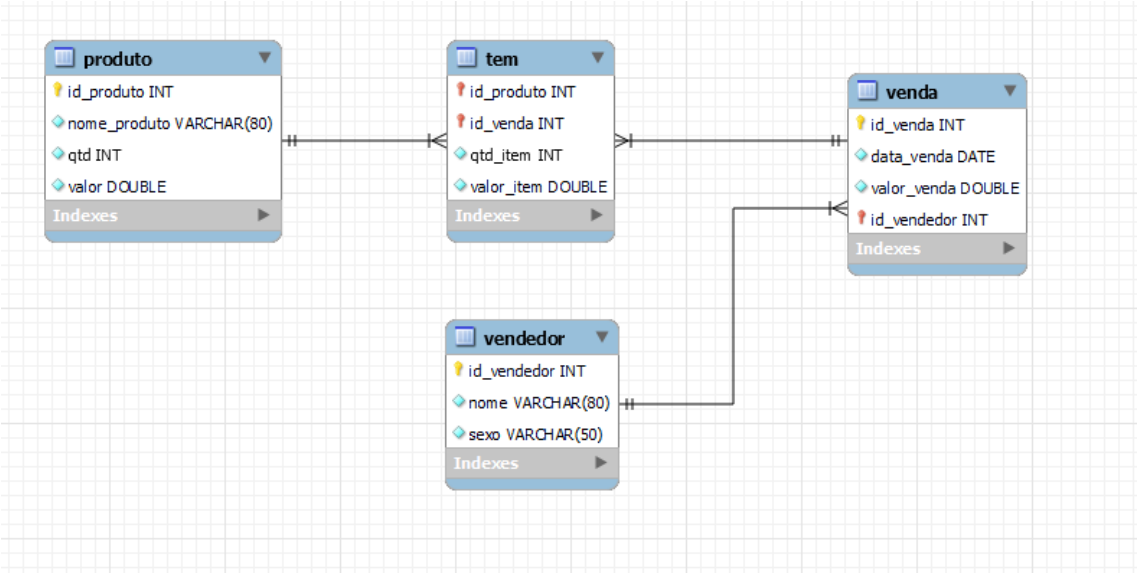
Modelo Lógico:



Código para o BD:

```
-- Schema atividade17

-- -----------------------------------------------------

CREATE SCHEMA IF NOT EXISTS `atividade17` DEFAULT CHARACTER SET utf8 ;

USE `atividade17` ;


-- -----------------------------------------------------

-- Table `atividade17`.`produto`

-- -----------------------------------------------------

CREATE TABLE IF NOT EXISTS `atividade17`.`produto` (

  `id_produto` INT NOT NULL AUTO_INCREMENT,

  `nome_produto` VARCHAR(80) NOT NULL,

  `qtd` INT NOT NULL,

  `valor` DOUBLE NOT NULL,

  PRIMARY KEY (`id_produto`),

  UNIQUE INDEX `id_produto_UNIQUE` (`id_produto` ASC) VISIBLE)

ENGINE = InnoDB;
```

```
-- -----------------------------------------------------
-- Table `atividade17`.`vendedor`
-- -----------------------------------------------------
CREATE TABLE IF NOT EXISTS `atividade17`.`vendedor` (
  `id_vendedor` INT NOT NULL AUTO_INCREMENT,
  `nome` VARCHAR(80) NOT NULL,
  `sexo` VARCHAR(50) NOT NULL,
  PRIMARY KEY (`id_vendedor`),
  UNIQUE INDEX `id_vendedor_UNIQUE` (`id_vendedor` ASC) VISIBLE)
ENGINE = InnoDB;


-- -----------------------------------------------------
-- Table `atividade17`.`venda`
-- -----------------------------------------------------
CREATE TABLE IF NOT EXISTS `atividade17`.`venda` (
  `id_venda` INT NOT NULL AUTO_INCREMENT,
  `data_venda` DATE NOT NULL,
  `valor_venda` DOUBLE NOT NULL,
  `id_vendedor` INT NOT NULL,
  PRIMARY KEY (`id_venda`, `id_vendedor`),
  UNIQUE INDEX `id_venda_UNIQUE` (`id_venda` ASC) VISIBLE,
  INDEX `fk_venda_vendedor1_idx` (`id_vendedor` ASC) VISIBLE,
  CONSTRAINT `fk_venda_vendedor1`
    FOREIGN KEY (`id_vendedor`)
    REFERENCES `atividade17`.`vendedor` (`id_vendedor`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION)
ENGINE = InnoDB;
```

```
-- -----------------------------------------------------
-- Table `atividade17`.`tem`
-- -----------------------------------------------------
CREATE TABLE IF NOT EXISTS `atividade17`.`tem` (
  `id_produto` INT NOT NULL,
  `id_venda` INT NOT NULL,
  `qtd_item` INT NOT NULL,
  `valor_item` DOUBLE NOT NULL,
  PRIMARY KEY (`id_produto`, `id_venda`),
  INDEX `fk_produto_has_venda_venda1_idx` (`id_venda` ASC) VISIBLE,
  INDEX `fk_produto_has_venda_produto_idx` (`id_produto` ASC) VISIBLE,
  CONSTRAINT `fk_produto_has_venda_produto`
    FOREIGN KEY (`id_produto`)
    REFERENCES `atividade17`.`produto` (`id_produto`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION,
  CONSTRAINT `fk_produto_has_venda_venda1`
    FOREIGN KEY (`id_venda`)
    REFERENCES `atividade17`.`venda` (`id_venda`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION)
ENGINE = InnoDB;
```

