

Wellington Cesar Fonseca

DESENVOLVIMENTO FULL STACK

MISSÃO PRÁTICA | NÍVEL 2 | MUNDO 3

RPG0015 - VAMOS MANTER AS INFORMAÇÕES!

OBJETIVO

1. Identificar os requisitos de um sistema e transformá-los no modelo adequado.
2. Utilizar ferramentas de modelagem para bases de dados relacionais.
3. Explorar a sintaxe SQL na criação das estruturas do banco (DDL).
4. Explorar a sintaxe SQL na consulta e manipulação de dados (DML)
5. No final do exercício, o aluno terá vivenciado a experiência de modelar a base de dados para um sistema simples, além de implementá-la, através da sintaxe SQL, na plataforma do SQL Server.

2º PROCEDIMENTO | ALIMENTANDO A BASE

Quais as diferenças no uso de sequence e identity?

- **Sequence:** São valores que utilizam do auto incremento sequencial, mais simples e fácil de manipular ex: 1,2,3,4...
- **Identity:** São valores totalmente exclusivos e não sequenciais, menos flexíveis para se trabalhar, mas os tornam praticamente unique, ex: uuid

Qual a importância das chaves estrangeiras para a consistência do banco?

- **Integridade Referencial:** Mantém tudo sob controle e garante que não haja links quebrados. Digamos que você queira fazer um pedido em uma loja. Você precisa ter certeza de que a loja sabe quem você é e possui suas informações.
- **Coerência de dados:** Mantém os dados sob controle e evita que coisas estranhas aconteçam, como registros que não têm país.
- **Facilidade de Manutenção:** Essas ferramentas tornam muito fácil acompanhar e atualizar informações, para que você não precise se preocupar com dados confusos ou conexões interrompidas.

Quais operadores do SQL pertencem à álgebra relacional e quais são definidos no cálculo relacional?

- **Produto Cartesiano (\times):** `SELECT * FROM table1, table2;`
- **Projeção (π):** `SELECT column1, column2 FROM table;`
- **Diferença (-):** `SELECT * FROM table1 EXCEPT SELECT * FROM table2;`
- **Junção (\bowtie):** `SELECT * FROM table1 JOIN table2 ON table1.id = table2.id;`
- **Seleção (σ):** `SELECT * FROM table WHERE condition;`
- **União (\cup):** `SELECT * FROM table1 UNION SELECT * FROM table2;`
- **Cálculo Relacional:** Baseado na lógica de predicados. Embora não seja diretamente aplicável em SQL, é importante compreender os princípios subjacentes.

Como é feito o agrupamento em consultas, e qual requisito é obrigatório?

- todas as colunas do `SELECT` deverão estar no `GROUP BY`
- deverão conter campos em comuns e mais de uma linha da tabela
- ser adicionado `GROUP BY`



script.sql

```
-- Dados completos de pessoas físicas.
select *
from
  pessoa
where
  tipo = 'pf';

-- Dados completos de pessoas jurídicas.
select *
from
  pessoa
where
  tipo = 'pj';

-- Movimentações de entrada, com produto, fornecedor, quantidade, preço unitário e
valor total.
select
  a.tipo
  , c.nome produto
  , b.nome fornecedor
  , a.quantidade
  , a.preco
  , a.quantidade * a.preco valor_total
from
  movimentos a
inner join
  pessoa b
on
  a.id_pessoa = b.id
inner join
  produto c
on
  a.id_produto = c.id
where
  a.tipo = 'e'
group by
  a.tipo
  , c.nome
  , b.nome
  , a.quantidade
  , a.preco ;

-- Movimentações de saída, com produto, comprador, quantidade, preço unitário e
valor total.
select
  a.tipo
  , c.nome produto
  , b.nome comprador
  , a.quantidade
  , a.preco
  , a.quantidade * a.preco valor_total
from
  movimentos a
inner join
  pessoa b
on
  a.id_pessoa = b.id
inner join
  produto c
on
  a.id_produto = c.id
where
  a.tipo = 's'
group by
  a.tipo
  , c.nome
  , b.nome
  , a.quantidade
  , a.preco ;

-- Valor total das entradas agrupadas por produto.
select
  a.tipo
  , c.nome produto
  , a.quantidade * a.preco valor_total
from
  movimentos a
inner join
  pessoa b
on
  a.id_pessoa = b.id
inner join
  produto c
on
  a.id_produto = c.id
where
  a.tipo = 'e'
group by
  a.tipo
  , c.nome
  , a.quantidade
  , a.preco ;
```

-- Valor total das saídas agrupadas por produto.

```
select
a.tipo
, c.nome produto
, a.quantidade * a.preco valor_total
from
movimentos a
inner join
pessoa b
on
a.id_pessoa = b.id
inner join
produto c
on
a.id_produto = c.id
where
a.tipo = 's'
group by
a.tipo
, c.nome
, a.quantidade
, a.preco ;
```

-- Operadores que não efetuaram movimentações de entrada (compra).

```
select *
from
usuarios a
where
a.id not in (
select
id_usuario
from
movimentos
where
tipo = 'e'
);
```

-- Valor total de entrada, agrupado por operador.

```
select
id_usuario
, quantidade * preco valor_total
from
movimentos
where
tipo = 'e'
group by
id_usuario
, quantidade
, preco
```

-- Valor total de saída, agrupado por operador.

```
select
id_usuario
, quantidade * preco valor_total
from
movimentos
where
tipo = 's'
group by
id_usuario
, quantidade
, preco
```

-- Valor médio de venda por produto, utilizando média ponderada.

```
select
id_produto
, sum(quantidade * preco) / sum(quantidade) valor_total
from
movimentos
where
tipo = 's'
group by
id_produto
, quantidade
, preco
```