIENTE` |

1. Relação de produtos em ordem decrescente de preço;

|  |
| --- |
| Select \* from PRODUTO  order by preco desc |

1. Relação de pedidos efetuados pelo cliente (cliente\_codigo = 15). Retornar nome do cliente, data do pedido e valor total;

|  |
| --- |
| SELECT CLIENTE.nome, PEDIDO.data, ITEM\_PEDIDO.quantidade, ITEM\_PEDIDO.valor  FROM CLIENTE INNER JOIN PEDIDO ON (CLIENTE.cliente\_codigo=PEDIDO.cliente\_codigo)  INNER JOIN ITEM\_PEDIDO ON (ITEM\_PEDIDO.pedido\_codigo=PRODUTO.pedido\_codigo)  INNER JOIN PRODUTO ON (PRODUTO.produto\_codigo=ITEM\_PEDIDO.produto\_codigo)  WHERE CLIENTE.codigo\_cliente=15 |

1. Todos os clientes que fizeram o pedido para o produto de código = 1276, ordenando pelo nome do cliente

|  |
| --- |
| SELECT CLIENTE.nome  FROM CLIENTE INNER JOIN PEDIDO ON (CLIENTE.cliente\_codigo=PEDIDO.cliente\_codigo)  INNER JOIN ITEM\_PEDIDO ON (ITEM\_PEDIDO.pedido\_codigo=PRODUTO.pedido\_codigo)  INNER JOIN PRODUTO ON (PRODUTO.produto\_codigo=ITEM\_PEDIDO.produto\_codigo)  WHERE PRODUTO.produto\_codigo =1276 |

1. Relação dos 10 produtos mais pedidos. Trazer código, descrição e total pedido.

|  |
| --- |
| SELECT PRODUTO.produto\_codigo, PRODUTO.descricao,ITEM\_PEDIDO.qunatidade FROM PRODUTO  INNER JOIN ITEM\_PEDIDO ON (ITEM\_PEDIDO.pedido\_codigo=PRODUTO.pedido\_codigo)  order by ITEM\_PEDIDO.quantidade LIMIT 10 |

1. Relação de clientes (futuros clientes) que não possuem pedidos efetuados. Trazer código e nome do cliente.

|  |
| --- |
| SELECT CLIENTE.cliente\_codigo, CLIENTE.nome  FROM CLIENTE  WHERE CLIENTE.cliente\_codigo NOT IN (SELECT PEDIDOS.codigo\_cliente  FROM PEDIDOS |