**EXERCÍCIOS PRÁTICOS**

**Banco de Dados**

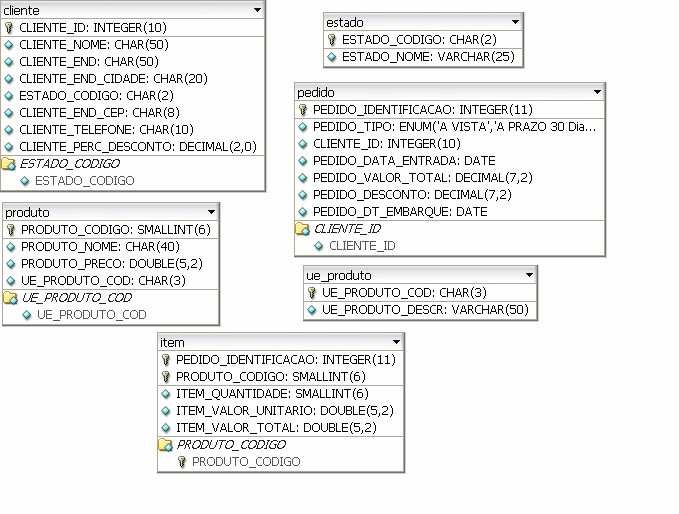
Introdução ao uso do SQL – *Structured Query Language*, considerando tanto DDL – *Data Definition Language* quanto DML – *Data Manipulation Language*.

Banco de Dados selecionado: MySQL

Para a execução dos comandos necessários pode-se utilizar um arquivo com extensão sql. Os tipos de comandos que podem ser disponibilizados em tal arquivo estão identificados nos exemplos abaixo.

***1.1. Criação de um Banco de Dados***

# 1.2. Exemplo



# 1.2. Aplicação de comando SQL-DDL

Considerando a sintaxe descrita para as operações SQL na parte 2 deste documento, para as tabelas mostradas abaixo realize, sobre o BD criado, seguintes atividades:



1) Criar as tabelas acima usando o comando CREATE TABLE (criação de uma nova tabela). Entretanto, não defina ainda as colunas relacionadas com as PK (*Primary Key* – identificador único de cada registro da tabela) e FK (*Foreign Key* – importada de outra tabela, para relacionar tabelas) em ambas tabelas.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

CREATE TABLE PRODUTO\_CC\_6A

( produto\_cc\_6A\_nome CHAR(40),

produto\_cc\_6A\_preco DOUBLE(5,2) )

ou

CREATE TABLE PRODUTO\_CC\_6A

( produto\_cc\_6A\_codigo VARCHAR(2) NOT NULL, produto\_cc\_6A\_nome CHAR(40), produto\_cc\_6A\_preco DECIMAL(5,2), PRIMARY KEY (produto\_cc\_6A\_codigo) )

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

CREATE TABLE UE\_PRODUTO\_CC\_6A

( ue\_produto\_cc\_6A\_descr CHAR(50) )

Ou

CREATE TABLE UE\_PRODUTO\_CC\_6A

( ue\_produto\_cc\_6A\_codigo VARCHAR(3) NOT NULL,

ue\_produto\_cc\_6A\_descr CHAR(50) )

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

2) Criar a chave primária para as duas tabelas através do comando ALTER TABLE (alteração de tabela). Inserir cada uma dessas PK na primeira posição de cada tabela.

1. PK de UE\_PRODUTO\_CC\_6A = ue\_produto\_cc\_6A\_codigo CHAR(3) NOT NULL
2. PK de PRODUTO\_CC\_6A = produto\_cc\_6A\_codigo SMALLINT NOT NULL

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

ALTER TABLE UE\_PRODUTO\_CC\_6A

ADD COLUMN ue\_produto\_cc\_6A\_codigo CHAR(3) NOT NULL FIRST;

ALTER TABLE UE\_PRODUTO\_CC\_6A

ADD PRIMARY KEY (ue\_produto\_cc\_6A\_codigo);

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

ALTER TABLE PRODUTO\_CC\_6A

ADD COLUMN produto\_cc\_6A\_codigo SMALLINT NOT NULL FIRST;

ALTER TABLE PRODUTO\_CC\_6A

ADD PRIMARY KEY (produto\_cc\_6A\_codigo);

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

1. Criar a chave estrangeira da tabela PRODUTO\_CC\_6A que aponta para a tabela

UE\_PRODUTO\_cc\_6A, analisando a ligação demonstrada no modelo com Integridade Referencial CASCADE.