Codigos SQL:

```
use atividade17;

insert into produto (id_produto, nome_produto, qtd, valor) values
(10, "Mouse Gamer Corsair Harpoon PRO", 15, 210.00),
(11, "Microsoft Office Home 2021", 11, 990.50),
(12, "HD Externo Seagate Expansion 2TB", 18, 525.99),
```

```
(13, "Teclado Mecânico Gamer HyperX", 23, 735.90);

insert into vendedor (id_vendedor, nome, sexo) values
(101, "Aldebaran Silva", "M"),
(102, "Carina Dias", "F"),
(103, "Nicolle Cherry", "F");

insert into venda (id_venda, data_venda, valor_venda, id_vendedor) values
(1001, "2022-06-22", 2040.50, 101),
(1002, "2022-06-30", 735.99, 102),
(1003, "2022-07-02", 420.00, 103);

insert into tem (id_produto, id_venda, qtd_item, valor_item) values
(11, 1001, 1, 990.50),
(10, 1001, 5, 210.00),
(12, 1002, 1, 525.99),
(10, 1002, 1, 210.00),
(10, 1003, 2, 420.00);

/*1-Acrescentar na tabela vendedor uma coluna denominada loja varchar(50).*/
alter table vendedor add loja VARCHAR(50) not null;

/*2-Atribuir uma loja para cada funcionário cadastrado <= voltar dps
Funcionário 101 ---------- loja Centro
Funcionário 102 ---------- loja Santo Antônio
Funcionário 103 ---------- loja Floresta*/
update vendedor set loja = "loja Centro" where id_vendedor = 101;
update vendedor set loja = "loja Santo Antônio" where id_vendedor = 102;
update vendedor set loja = "loja Floresta" where id_vendedor = 103;
```

/*3-Incluir um novo produto com nome “Mouse sem fio Logitech MX Master 3” e valor do produto = 439.90, quantidade a sua escolha.*/

insert into produto (nome_produto, qtd, valor) values

("Mouse sem fio Logitech MX Master 3", 30,439.90);

/*4-Criar na tabela vendedor um atributo chamado email e faça o povoamento do mesmo seguindo a regra nome@praticabd.com.br; exemplo vendedor 102*/

select * from vendedor;

alter table vendedor add email VARCHAR(60);

update vendedor set email = "carina@praticabd.com.br" where id_vendedor = 102;

update vendedor set email = "aldebaran@praticabd.com.br" where id_vendedor = 101;

update vendedor set email = "nicolle@praticabd.com.br" where id_vendedor = 103;

