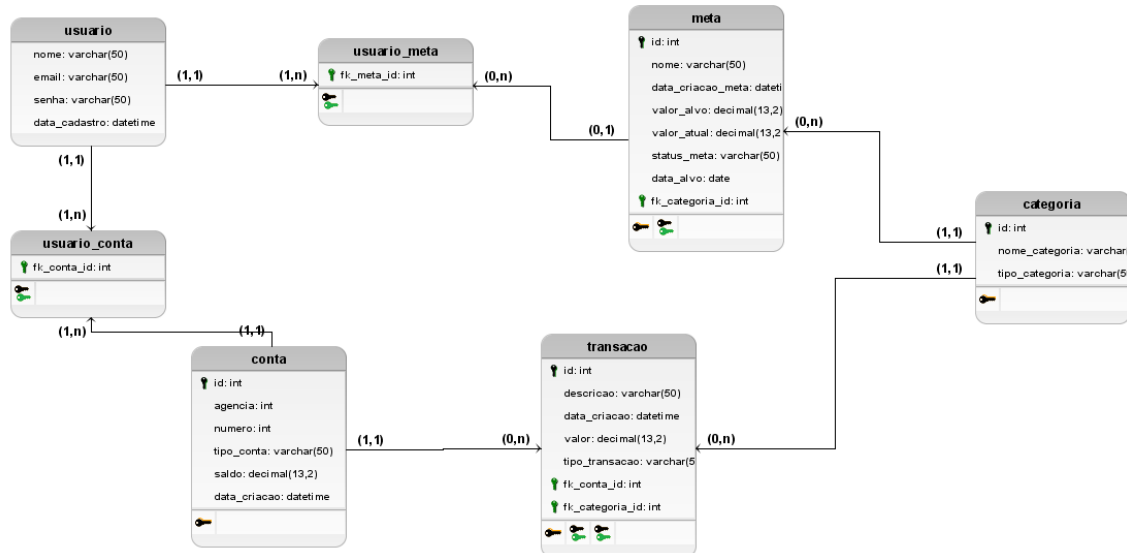


Nome: Nicole dos Santos

RM: 565891

## Modelo Relacional



## Script de Atualização SQL

-- Alterar os dados de um usuário (como nome, e-mail e etc.). -- Ou seja, atualizar os dados de um usuário, a partir do seu código de referência.

UPDATE USUARIO

SET NOME='Gabriel', EMAIL='gabriel@teste.com', SENHA='9876'

WHERE ID=1;

-- Alterar os dados de uma receita já cadastrada, -- a partir do código da tabela, ou seja, utilize o código do usuário e o código da receita.

UPDATE TRANSACAO

SET DESCRICAO='13 salario', VALOR=3000, TIPO\_TRANSACAO='ENTRADA',

FK\_CONTA\_ID=1, FK\_CATEGORIA\_ID=1

WHERE ID=1 AND FK\_CONTA\_ID=1;

-- Alterar os dados de uma despesa já cadastrada, -- a partir do código da tabela, ou seja, utilize o código do usuário e o código da despesa.

```
UPDATE TRANSACAO
```

```
SET DESCRICAO='Mercado', VALOR=600, TIPO_TRANSACAO='SAIDA',  
FK_CONTA_ID=1,
```

```
FK_CATEGORIA_ID=2
```

```
WHERE ID=2 AND FK_CONTA_ID=1;
```

-- Alterar os dados de um investimento já cadastrado, -- a partir do código da tabela, ou seja, utilize o código do usuário e o código do investimento.

```
UPDATE META SET NOME='Viagem', VALOR_ALVO=10000, VALOR_ATUAL=30,  
STATUS_META='ATIVO',
```

```
DATA_ALVO= DATE '2025-12-31', FK_CATEGORIA_ID=3
```

```
WHERE ID=1;
```

### Scripts Populando Tabelas

-- Cadastrar os dados de um novo usuário.

```
INSERT INTO USUARIO (ID, NOME, EMAIL, SENHA, DATA_CADASTRO)
```

```
VALUES(1, 'Lucas', 'lucas@teste.com', '1234', SYSDATE);
```

-- Cadastra uma nova conta

```
INSERT INTO CONTA (ID, AGENCIA, NUMERO, TIPO_CONTA, SALDO, DATA_CRIACAO)  
VALUES(1, 3456, 987653, 'CORRENTE', 0, SYSDATE);
```

-- Associando a conta recém criada ao usuário

```
INSERT INTO USUARIO_CONTA (FK_CONTA_ID, FK_USUARIO_ID)
```

```
VALUES(1, 1);
```

-- Cadastrando categoria para a transacao

```
INSERT INTO CATEGORIA (ID, NOME_CATEGORIA, TIPO_CATEGORIA)
```

```
VALUES(1, 'Receitas', 'Entrada');
```