* 1. A FK de PRODUTO\_CC\_6A = ue\_produto\_cc\_6A\_codigo CHAR(3) NOT NULL;
  2. Essa chave aponta para o campo ue\_produto\_cc\_6A\_codigo da tabela UE\_PRODUTO\_ CC\_6A

A FK deve ser inserida na última posição do tabela PRODUTO\_cc\_6a.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

ALTER TABLE PRODUTO\_CC\_6A ADD COLUMN

ue\_produto\_cc\_6A\_codigo CHAR(3) NOT NULL AFTER produto\_cc\_6A\_preco;

ALTER TABLE PRODUTO\_CC\_6a

ADD FOREIGN KEY (ue\_produto\_cc\_6A\_codigo)

REFERENCES UE\_PRODUTO\_cc\_6A(ue\_produto\_cc\_6A\_codigo)

ON UPDATE CASCADE;

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

1. Criar um Índice único na tabela PRODUTO\_CC\_6A para a coluna produto\_nome\_CC\_6a

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

CREATE UNIQUE INDEX indice\_produto\_nome ON PRODUTO\_CC\_6A (produto\_cc\_6A\_nome(5));

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

**Para alterar uma coluna de uma tabela:**

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

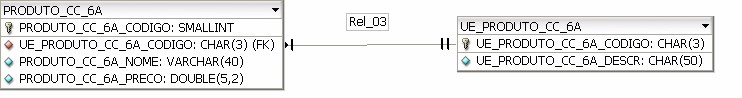
ALTER TABLE ue\_produto\_cc\_6a

CHANGE COLUMN ue\_produto\_cc\_6A\_codigo

ue\_produto\_cc\_6A\_codigo CHAR(3) NOT NULL;



OBS: A execução das tarefas 2 e 3 sobre as tabelas acima deve ser a modificação das mesmas para a seguinte estrutura:



# 1.3. Aplicação de comando SQL-DML

**Observação**:

Antes e depois de executar qualquer comando de modificação (INSERT, UPDATE ou DELETE), verifique o conteúdo da tabela para se certificar de que a modificação tenha sido executada com sucesso.

Para os comandos INSERT, UPDATE E DELETE, utilizaremos as tabelas de apoio criadas acima (PRODUTO\_CC\_6A e EU\_PRODUTO\_CC\_6A). Para o comando SELECT, utilizaremos as seguintes tabelas: CLIENTE, ESTADO, ITEM, PEDIDO, PRODUTO, UE\_PRODUTO, conforme modelo de dados definido pelo modelo apresentado em primeiro neste documento. Essas tabelas devem ser populadas com alguns dados de teste.

**1.3.1. INSERT (inserção de um novo registro em uma tabela)**

Executar inicialmente o seguinte comando SQL:

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

INSERT INTO UE\_PRODUTO\_CC\_6A (ue\_produto\_cc\_6a\_codigo, ue\_produto\_cc\_6a\_descr)

SELECT ue\_produto\_cod, ue\_produto\_descr

FROM UE\_PRODUTO;

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Este comando faz uma cópia de cada registro da tabela UE\_PRODUTO e o insere na tabela UE\_PRODUTO\_CC\_6A.