/*5-Insira mais seis (6) vendedores na tabela vendedor de sua livre escolha.*/

/*6-Associe esses seis (6) novos vendedores para suas lojas, de sua livre escolha.*/

insert into vendedor (nome, sexo, loja, email) values

('João da Silva', 'M', 'loja Centro', 'joao.silva@hotmail.com'),

('Maria Oliveira', 'F', 'loja Santo Antônio', 'maria.oliveira@gmail.com'),

('Pedro Santos', 'M', 'loja Floresta', 'pedro.santos@hotmail.com'),

('Ana Costa', 'F', 'loja Centro', 'ana.costa@gmail.com'),

('Carlos Oliveira', 'M', 'loja Santo Antônio', 'carlos.oliveira@hotmail.com'),

('Juliana Santos', 'F', 'loja Floresta', 'juliana.santos@gmail.com');

/*7-Atualizar o nome do produto de código 11, para Microsoft Office Pro 2022 e seu valor para 1279.00.*/

update produto set valor = 1279.00 where id_produto = 11;

/*8-Crie Triggers na tabela itens de vendas para controlar o estoque de produtos conforme for a adição, atualização ou exclusão da quantidade de produtos vendidos na tabela itens de vendas.*/

/*9-Inserir uma nova venda na tabela venda e atribua 0 (zero) no campo valor_venda e associe ao vendedor 101 no dia 28/06/2022.*/

insert into venda (data_venda, valor_venda, id_vendedor) values

("2022-06-28", 0, 101);

/*10-Adicione mais um comando na trigger criada na tabela itens de vendas para atualizar o campo valor_venda (que atualmente está com zero) da tabela

venda, o valor total da venda deve aparecer logo depois de executar o comando SQL da questão seguinte.*/

/*11-Insira a venda feita pelo vendedor 101 de 1 Teclado Mecânico Gamer HyperX e 1 Mouse Gamer Corsair Harpoon PRO na tabela Itens de vendas.*/

insert into produto (nome_produto, qtd, valor) values

("Teclado Mecânico Gamer HyperX", 10, 350.00),

("Mouse Gamer Corsair Harpoon PRO", 15, 450.00);

insert into venda (data_venda, valor_venda, id_vendedor) values

("2022-07-02", 350.00, 101),

("2022-07-10", 450.00, 101);

insert into tem (id_produto, id_venda, qtd_item, valor_item) values

(15, 1005, 1, 350.00),

(16, 1006, 1, 450.00);

/*12-Faça um comando SQL para mostrar a última venda realizada junto com seu valor total (valor_venda) e o vendedor responsável.*/

select venda.valor_venda, vendedor.nome from venda inner join vendedor on venda.id_vendedor = vendedor.id_vendedor order by venda.data_venda desc limit 1;

/*13-Faça um script SQL para conferir se o cálculo do valor da venda da última venda realizada está correta.*/

/*14-Insira 4 (quatro) novas vendas da sua escolha. Invente todos os dados

(diversifique os produtos), logo em seguida insira as vendas na tabela itens das vendas. Realize os mesmos procedimentos das questões anteriores.*/

```
insert into produto (nome_produto, qtd, valor) values
("Fone de Ouvido Havit HV-H2002d", 50, 219.00),
("Mouse Gamer Revo", 5, 50.00),
("Mousepad Grande", 20, 50.00),
("Geforce GT740", 100, 500.00);
insert into venda (data_venda, valor_venda, id_vendedor) values
("2022-07-20", 219.00, 101),
("2022-07-22", 250.00, 109),
("2022-08-02", 50.00, 107),
("2022-08-11", 500.00, 106);
insert into tem (id_produto, id_venda, qtd_item, valor_item) values
(17, 1007, 1, 219.00),
(18, 1008, 1, 250.00),
(19, 1009, 1, 50.00),
(20, 1010, 1, 500.00);
select venda.valor_venda, vendedor.nome from venda inner join vendedor on venda.id_vendedor = vendedor.id_vendedor order by venda.data_venda desc limit 1;
```

/*15-Insira seus dados como sendo um vendedor da loja Centro.*/

```
insert into vendedor (nome, sexo, loja, email) values
("Arthur Correa", "M", "loja Centro", "arthurbarbosakk@gmail.com");
```

