Script do Banco Mestre da Grelha

Criação do Banco:

create database MestreDaGrelha\_Testes

Tabela Usuários:

create table usuarios (

usuario\_id int primary key not null,

nome varchar(100),

email varchar(100) unique not null,

senha varchar(100) not null,

endereco varchar(255) not null,

telefone varchar(20) not null,

data\_criacao TIMESTAMP default CURRENT\_TIMESTAMP

)

Tabela Pedidos:

create table pedidos (

pedido\_id int primary key not null,

usuario\_id int not null,

data\_pedido TIMESTAMP default CURRENT\_TIMESTAMP,

status varchar(20) not null,

valor\_total decimal(10,2) not null,

foreign key (usuario\_id) references usuarios(usuario\_id)

)

Tabela Forma Pagamento:

create table forma\_pagamento (

forma\_pagamento\_id int primary key not null,

descricao varchar(50) not null /\*Débito, pix, crédito etc\*/

)

Tabela Pagamento CHAVE COMPOSTA:

create table pagamento (

pagamento\_id int not null,

pedido\_id int not null,

forma\_pagamento\_id int not null,

valor\_pago decimal(10, 2) not null,

data\_pagamento TIMESTAMP default CURRENT\_TIMESTAMP,

status varchar(20) not null, /\*Aprovado, rejeitado etc\*/

primary key (pagamento\_id, pedido\_id),

foreign key (pedido\_id) references pedidos(pedido\_id),

foreign key (forma\_pagamento\_id) references forma\_pagamento(forma\_pagamento\_id)

)

Tabela Categoria:

create table categoria (

categoria\_id int primary key not null,

nome varchar(50) not null,

descricao varchar(255) not null

)

Tabela Item/Produto CHAVE COMPOSTA:

create table item\_produto (

produto\_id int not null,

categoria\_id int not null,

nome varchar(100) not null,

descricao text not null,

preco decimal(10, 2) not null,

estoque int not null,

primary key (produto\_id, categoria\_id),

foreign key (categoria\_id) references categoria(categoria\_id)

)

Tabela Item/Pedido:

create table itens\_pedido (

item\_pedido int primary key not null,

pedido\_id int not null,

produto\_id int not null,

categoria\_id int not null,

quantidade int not null,

preco\_unitario decimal(10 ,2) not null,

total decimal(10, 2) generated always as (quantidade \* preco\_unitario) stored, /\*STORED já calcula e grava no banco, assim toda consulta, ele não precisa recalcular os valores\*/

foreign key (pedido\_id) references pedidos(pedido\_id),

foreign key (produto\_id, categoria\_id) references item\_produto(produto\_id, categoria\_id)

)

Tabela Auditoria Preço Item TRIGGER:

create table audit\_preco\_item (

audit\_id int primary key not null,

produto\_id int not null,

categoria\_id int not null,

data\_auditoria TIMESTAMP default CURRENT\_TIMESTAMP,

preco\_antigo decimal (10, 2) not null,

preco\_novo decimal (10, 2) not null,

motivo varchar(255) not null,

foreign key (produto\_id, categoria\_id) references item\_produto(produto\_id, categoria\_id)

)