Para os exercícios abaixo, utilize o comando INSERT do SQL, o qual tem a sintaxe geral mostrada abaixo na forma de um exemplo:

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

INSERT INTO <nome da tabela> (<colunas nas quais os valores serão inseridos separadas por vírgulas>)

VALUES (<valores para cada coluna listada);

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Exemplo:

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

INSERT INTO PRODUTO\_CC\_6A (produto\_cc\_6a\_codigo, ue\_produto\_cc\_6a\_codigo) VALUES (112, ‘MI’);

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Agora, ao invés de copiar os valores de uma outra tabela, eles foram informados diretamente.

1. Inserir uma linha na tabela PRODUTO\_CC\_6A informando todas as colunas, conforme formato geral.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

INSERT INTO PRODUTO

VALUES (145,'Farinha',13.00,'Kg');

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

1. Inserir uma linha na tabela PRODUTO\_CC\_6A informando somente as colunas obrigatórias (aquelas que não podem ser vazias).

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

INSERT INTO PRODUTO\_CC\_6A (produto\_cc\_6a\_codigo, ue\_produto\_cc\_6a\_codigo)

VALUES (112, ‘MI’);

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

1. Limpar toda a tabela PRODUTO\_CC\_6ª (isto é, excluir todos os seus registros), para poder executar o comando do item 4.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

DELETE FROM PRODUTO\_CC\_6A;

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Popular toda a tabela PRODUTO\_CC\_6A a partir de um comando SELECT (seleção) na tabela PRODUTO.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

INSERT INTO PRODUTO\_CC\_6a (produto\_cc\_6a\_nome, produto\_cc\_6a\_preco, produto\_cc\_6a\_codigo, ue\_produto\_cc\_6a\_codigo)

SELECT produto\_nome, produto\_preco, produto\_codigo, ue\_produto\_cod

FROM produto;

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

1.3.2. UPDATE (atualização)

Este comando altera os registros das tabelas do Banco de Dados. Sua sintaxe pode ser resumida com o seguinte exemplo:

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

UPDATE <nome da tabela a ser alterada>

SET <coluna a ser alterada> = <novo valor>

WHERE <condiçao de seleção do registro a ser alterado>;



\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Exemplo:



\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

UPDATE produto\_cc\_6a

SET produto\_cc\_6a\_preco = 12

WHERE produto\_cc\_6a\_preco < 10;

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Este exemplo atualiza a tabela PRODUTO\_CC\_6A, a coluna PRODUTO\_CC\_6A\_PRECO com o valor 12, para aqueles registros onde o valor da coluna PRODUTO\_CC\_6A\_PRECO sao menor do que 10.

1. Alterar a tabela PRODUTO\_CC\_6A, aplicando um aumento de 50% sobre o preço, somente para os produtos cujo preço for menor que 10.00.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

UPDATE produto\_cc\_6a

SET produto\_cc\_6a\_preco = produto\_cc\_6a\_preco \*1.5

WHERE produto\_cc\_6a\_preco < 10;

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

1. Alterar a tabela PRODUTO\_CC\_6A, somente para o produto cujo código = 3, mudando o nome para “Feijão Preto e Branco”.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

UPDATE produto\_cc\_6a

SET produto\_cc\_6a\_nome = ‘Feijão Preto e Branco’

WHERE produto\_cc\_6a\_codigo = 3;

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

1. Alterar o preço dos produtos na tabela PRODUTO\_CC\_6A dando um desconto de 40%, somente para os produtos cujo código da unidade de estoque (ue\_produto\_descr\_CC\_6A) seja = 5.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

UPDATE produto\_cc\_6a

SET produto\_cc\_6a\_preco = produto\_cc\_6a\_preco \*0.6

WHERE ue\_produto\_cc\_6a\_codigo = 5;

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

1. Alterar a descrição da unidade de estoque para “inutilizado” somente para as unidades de estoque para as quais não exista nenhum produto associado. (\*\*\*\*\*)

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

UPDATE ue\_produto\_cc\_6a

SET ue\_produto\_cc\_6a\_descr = ‘inutilizado’

WHERE ue\_produto\_cc\_6a\_codigo NOT IN (SELECT DISTINCT

ue\_produto\_cc\_6a\_codigo from produto\_cc\_6a);

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Observar que neste caso a cláusula de condição questionou se o valor da coluna UE\_PRODUTO\_CC\_6A\_CODIGO não estava (NOT IN) numa seleção de valores da coluna UE\_PRODUTO\_CC\_6A\_CODIGO diferentes da tabela PRODUTO\_CC\_6A.