/*16-Insira os dados de um amigo como sendo da loja Floresta.*/

```
insert into vendedor (nome, sexo, loja, email) values
("Diogo Dias", "M", "loja Floresta", "dgdias@gmail.com");
```

/*17-Listar o nome e o id de todos os vendedores que trabalham na loja Centro.*/

```
select * from vendedor where loja = "loja Centro";
```

/*18-Crie uma View para questão anterior, com a diferença de mostrar os dados embaralhados.*/

select * from vend_centro;

/*19-Crie uma Stored Procedures para questão anterior utilizando parâmetros, além

de escolher qual loja exibir, escolher também a quantidade de registros a

serem exibidos na consulta.*/

call vend_centro_proc("loja Floresta", 4);

/*20-Listar o nome e o id de todos os vendedores ordenados alfabeticamente.*/

select vendedor.id_vendedor, vendedor.nome from vendedor order by nome asc;

/*21-Listar o código, nome e quantidade de todos os produtos ordenados alfabeticamente.*/

select id_produto, nome_produto, qtd from produto order by nome_produto asc;

/*22-Listar todas as vendas junto com o nome e email do seu respectivo vendedor ordenados pelo valor da venda de forma decrescente.*/

select venda.id_venda, vendedor.nome, vendedor.email from venda inner join vendedor order by vendedor.nome asc;

/*23-Altere o nome do seu último vendedor para Moe Szyslak, sexo masculino, email moebar_duff@outlook.com e loja Santo Antônio.*/

update vendedor set nome = "Moe Szyslak", sexo = "M", email = "email moebar_duff@outlook.com", loja = "loja Santo Antônio" where id_vendedor = 111;

/*24-Insira 10 novos produtos da sua escolha (dados e informações corretas).*/

INSERT INTO produto (nome_produto, qtd, valor) VALUES

("Fone de Ouvido Havit HV-H2002d", 50, 219.00),

("Teclado Mecânico Redragon K552", 30, 159.90),

("Mouse Gamer Logitech G502", 45, 249.00),

("Monitor 24\" LG UltraWide", 20, 699.00),

("HD Externo Seagate 1TB", 40, 349.90),

("Smartwatch Apple Watch Series 8", 25, 3499.00),

("Cadeira Gamer DXRacer Racing", 10, 1299.00),

("Microfone Condensador USB Blue Yeti", 12, 649.00),

("Webcam Logitech C920", 35, 389.00),

("Roteador TP-Link Archer AX50", 22, 499.00),

("Caixa de Som Bluetooth JBL Flip 6", 28, 549.00);

/*25-Insira 5 (cinco) novas vendas da sua escolha. Invente todos os dados (diversifique os produtos), logo em seguida insira as vendas na tabela itens de venda.

Realize os mesmos procedimentos para adição de novas vendas utilizando as triggers criadas anteriormente.*/

insert into venda (data_venda, valor_venda, id_vendedor) values

("2023-02-22", 159.90, 101),

("2023-03-25", 3499.00, 109),

("2023-08-12", 499.00, 107),

("2022-08-17", 549.00, 106),

("2022-09-27", 699.00, 102);

insert into tem (id_produto, id_venda, qtd_item, valor_item) values

(22, 1011, 1, 159.90),

(26, 1012, 1, 3499.00),

(30, 1013, 1, 499.00),

(31, 1014, 1, 549.00),

(24, 1015, 1, 699.00);

