NORMALIZAÇÃO

Normalização:

-Fizemos a normalização de uma tabela de cada vez, começando pela 1FN e avançando para a 2FN e 3FN.

-Eliminamos informações duplicadas para otimizar o armazenamento.

Resultados:

Tabelas Normalizadas:

Listamos as tabelas resultantes após a aplicação da normalização.

Explicitamos as chaves primárias e estrangeiras para facilitar a compreensão das relações entre as tabelas.

Conclusão

A normalização eficiente tornou o nosso projeto mais organizado, reduzindo redundâncias e melhorando a integridade do trabalh. Este processo é crucial para a manutenção a longo prazo do BD.

FORNECEDOR (id_fornecedor, nome, contato)

PRODUTOS (id_produto, estoque, nome, valor, id_fornecedor, id_categoria)

CATEGORIA (id_categoria, nome)

SUPERMERCADO (id_supermercado, nome, contato, endereco)

PEDIDOS (id_pedido, quantidade, id_supermercado)

PEDIDO (id_pedido, quantidade)

PEDIDO_PRODUTO (id_produto, id_pedido)

1a Forma Normal (1FN):

Para garantir a 1NF, certificamo-nos de que os atributos sejam atômicos:

FORNECEDOR: Nenhuma modificação necessária.

PRODUTOS: Nenhuma modificação necessária.

CATEGORIA: Nenhuma modificação necessária.

SUPERMERCADO: Nenhuma modificação necessária.

PEDIDOS: Nenhuma modificação necessária.

PEDIDO: Nenhuma modificação necessária.

PEDIDO_PRODUTO: Nenhuma modificação necessária.

NORMALIZAÇÃO

2a Forma Normal (2FN):

Nenhuma tabela possui uma chave primária composta, portanto, todas estão automaticamente na 2FN.

3a Forma Normal (3FN):

FORNECEDOR (id_fornecedor, nome, contato): Nenhuma dependência transitiva.

PRODUTOS (id_produto, estoque, nome, valor, id_fornecedor, id_categoria): Nenhuma dependência transitiva.

CATEGORIA (id_categoria, nome): Nenhuma dependência transitiva.

SUPERMERCADO (id_supermercado, nome, contato, endereco): Nenhuma dependência transitiva.

PEDIDOS (id_pedido, quantidade, id_supermercado): Nenhuma dependência transitiva.

PEDIDO (id_pedido, quantidade): Nenhuma dependência transitiva.

PEDIDO_PRODUTO (id_produto, id_pedido): Nenhuma dependência transitiva.

Tabelas Normalizadas:

FORNECEDOR (id_fornecedor, nome, contato)

PRODUTOS (id_produto, estoque, nome, valor, id_fornecedor, id_categoria)

CATEGORIA (id_categoria, nome)

SUPERMERCADO (id_supermercado, nome, contato, endereco)

PEDIDOS (id_pedido, quantidade, id_supermercado)

PEDIDO (id_pedido, quantidade)

PEDIDO_PRODUTO (id_produto, id_pedido)

NORMALIZAÇÃO

FORNECEDOR

PK: id_fornecedor

PRODUTOS

PK: id_produto

FK: id_fornecedor (Referenciando FORNECEDOR)

FK: id_categoria (Referenciando CATEGORIA)

CATEGORIA

PK: id_categoria

SUPERMERCADO

PK: id_supermercado

PEDIDOS

PK: id_pedido

FK: id_supermercado (Referenciando SUPERMERCADO)

PEDIDO

PK: id_pedido

PEDIDO_PRODUTO

PK: id_produto, id_pedido

FK: id_produto (Referenciando PRODUTOS)

FK: id_pedido (Referenciando PEDIDO)

NORMALIZAÇÃO

```sql

CREATE TABLE cps.tbFornecedor (
    id_fornecedor SERIAL PRIMARY KEY,
    nome VARCHAR(255) NOT NULL,
    contato VARCHAR(255) NOT NULL
);


CREATE TABLE cps.tbProdutos (
    id_produto SERIAL PRIMARY KEY,
    estoque INTEGER NOT NULL,
    nome VARCHAR(255) NOT NULL,
    valor DECIMAL(10, 2) NOT NULL,
    id_fornecedor INTEGER REFERENCES cps.tbFornecedor(id_fornecedor),
    id_categoria INTEGER REFERENCES cps.tbCategoria(id_categoria)
);


CREATE TABLE cps.tbCategoria (
    id_categoria SERIAL PRIMARY KEY,
    nome VARCHAR(255) NOT NULL
);
```

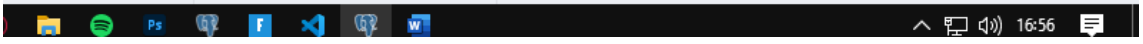
Query Query History

```sql
CREATE TABLE cps.tbSupermercado (
    id_supermercado SERIAL PRIMARY KEY,
    nome VARCHAR(255) NOT NULL,
    contato VARCHAR(255) NOT NULL,
    endereco VARCHAR(255) NOT NULL
);


CREATE TABLE cps.tbPedidos (
    id_pedido SERIAL PRIMARY KEY,
    quantidade INTEGER NOT NULL,
    id_supermercado INTEGER REFERENCES cps.tbSupermercado(id_supermercado)
);

CREATE TABLE cps.tbPedido (
    id_pedido SERIAL PRIMARY KEY,
    quantidade INTEGER NOT NULL
);

CREATE TABLE cps.tbProdutosPedido (
    id_produto INTEGER REFERENCES cps.tbProdutos(id_produto),
    id_pedido INTEGER REFERENCES cps.tbPedido(id_pedido)
```

Data Output Messages Notifications

```
CREATE TABLE

Query returned successfully in 162 msec.
```

Total rows: 0 of 0 Query complete 00:00:00.162 Ln 25, Col 17



NORMALIZAÇÃO