## 1.3.3. DELETE (deleção de registro)

1. Excluir da tabela PRODUTO\_CC\_6A o produto cujo código seja igual a 20.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

DELETE FROM produto\_cc\_6a

WHERE ue\_produto\_cc\_6a\_codigo = 20;

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

1. Excluir na tabela PRODUTO\_CC\_6A somente os produtos cuja descrição da unidade de estoque (ue\_produto\_cc\_6a\_descr) seja = “litro”.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

DELETE FROM produto\_cc\_6a

WHERE ue\_produto\_cc\_6a\_codigo IN (SELECT ue\_produto\_cc\_6a\_codigo FROM ue\_produto\_cc\_6a WHERE ue\_produto\_cc\_6a\_descr = 'LITRO');

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

## 1.3.4. SELECT (seleção de registro)

1. Selecionar todas as colunas (\*) de todos os clientes da tabela CLIENTE.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

SELECT \*

FROM cliente;

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

1. Selecionar somente a coluna nome de todos os produtos.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

SELECT produto\_nome

FROM produto;

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

1. Selecionar as seguintes colunas dos clientes de maneira distinta: cidade, estado e cep .

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

SELECT DISTINCT cliente\_end\_cidade,estado\_codigo, cliente\_end\_cep

FROM cliente;

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

1. Selecionar todos os pedidos para o cliente cujo código seja = 09 e cujo valor total seja maior que

50.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

SELECT \*

FROM pedido

WHERE cliente\_id= 09 AND pedido\_valor\_total > 50.0;

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

1. Selecionar todos os pedidos cujo valor total seja menor que 100 ou maior que 500.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

SELECT \*

FROM pedido

WHERE pedido\_valor\_total < 100 OR pedido\_valor\_total > 500;

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

1. Selecionar todos os pedidos cuja data de entrada seja = 04/12/1999.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

SELECT \*

FROM pedido

WHERE pedido\_data\_entrada = ‘99/12/04’; (‘1999/12/04’ ou ‘99-12-04’)

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

1. Selecionar todos os pedidos cuja data de entrada seja = 02/12/1999 e cujo valor total esteja entre 20 e 50.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

SELECT \*

FROM pedido

WHERE (pedido\_valor\_total BETWEEN 20 AND 50)

AND pedido\_data\_entrada = ‘99/12/02’;

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

SELECT \*

FROM pedido

WHERE pedido\_valor\_total > 20 AND pedido\_valor\_total < 50

AND pedido\_data\_entrada = ‘99/12/02’;

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

8. Selecionar todos os clientes cujo código NÃO esteja entre 05 e 25.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

SELECT \*

FROM cliente

WHERE cliente\_id NOT BETWEEN 5 AND 25;

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* ou

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

SELECT \*

FROM cliente

WHERE cliente\_id > 25 OR cliente\_id < 5;

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

1. Selecionar todos os produtos cujo nome possua a primeira letra = ‘P’.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

SELECT \*

FROM produto

WHERE produto\_nome LIKE 'P%';

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

1. Selecionar todos os produtos cujo nome possua a string 'an' em qualquer posição do nome.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

SELECT \*

FROM produto

WHERE produto\_nome LIKE '%an%';

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

1. Selecionar todos os produtos cujo nome comece com C ou F ou M, independente do resto.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

SELECT \*

FROM produto

WHERE produto\_nome LIKE 'C%'

OR produto\_nome LIKE 'F%'

OR produto\_nome LIKE 'M%' ;

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

1. Selecionar todos os produtos cujo nome possua 8 caracteres, onde as duas primeiras letras sejam iguais a ‘ma’, independente do conteúdo das 6 próximas letras.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