/*26-Confira como está seu estoque de produtos exibindo o nome e a quantidade na consulta, crie um Stored Procedures para sempre conferir essas informações.*/

```
call conferir_estoque;

/*27-Aumentar em 10% o valor de todos os produtos da sua empresa.*/
UPDATE produto SET valor = valor * 1.10;

/*28-Acrescente o campo nascimento na tabela vendedor para receber a data de nascimento dos vendedores.*/
alter table vendedor add column data_nascimento date;

/*29-Insira na tabela vendedor as datas de nascimento de todos os vendedores. Você pode escolher uma data qualquer.*/
UPDATE vendedor
SET data_nascimento = '1982-05-24'
WHERE id_vendedor = 102;

UPDATE vendedor
SET data_nascimento = '1990-11-15'
WHERE id_vendedor = 103;

UPDATE vendedor
SET data_nascimento = '1975-02-28'
WHERE id_vendedor = 104;

UPDATE vendedor
SET data_nascimento = '1988-09-04'
WHERE id_vendedor = 105;

UPDATE vendedor
SET data_nascimento = '1978-12-12'
WHERE id_vendedor = 106;
```

```
UPDATE vendedor
SET data_nascimento = '1985-07-21'
WHERE id_vendedor = 107;

UPDATE vendedor
SET data_nascimento = '1992-10-30'
WHERE id_vendedor = 108;

UPDATE vendedor
SET data_nascimento = '1980-06-17'
WHERE id_vendedor = 1009;

UPDATE vendedor
SET data_nascimento = '1973-08-05'
WHERE id_vendedor = 110;

UPDATE vendedor
SET data_nascimento = '1995-03-22'
WHERE id_vendedor = 111;
```

/*30-Exclua a última venda cadastrada, antes verifique e exclua os itens de vendas associados a ela, execute o Stored Procedures da questão 26, verifique se o estoque desses produtos foi restaurado.*/

```
DELETE FROM tem WHERE id_venda = 1014;
DELETE FROM venda WHERE id_venda = 1015;
call conferir_estoque;
```

/*31-Faça um script que exclua produto do código 14 cadastrado (se este produto estiver associado a uma venda, mantenha o script e produto cadastrado).*/

```
call excSeNassociado(14);
```

/*32-Qual a quantidade de produtos que temos cadastrados? – Total de registros na tabela de produtos, faça um Stored Procedures para armazenar esta informação e logo em seguida

faça o comando SQL para buscar a mesma lá dentro.*/

CREATE TABLE IF NOT EXISTS informacoes_produto (total_produtos INT);

call attProd;

SELECT total_produtos FROM informacoes_produto;

/*33-Exiba os produtos vendidos no mês de junho e qual vendedor realizou essas vendas.*/

select produto.nome_produto, vendedor.nome, venda.data_venda from venda inner join vendedor on vendedor.id_vendedor = venda.id_vendedor inner join produto where venda.data_venda between "2022-06-01" and "2022-06-31";

/*34-Listar o nome do produto mais caro.*/

select nome_produto, valor from produto where valor = (SELECT MAX(valor) FROM produto);

/*35-Listar o nome do produto mais barato.*/

select nome_produto, valor from produto where valor = (SELECT MIN(valor) FROM produto);

/*36-Crie Stored Procedures que mostre as duas informações anteriores em simultâneo.*/

call valorMinMax;

/*37-Listar o nome de todos os produtos que iniciam com a letra M.*/

select * from produto where nome_produto like "M%";

/*38-Exibir o total de venda da empresa.*/

select sum(valor_venda) from venda;

/*39-Crie um View para questão acima.*/

```
select * from total_vendas_view;
```

/*40-Exibir o total de vendas da empresa para o vendedor de id 102.*/

```
select sum(valor_venda) as total_venda, vendedor.nome from venda inner join vendedor on venda.id_vendedor = vendedor.id_vendedor where venda.id_vendedor = 102;
```

/*41-Busque o nome de todos os vendedores que comecem com a letra do seu nome.*/

```
select * from vendedor where nome like "A%";
```

/*42-Exiba na tela o nome de todos os vendedores que terminem com a primeira letra do seu nome.*/

```
select * from vendedor where nome like "%A";
```

/*43-Listar todos os produtos utilizando um Stored Procedures.*/

```
call listProd;
```

/*44-Conte quantos vendedores do sexo masculino existem na base de dados.*/

```
select count(sexo) from vendedor where sexo = "M";
```

/*45-Conte quantos vendedores do sexo feminino existem na base de dados.*/

```
select count(sexo) from vendedor where sexo = "F";
```

/*46-Crie Stored Procedures que mostre as duas informações anteriores em simultâneo.*/

```
call sexoFunc;
```

/*47-Busque a data de nascimento do vendedor Aldebaran.*/

```
select data_nascimento from vendedor where id_vendedor = 101;
```

