

Lista de Exercícios 10

Observações:

- As instruções SQL para criação dos bancos de dados estão disponíveis no *Arquivos Compartilhados* da página acadêmica;

Exercício 1:

Selecione 15 consultas das listas de exercícios anteriores e crie uma visão para cada uma delas.

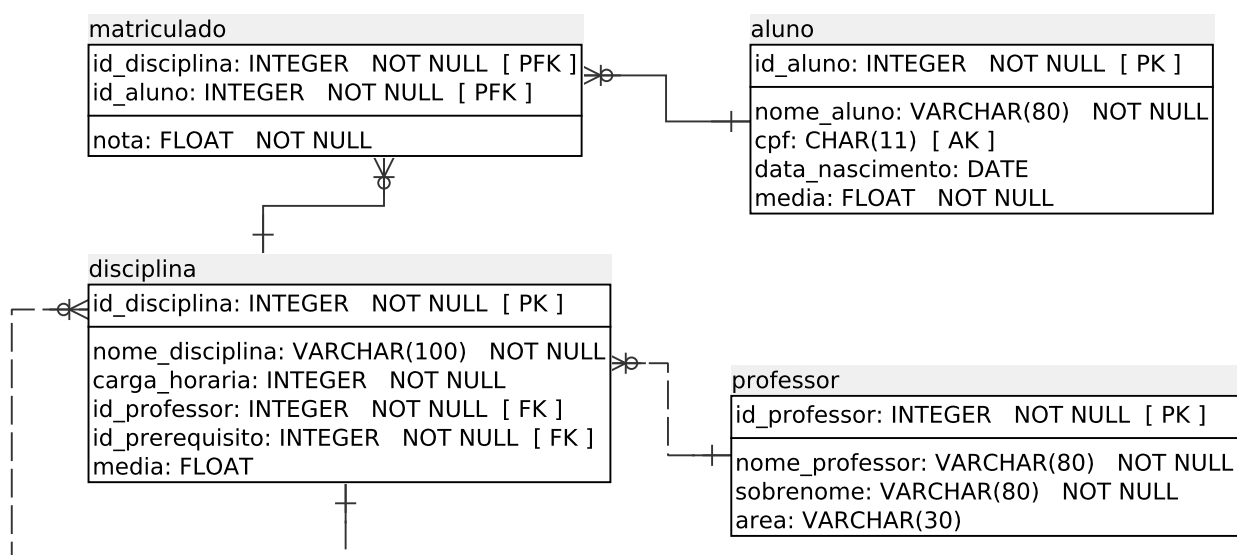


Figura 1: Banco de dados acadêmico II

Exercício 2:

Considere o banco de dados da Figura 1. Considere também que as notas iguais a **-1** (menos um) na tabela matriculado indicam que o aluno está apenas matriculado, mas não possui nenhuma nota. Escreva as instruções SQL para executar as seguintes ações:

- (a) Crie as regras necessárias para que a média das disciplinas seja atualizada automaticamente (lembre-se de desconsiderar as matrículas sem notas);

Solução:

```
CREATE OR REPLACE RULE mat_insert_dis_med AS
ON INSERT TO matriculado DO ALSO
  UPDATE disciplina AS d
  SET media = dm.media
  FROM (
    SELECT id_disciplina, AVG(nota) AS media
    FROM matriculado
    WHERE nota <> -1
    GROUP BY id_disciplina) AS dm
  WHERE d.id_disciplina = dm.id_disciplina
  AND d.id_disciplina = NEW.id_disciplina;

CREATE OR REPLACE RULE mat_delete_dis_med AS
ON DELETE TO matriculado DO ALSO
  UPDATE disciplina AS d
  SET media = dm.media
  FROM (
    SELECT id_disciplina, AVG(nota) AS media
    FROM matriculado
    WHERE nota <> -1
    GROUP BY id_disciplina) AS dm
  WHERE d.id_disciplina = dm.id_disciplina
  AND d.id_disciplina = OLD.id_disciplina;

CREATE OR REPLACE RULE mat_update_dis_med AS
ON UPDATE TO matriculado DO ALSO
  UPDATE disciplina AS d
  SET media = dm.media
  FROM (
    SELECT id_disciplina, AVG(nota) AS media
    FROM matriculado
    WHERE nota <> -1
    GROUP BY id_disciplina) AS dm
  WHERE d.id_disciplina = dm.id_disciplina
  AND d.id_disciplina IN (OLD.id_disciplina, NEW.id_disciplina);
```