SELECT \*

FROM produto

WHERE produto\_nome LIKE 'ma\_\_\_\_\_\_';

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

SELECT \*

FROM produto

WHERE produto\_nome LIKE ‘ma%’ AND LENGTH(produto\_nome) = 8;

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

1. Selecionar todos os produtos cujo nome possua 7 caracteres e as duas primeiras letras não interessem, as 3 próximas sejam iguais a ‘ACA’ e as duas ultimas também não interessem.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

SELECT \*

FROM produto

WHERE produto\_nome LIKE '\_\_ACA\_\_';

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

1. Selecionar todos os Produtos cujo Nome possua a string 'a\_p' como parte do nome do produto, em qualquer posição do nome.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

SELECT \*

FROM produto

WHERE produto\_nome LIKE '%a\_p%';

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Selecionar todos os clientes cuja UF seja ‘MG’ ou ‘ES’.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

SELECT \*

FROM cliente

WHERE estado\_codigo = 'MG'

OR estado\_codigo = 'ES'; \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* ou

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

SELECT \*

FROM cliente

WHERE estado\_codigo IN('MG','ES');

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

16. Selecionar todos os clientes cujo estado NAO seja RJ nem SP.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

SELECT \*

FROM cliente

WHERE estado\_codigo <> 'RJ'

AND estado\_codigo <> 'SP';

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Ou

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

SELECT \*

FROM cliente

WHERE estado\_codigo NOT IN ('RJ','SP');

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

1. Selecionar todos os produtos cujo preço seja menor que 20 e a unidade de estoque seja

‘kilograma’ ou ‘litro’.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

SELECT \*

FROM produto

WHERE produto\_preco < 20

AND ue\_produto\_cod IN (SELECT ue\_produto\_cod FROM ue\_produto

WHERE ue\_produto\_descr = 'kilograma' OR ue\_produto\_descr = 'litro');

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

1. Selecionar todos os produtos cuja unidade de estoque seja ‘KG’ ou ‘UM’ e o preço seja maior que 10.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

SELECT \*

FROM produto

WHERE produto\_preco > 10

AND (ue\_produto\_codigo = 'KG' OR ue\_produto\_codigo = 'UM');

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

1. Selecionar todos os produtos cujo preço seja menor que 5, incluindo os produtos cujo preço esteja nulo.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

SELECT \*

FROM produto

WHERE produto\_preco < 5 OR produto\_preco is null;

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

1. Selecionar o nome e telefone dos clientes cujo código do cliente seja menor que 10 colocando um novo título (rótulo) para cada uma das colunas exibidas no resultado.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

SELECT cliente\_nome AS cliente, cliente\_telefone AS telefoe

FROM cliente

WHERE cliente\_id < 10;

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

1. Selecionar todos os produtos, demonstrando, o código e o nome do produto, o preço atual de cada produto, e o preço projetado com 30% de aumento.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

SELECT produto\_codigo AS codigo\_do\_produto, produto\_nome AS nome\_do\_produto, produto\_preco AS preco\_atual\_do\_produto, produto\_preco \* 1.3 AS produto\_com\_aumento

FROM produto;

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

1. Selecionar todos os produtos por ordem de valor descendente, acrescentando um novo rótulo para as colunas exibidas.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

SELECT produto\_codigo AS codigo\_do\_produto, produto\_nome AS nome\_do\_produto, produto\_preco AS preco\_atual\_do\_produto, ue\_produto\_cod AS unidade\_medida\_produto

FROM produto

ORDER BY produto\_preco DESC;

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

1. Selecionar todos os Produtos por ordem de nome ascendente e valor descendente.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

SELECT \*

FROM produto

ORDER BY produto\_nome ASC, produto\_preco DESC;

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

1. Selecionar código do cliente, nome do cliente e data de entrada dos pedidos por ordem descendente de data de entrada, usando aliases (renomeação de tabela).