-- Cadastrar os dados de uma nova receita (entrada de dinheiro) para um usuário.

```
INSERT INTO TRANSACAO (ID, DESCRICAO, DATA_CRIACAO, VALOR,  
TIPO_TRANSACAO, FK_CONTA_ID, FK_CATEGORIA_ID)
```

```
VALUES(1, 'Salario', sysdate, 2000, 'ENTRADA', 1, 1);
```

-- Cadastrando categoria para a transacao

```
INSERT INTO CATEGORIA (ID, NOME_CATEGORIA, TIPO_CATEGORIA)
```

```
VALUES(2, 'Despesas', 'Saida');
```

-- Cadastrar os dados de uma nova despesa (gasto) de um usuário.

```
INSERT INTO TRANSACAO (ID, DESCRICAO, DATA_CRIACAO, VALOR,  
TIPO_TRANSACAO, FK_CONTA_ID, FK_CATEGORIA_ID)
```

```
VALUES(2, 'Compra Mercado', sysdate, 500, 'SAIDA', 1, 2);
```

-- Cadastrando categoria para a meta - investimento

```
INSERT INTO CATEGORIA (ID, NOME_CATEGORIA, TIPO_CATEGORIA)
```

```
VALUES(3, 'Investimento', 'Entrada');
```

-- Cadastrar os dados de um novo investimento feito por um usuário.

```
INSERT INTO META (ID, NOME, DATA_CRIACAO_META, VALOR_ALVO, VALOR_ATUAL,  
STATUS_META, DATA_ALVO, FK_CATEGORIA_ID)
```

```
VALUES(1, 'Viagem Internacional', sysdate, 30000, 0, 'ATIVA', DATE '2025-12-25', 3);
```

```
INSERT INTO USUARIO_META (FK_META_ID, ID, FK_USUARIO_ID)
```

```
VALUES(1, 1, 1);
```

Scripts Consultas

### --Consultas Simples

-- Consultar os dados de um usuário específico (usando o código dele).

```
SELECT * FROM USUARIO WHERE ID = 1;
```

--Consultar os dados de uma despesa específica (usando o código do usuário e o da despesa).

```
SELECT * FROM TRANSACAO WHERE ID = 2 AND FK_CONTA_ID = 1;
```

-- Consultar os dados de um investimento específico (com código do usuário e do investimento).

```
SELECT * FROM META WHERE ID = 1;
```

### --Consultas Ordenadas

--Consultar todas as despesas de um usuário, --ordenando da mais nova para a mais antiga. --Ou seja, da mais recente para a mais antiga.

```
SELECT t.* FROM transacao t
```

```
JOIN usuario_conta uc ON t.fk_conta_id = uc.fk_conta_id
```

```
JOIN usuario u ON u.id = uc.fk_usuario_id
```

```
WHERE u.id = 1 AND t.fk_categoria_id = 2
```

```
ORDER BY DATA_CRIACAO DESC
```

-- Consultar todos os investimentos de um usuário, --do mais novo para o mais antigo.  
--Ou seja, do investimento mais recente para o mais antigo.

```
SELECT m.* FROM META m
```

```
JOIN USUARIO_META um ON m.id = um.FK_META_ID
```

```
JOIN USUARIO u ON u.id = um.FK_USUARIO_ID
```

```
WHERE u.id = 1
```

```
ORDER BY DATA_CRIACAO_META DESC
```

-- Consulta para o Dashboard

-- Consultar as informações principais de um usuário, --junto com a última despesa e o último investimento feitos deste usuário. --Ou seja, essa consulta deve retornar apenas uma linha, --com as principais informações de um usuário, assim como informações da última despesa --e do último investimento.

SELECT

u.nome AS nome\_usuario,

d.descricao AS descricao\_despesa,

d.VALOR AS valor\_despesa,

m.nome AS investimento,

m.VALOR\_ATUAL AS valor\_atual\_investido

FROM TRANSACAO d

JOIN USUARIO\_CONTA uc ON uc.FK\_CONTA\_ID = d.fk\_conta\_id

JOIN USUARIO u ON u.id = uc.FK\_USUARIO\_ID

JOIN USUARIO\_META um ON um.FK\_USUARIO\_ID = u.ID

JOIN META m ON m.id = um.FK\_META\_ID

WHERE u.id = 1

AND d.FK\_CATEGORIA\_ID = 2

AND d.id = (

-- Última transação para este usuário e categoria

SELECT MAX(t.id)

FROM TRANSACAO t

JOIN USUARIO\_CONTA uc2 ON uc2.FK\_CONTA\_ID = t.fk\_conta\_id

WHERE uc2.FK\_USUARIO\_ID = u.id

```
        AND t.FK_CATEGORIA_ID = 2

    ) AND m.id = (

        -- Última meta para este usuário

        SELECT MAX(m2.id)

        FROM META m2 JOIN USUARIO_META um2 ON um2.FK_META_ID = m2.id

        WHERE um2.FK_USUARIO_ID = u.id

    );
```