- (b) Matricular os novos alunos automaticamente nas disciplinas sem pré-requisitos;

Solução:

```
CREATE OR REPLACE RULE aluno_insert_matricula AS
ON INSERT TO aluno DO ALSO (
  INSERT INTO matriculado(id_disciplina, id_aluno, nota)
  SELECT d.id_disciplina, currval('aluno_id_aluno_seq'), -1
  FROM disciplina AS d
  WHERE id_prerequisito IS NULL;
);
```

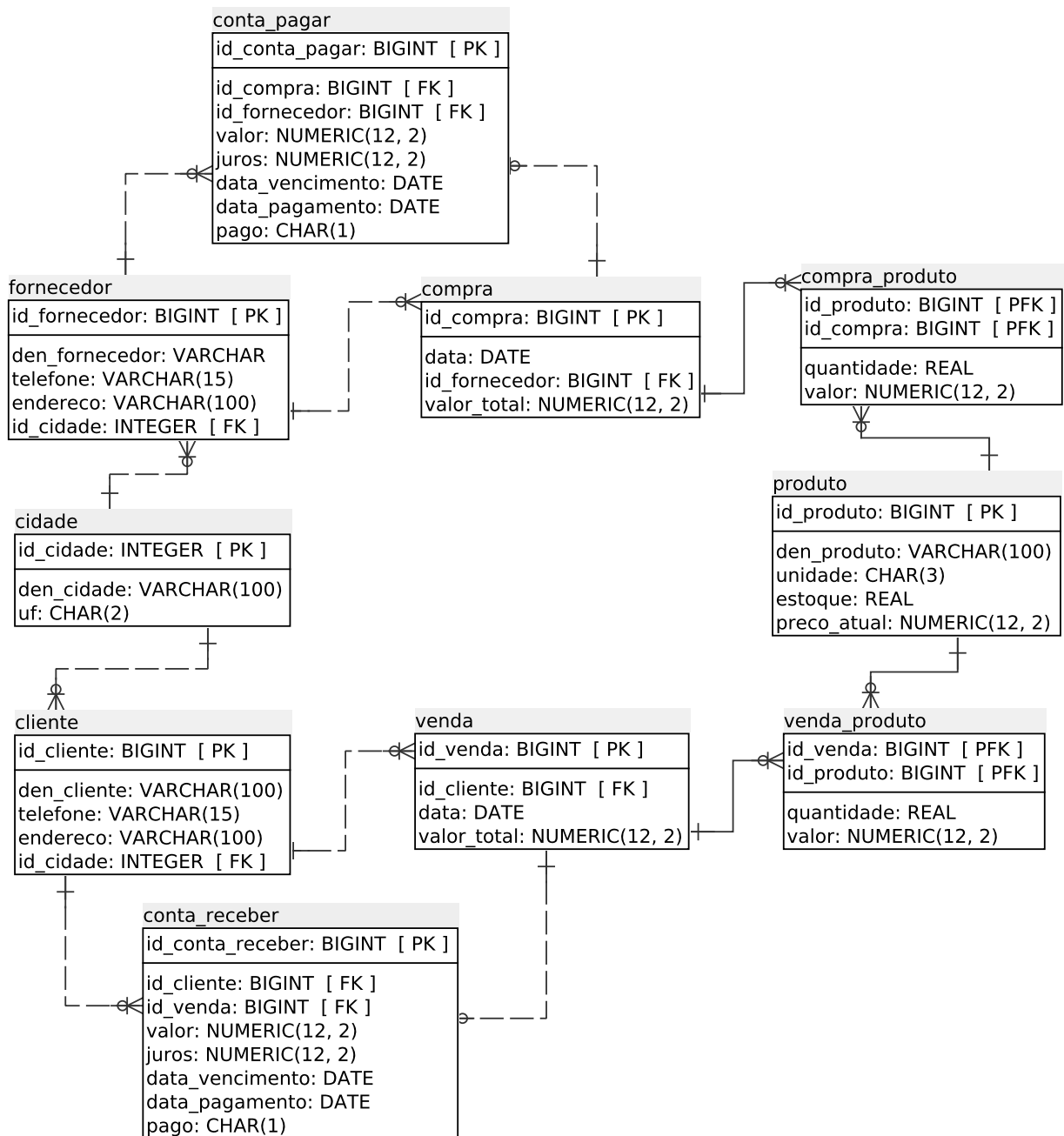


Figura 2: Banco de dados de uma empresa de varejo

Exercício 3:

Considere o banco de dados de uma empresa de varejo cujo esquema lógico é apresentado na Figura 2. Escreva as instruções SQL para executar as seguintes ações:

- (a) Atualizar automaticamente os estoques dos produtos de acordo com as compras e vendas;

Solução:

```
CREATE OR REPLACE RULE venda_produto_insert_estoque AS
ON INSERT TO venda_produto DO ALSO (
    UPDATE produto AS p
    SET estoque = estoque - NEW.quantidade
    WHERE p.id_produto = NEW.id_produto;
);

CREATE OR REPLACE RULE venda_produto_delete_estoque AS
ON DELETE TO venda_produto DO ALSO (
    UPDATE produto AS p
    SET estoque = estoque + OLD.quantidade
    WHERE p.id_produto = OLD.id_produto;
);

CREATE OR REPLACE RULE venda_produto_update_estoque AS
ON UPDATE TO venda_produto DO ALSO (
    UPDATE produto AS p
    SET estoque = estoque - NEW.quantidade
    WHERE p.id_produto = NEW.id_produto;

    UPDATE produto AS p
    SET estoque = estoque + OLD.quantidade
    WHERE p.id_produto = OLD.id_produto;
);

CREATE OR REPLACE RULE compra_produto_insert_estoque AS
ON INSERT TO compra_produto DO ALSO (
    UPDATE produto AS p
    SET estoque = estoque + NEW.quantidade
    WHERE p.id_produto = NEW.id_produto;
);

CREATE OR REPLACE RULE compra_produto_delete_estoque AS
ON DELETE TO compra_produto DO ALSO (
    UPDATE produto AS p