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

SELECT t1.cliente\_id, t1.cliente\_nome, t2.pedido\_data\_entrada

FROM cliente AS t1, pedido AS t2

WHERE t1.cliente\_id = t2.cliente\_id

ORDER BY t2.pedido\_data\_entrada DESC;

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

1. Contar a quantidade de pedidos feitos para o cliente 05.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

SELECT COUNT(\*) AS Total\_de\_pedidos

FROM pedido

WHERE cliente\_id = 5;

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

1. Obter o MAIOR e o MENOR código de cliente da tabela cliente.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

SELECT MAX(cliente\_id) AS maior\_id\_cliente, MIN(cliente\_id) AS menor\_id\_cliente FROM cliente;

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

1. Obter o MAIOR e o MENOR valor de pedido.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

SELECT MAX(pedido\_valor\_total) AS maior\_valor\_pedido,

MIN(pedido\_valor\_total) AS menor\_valor\_pedido

FROM pedido;

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

1. Obter o somatório do valor total geral de todos os pedidos.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

SELECT SUM(pedido\_valor\_total) AS valor\_total\_pedidos

FROM pedido;

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

1. Obter o somatório do valor total das vendas da tabela pedido no período de 02/12/1999 até 04/12/1999.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

SELECT SUM(pedido\_valor\_total) AS valor\_total\_pedidos

FROM pedido

WHERE pedido\_data\_entrada >= '99/12/02' AND pedido\_data\_entrada <= 9/12/04';

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Ou

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

SELECT SUM(pedido\_valor\_total) AS valor\_total\_pedidos

FROM pedido

WHERE pedido\_data\_entrada BETWEEN '99/12/02' AND ‘9/12/04';

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

30. Obter a média do valor total das vendas da tabela pedidos do ano de 1999.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

SELECT AVG(pedido\_valor\_total) AS valor\_medio\_pedidos

FROM pedido

WHERE pedido\_data\_entrada >= '99/01/01' AND pedido\_data\_entrada <= '99/12/31';

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Ou

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

SELECT AVG (pedido\_valor\_total) Media

FROM pedido

WHERE EXTRACT(YEAR FROM pedido\_data\_entrada) = 1999

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Ou

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

SELECT AVG(pedido\_valor\_total) AS valor\_medio\_pedidos

FROM pedido

WHERE pedido\_data\_entrada >= '99/01/01' AND pedido\_data\_entrada LIKE ‘99%’;

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Ou

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

SELECT AVG (pedido\_valor\_total) Media

FROM pedido

WHERE YEAR(pedido\_data\_entrada) = 1999

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

1. Mostrar o código do produto e a média de quantidade por produto vendido.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

SELECT produto\_codigo, AVG(item\_quantidade)

FROM item

GROUP BY produto\_codigo;

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

1. Mostrar a quantidade de clientes por Unidade Federativa.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

SELECT estado\_codigo, COUNT(\*)

AS Qtde\_clientes

FROM cliente

GROUP BY estado\_codigo;

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

1. Mostrar a quantidade de clientes por Unidade Federativa para clientes com desconto maior que 10%.

Ordem de avaliação:

* 1. Clausula WHERE
  2. Clausula GROUP BY
  3. Função de Agregação (SUM, MAX, MIN, COUNT)

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

SELECT estado\_codigo, COUNT(\*) AS Qtde\_clientes

FROM cliente

WHERE cliente\_perc\_desconto > 10 ;

GROUP BY estado\_codigo;

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

1. Mostrar o código do pedido, e a média de valor dos mesmos, somente para os pedidos cujo valor está acima da média total.