/*48-Retorne a loja em que o vendedor Moe Szyslak trabalha.*/

```
select loja from vendedor where id_vendedor = 111;
```

/*49-Listar o nome de todos os vendedores do sexo masculino que tenha a palavra Silva em seu nome.*/

```
select nome, sexo from vendedor where sexo = "M" and nome like "%Silva%";
```

/*50-Listar o nome de todos os vendedores que nunca realizaram uma venda.*/

```
select nome, valor_venda from vendedor inner join venda on vendedor.id_vendedor = venda.id_vendedor where venda.valor_venda = 0;
```

/*51-Listar o nome de todos os produtos que já foram vendidos.*/

```
select nome_produto from produto inner join venda where valor_venda > 0;
```

/*52-Listar o nome de todos os vendedores que realizaram alguma venda.*/

```
select nome from vendedor inner join venda on vendedor.id_vendedor = venda.id_vendedor where valor_venda > 0;
```

/*53-Exiba a quantidade de vendedores que tem na loja Santo Antônio.*/

```
select count(id_vendedor) from vendedor where loja = "loja Santo Antônio";
```

/*54-Qual é o produto mais caro que a empresa comercializa? E quanto tem no estoque?*/

```
select nome_produto, valor, qtd from produto where valor = (select max(valor) from produto);
```

/*55-Qual é o produto mais barato que a empresa comercializa? E quanto tem no estoque?*/

```
select nome_produto, valor, qtd from produto where valor = (select min(valor) from produto);
```

/*56-Qual a média de preços dos produtos?*/

```
select AVG(valor) as media_valor from produto;
```

/*57-Qual a quantidade de vendedores que temos cadastrados? – Total de registros na tabela de vendedor, faça um Stored Procedures para armazenar esta

informação e logo em seguida faça o comando SQL para buscar a mesma lá dentro.*/

call contVendedor;

/*58-Quais os produtos possuem seu preço superior à média de preços de nossa base.*/

select nome_produto from produto where valor > (select avg(valor) from produto);

/*59-Exibir o total de vendas da empresa para o vendedor de id 105.*/

select count(valor_venda) from vendedor inner join venda on vendedor.id_vendedor = venda.id_vendedor where vendedor.id_vendedor = 105;

/*60-Listar o nome, id e sexo do vendedor responsável pela venda de número 1001 e qual foi o valor da venda.*/

select vendedor.id_vendedor, nome, sexo, valor_venda from vendedor inner join venda on vendedor.id_vendedor = venda.id_vendedor where venda.id_venda = 1001;

/*61-Aumente em 50% o valor do produto cadastrado onde o código for 19.*/

update produto set valor = valor * 1.50 where id_produto = 19;

/*62-Diminua em 15% o valor do produto cadastrado onde o código for 22.*/

update produto set valor = valor * 0.85 where id_produto = 22;

/*63-Crie uma Stored Procedures para as duas questões anteriores utilizando parâmetros e entrada e saída. VOLTAR*/

call aumentarValorProd(22);

/*64-Listar o nome de todos os vendedores que realizaram mais de uma venda.*/

select nome from vendedor inner join venda on vendedor.id_vendedor = venda.id_vendedor group by vendedor.id_vendedor, vendedor.nome having count(venda.id_venda) > 1;

/*65-Listar o nome de todos os vendedores e o código de sua venda, assim como o valor da venda.*/

select nome, id_venda, valor_venda from vendedor inner join venda on vendedor.id_vendedor = venda.id_vendedor;

/*66-Qual foi a maior venda feita pela empresa até agora? E quem foi o responsável por esta venda.*/

select nome, max(valor_venda) from vendedor as v inner join venda as va on v.id_vendedor = va.id_vendedor;

/*67-Altere a quantidade de itens da tabela itens de venda onde o código do produto for 16 e 18 ou outro de sua livre escolha.*/

update produto set qtd = 50 where id_produto = 16;

update produto set qtd = 60 where id_produto = 18;

update produto set qtd = 30 where id_produto = 22;