    SET estoque = estoque - OLD.quantidade
    WHERE p.id_produto = OLD.id_produto;
);

CREATE OR REPLACE RULE compra_produto_update_estoque AS
ON UPDATE TO compra_produto DO ALSO (
    UPDATE produto AS p
    SET estoque = estoque + NEW.quantidade
    WHERE p.id_produto = NEW.id_produto;

    UPDATE produto AS p
    SET estoque = estoque - OLD.quantidade
    WHERE p.id_produto = OLD.id_produto;
);
```

- (b) Atualizar automaticamente o total das vendas de acordo com os itens vendidos;

Solução:

```
CREATE OR REPLACE RULE venda_produto_insert_estoque AS
ON INSERT TO venda_produto DO ALSO (
    UPDATE venda AS v
    SET valor_total = total
    FROM (
        SELECT id_venda,
               SUM(quantidade * valor) AS total
        FROM venda_produto
        WHERE id_venda = NEW.id_venda
        GROUP BY id_venda
    ) AS vpt
    WHERE v.id_venda = vpt.id_venda
    AND v.id_venda = NEW.id_venda;
);

CREATE OR REPLACE RULE venda_produto_delete_estoque AS
ON DELETE TO venda_produto DO ALSO (
    UPDATE venda AS v
    SET valor_total = total
    FROM (
        SELECT id_venda,
               SUM(quantidade * valor) AS total
        FROM venda_produto
        WHERE id_venda = OLD.id_venda
        GROUP BY id_venda
    ) AS vpt
    WHERE v.id_venda = vpt.id_venda
    AND v.id_venda = OLD.id_venda;
);

CREATE OR REPLACE RULE venda_produto_update_estoque AS
ON UPDATE TO venda_produto DO ALSO (
    UPDATE venda AS v
    SET valor_total = total
    FROM (
        SELECT id_venda,
               SUM(quantidade * valor) AS total
        FROM venda_produto
        WHERE id_venda IN (NEW.id_venda, OLD.id_venda);
        GROUP BY id_venda
    ) AS vpt
    WHERE v.id_venda = vpt.id_venda
    AND v.id_venda IN (NEW.id_venda, OLD.id_venda);
);
```