Ordem de avaliação:

* 1. Clausula WHERE
  2. Clausula GROUP BU
  3. Função de Agregação (SUM, MAX, MIN, COUNT)
  4. clausula HAVING

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

SELECT pedido\_identificacao, AVG(pedido\_valor\_total) AS valor\_medio

FROM pedido

GROUP BY pedido\_identificacao

HAVING pedido\_valor\_total > valor\_medio;

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

1. Mostrar código e o nome dos clientes que já fizeram pedidos, eliminando as repetições, classificando por nome do cliente descendente.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

SELECT cliente\_id, cliente\_nome

FROM cliente

WHERE cliente\_id IN (SELECT DISTINCT cliente\_id FROM pedido)

ORDER BY cliente\_nome DESC;

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

1. Mostrar código, nome dos produtos e valor total pedido por produto, dos produtos que já foram pedidos, classificando por nome do produto descendente selecionando somente os itens cuja soma total pedido por produto seja > 80, usando aliases.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

SELECT SUM(t1.item\_valor\_total) AS ValorTotalPorProduto, t2.produto\_nome AS nomeProduto, t2.produto\_codigo AS produtoCodigo

FROM item AS t1, produto AS t2

WHERE t1.produto\_codigo= t2.produto\_codigo

GROUP BY produtoCodigo

HAVING ValorTotalPorProduto > 80

ORDER BY nomeProduto DESC;

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

1. Mostrar código do pedido, código e nome do cliente, código e nome do produto e descrição da unidade de estoque do produto.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

SELECT t1.pedido\_identificacao, t2.cliente\_id, t3.cliente\_nome, t4.produto\_nome, t4.ue\_produto\_cod

FROM item AS t1, pedido AS t2, cliente AS t3, produto AS t4

WHERE t1.pedido\_identificacao = t2. pedido\_identificacao

AND t2.cliente\_id = t3.cliente\_id

AND t1.produto\_codigo = t4.produto\_codigo

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

1. Mostrar os dados dos produtos para os quais ainda não foi feito pedido.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

SELECT produto\_codigo, produto\_nome

FROM produto

WHERE produto\_codigo NOT IN (SELECT DISTINCT produto\_codigo FROM item)

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

1. Mostrar código, nome e preço dos produtos cujo preço do produto seja maior que a média de preço de todos os produtos.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

SELECT produto\_codigo, produto\_nome, produto\_preco

FROM produto

WHERE produto\_preco > (SELECT AVG(produto\_preco) FROM produto);

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

1. Mostrar código, nome e preço dos produtos cujo preço seja maior que a média de preço de todos os produtos e a unidade de estoque do produto seja KG ou UM ou L e tenha a letra ‘h’ em qualquer parte do nome do produto.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

SELECT produto\_codigo, produto\_nome, produto\_preco, ue\_produto\_cod FROM produto

WHERE (produto\_preco > (SELECT AVG(produto\_preco) FROM produto))

AND (ue\_produto\_cod = 'KG' OR ue\_produto\_cod = 'UM' OR ue\_produto\_cod = 'L')

AND (produto\_nome LIKE '%h%');

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Ou

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

SELECT produto\_codigo, produto\_nome, produto\_preco, ue\_produto\_cod FROM produto

WHERE (produto\_preco > (SELECT AVG(produto\_preco) FROM produto))

AND ue\_produto\_cod IN (‘KG','UM','L')

AND produto\_nome LIKE '%h%';

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

1. Mostrar os dados dos clientes que compraram no dia 02/12/1999 e que não compraram no dia 04/12/1999.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

SELECT cliente\_nome, cliente\_id

FROM cliente

WHERE cliente\_id IN (SELECT DISTINCT cliente\_id FROM pedido

WHERE pedido\_data\_entrada = '99/12/02')

AND cliente\_id NOT IN (SELECT DISTINCT cliente\_id FROM pedido WHERE pedido\_data\_entrada = '99/12/04');

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

1. Mostrar os dados dos produtos que nunca foram comprados por clientes que moram no estado do “Rio de Janeiro”.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

SELECT \* from produto

WHERE produto\_codigo NOT IN

(SELECT item.produto\_codigo

FROM item, pedido, cliente

WHERE item.pedido\_identificacao=pedido.pedido\_identificacao

AND pedido.cliente\_id = cliente.cliente\_id AND cliente.estado\_codigo = 'RJ');

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*