/*68-Faça um script SQL para conferir se os valores das vendas da tabela venda tiveram alguma alteração depois que foi realizado o script da questão anterior. VOLTAR*/

select v.id_venda, v.valor_venda as valor_atual, vo.valor_venda AS valor_original from venda v inner join venda vo on v.id_venda = vo.id_venda where v.valor_venda <> vo.valor_venda;

/*69-Crie uma função para buscar vendedor por código, caso a função encontre o vendedor, exiba o nome dele, senão exiba a mensagem que o vendedor não

existe ou não foi encontrado.*/

SELECT BuscarVendedorPorCodigo(101) AS resultado;

/*70-Atualize todos os e-mails dos vendedores para o endereço @gghardware.com.br.*/

update vendedor set email = CONCAT(email, '@gghardware.com.br');

/*71-Listar os nomes e os e-mails dos vendedores agrupados por sexo.*/

select nome, email, sexo from vendedor order by sexo asc;

/*72-Faça um script SQL que exiba as vendas entre 1000 a 2000 reais, junto com a data e o vendedor responsável.*/

select nome, data_venda, valor_venda from vendedor inner join venda on vendedor.id_vendedor = venda.id_vendedor where valor_venda between 1000 and 2000;

/*73-Faça um script SQL utilizando o SQL CASE para retornar sobre status do seu estoque, caso seu estoque esteja em um nível com até 10 itens, retorne que o

estoque deste produto está adequado, caso esteja abaixo de 5 itens, retorne que o estoque já está em alerta para nova reposição, caso esteja abaixo de 2,

retorne que o estoque precisa ser reabastecido com urgência, acima de 10

retorne estoque OK.*/

call verificaEstoque;

/*74-Crie uma função para buscar produto por código, caso a função encontre o produto, exiba o nome dele e o valor, senão exiba a mensagem que o produto não existe ou não foi encontrado.*/

call buscarProdutoCodigo(17);

/*75-Faça um script SQL utilizando o SQL CASE para retornar a comissão dos vendedores, caso eles tenham vendido acima de 3 mil reais, retorne o valor de

sua comissão de 15% sobre o valor total de suas vendas, caso seja abaixo de 3 mil até 2 mil, retorne sua comissão de 10%, caso seja abaixo de 2 mil retorne 5%,

se for abaixo de 1 mil retorne que não terá comissão.*/

select vendedor.id_vendedor, vendedor.nome, SUM(valor_venda) AS total_vendas, case when SUM(valor_venda) > 3000 THEN SUM(valor_venda) * 0.15 when SUM(valor_venda) >= 2000 THEN SUM(valor_venda) * 0.10

when SUM(valor_venda) >= 1000 THEN SUM(valor_venda) * 0.05 else 0 end as valor_comissao, case when SUM(valor_venda) > 3000 THEN '15% de comissão' when SUM(valor_venda) >= 2000 THEN '10% de comissão'

when SUM(valor_venda) >= 1000 THEN '5% de comissão' else 'Nenhuma comissão' end as status_comissao from vendedor inner join venda ON vendedor.id_vendedor = venda.id_vendedor

group by vendedor.id_vendedor, vendedor.nome;

/*76-Mostre o nome de todos os produtos, e a seus valores convertidos em dólar e euro.*/

call valorConvertidos;

/*77-Mostre o Top 3 produtos mais vendidos da sua loja.*/

select nome_produto from produto inner join tem on produto.id_produto = tem.id_produto group by nome_produto order by qtd_item desc limit 3;

/*78-Liste os vendedores que nasceram nos anos 90 e que trabalham na loja Centro. (Nenhum trabalha na loja Centro)*/

select nome, loja, data_nascimento from vendedor where loja = "loja Centro" and data_nascimento between "1990-01-01" and "1999-12-31";

/*79-Liste os vendedores que nasceram nos anos 2000 e que trabalham na loja Floresta e do Sexo Feminino.*/

select nome, loja, data_nascimento from vendedor where loja = "loja Floresta" and data_nascimento between "2000-01-01" and "2009-12-31";

/*80-Delete o vendedor que não conseguiu vender produtos, delete o produto que foi menos vendido ou que nunca foi vendido.*/

delete from vendedor where id_vendedor = 111;

delete from produto where id_produto = 31;