- (c) Atualizar automaticamente o total das compras de acordo com os itens comprados;

Solução:

```
CREATE OR REPLACE RULE compra_produto_insert_estoque AS
ON INSERT TO compra_produto DO ALSO (
    UPDATE compra AS v
    SET valor_total = total
    FROM (
        SELECT id_compra,
               SUM(quantidade * valor) AS total
        FROM compra_produto
        WHERE id_compra = NEW.id_compra
        GROUP BY id_compra
    ) AS vpt
    WHERE v.id_compra = vpt.id_compra
    AND v.id_compra = NEW.id_compra;
);

CREATE OR REPLACE RULE compra_produto_delete_estoque AS
ON DELETE TO compra_produto DO ALSO (
    UPDATE compra AS v
    SET valor_total = total
    FROM (
        SELECT id_compra,
               SUM(quantidade * valor) AS total
        FROM compra_produto
        WHERE id_compra = OLD.id_compra
        GROUP BY id_compra
    ) AS vpt
    WHERE v.id_compra = vpt.id_compra
    AND v.id_compra = OLD.id_compra;
);

CREATE OR REPLACE RULE compra_produto_update_estoque AS
ON UPDATE TO compra_produto DO ALSO (
    UPDATE compra AS v
    SET valor_total = total
    FROM (
        SELECT id_compra,
               SUM(quantidade * valor) AS total
        FROM compra_produto
        WHERE id_compra IN (NEW.id_compra, OLD.id_compra)
        GROUP BY id_compra
    ) AS vpt
    WHERE v.id_compra = vpt.id_compra
    AND v.id_compra IN (NEW.id_compra, OLD.id_compra);
);
```

- (d) Crie os atributos de limite de crédito e saldo de crédito para os clientes. Atribua o valor de 30% do total de vendas de cada cliente para seu limite de crédito e para seu saldo de crédito. Faça com que o saldo de crédito seja atualizado automaticamente de acordo com as vendas.

Solução:

```
ALTER TABLE cliente
ADD limite_credito REAL;

ALTER TABLE cliente
ADD saldo_credito REAL;

UPDATE cliente AS c
SET limite_credito = limite,
    saldo_credito = limite
FROM (
    SELECT id_cliente,
           SUM(valor_total) AS limite
    FROM venda
    GROUP BY id_cliente
) AS cl
WHERE c.id_cliente = cl.id_cliente;

CREATE OR REPLACE RULE venda_insert_cliente_saldo AS
ON INSERT TO venda DO ALSO (
    UPDATE cliente AS c
    SET saldo_credito = saldo_credito - v.valor_total
    FROM venda AS v
    WHERE v.id_cliente = NEW.id_cliente;
);

CREATE OR REPLACE RULE venda_produto_delete_estoque AS
ON DELETE TO venda DO ALSO (
    UPDATE cliente AS c
    SET saldo_credito = saldo_credito + v.valor_total
    FROM venda AS v
    WHERE v.id_cliente = OLD.id_cliente;
);

CREATE OR REPLACE RULE venda_produto_update_estoque AS
ON UPDATE TO venda DO ALSO (
    UPDATE cliente AS c
    SET saldo_credito = saldo_credito - v.valor_total
    FROM venda AS v
    WHERE v.id_cliente = NEW.id_cliente;

    UPDATE cliente AS c
    SET saldo_credito = saldo_credito + v.valor_total
    FROM venda AS v
    WHERE v.id_cliente = OLD.id_cliente;
);
```