

Nome do Aluno: Matheus Oliveira de Jesus

Matrícula: 2023.10.10087-8

Disciplina: Vamos Manter as Informações?

Turma: RPG0015

Este relatório tem como objetivo apresentar códigos e resultados referente aos exercícios solicitados em: Missão Prática | Nível 2 | Mundo 3.

Respostas: 2º Procedimento | Alimentação a Base

1 – Título da Prática: Alimentando a Base

2 – Objetivo da Prática:

- Identificar os requisitos de um sistema e transformá-los no modelo adequado;
- Utilizar ferramentas de modelagem para bases de dados relacionais;
- Explorar a sintaxe SQL na criação das estruturas do banco (DDL);
- Explorar a sintaxe SQL na consulta e manipulação de dados (DML);
- 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.

3 – Códigos solicitados

a)

```
INSERT INTO Usuarios (nome, senha)
```

```
VALUES ('op1', 'op1');
```

```
INSERT INTO Usuarios (nome, senha)
```

```
VALUES ('op2', 'op2');
```

```
SELECT * FROM Usuarios;
```

b)

```
INSERT INTO Produto (nome, quantidade, precoVenda)
```

```
VALUES ('Banana', 100, 5.00);
```

```
INSERT INTO Produto (nome, quantidade, precoVenda)
```

```
VALUES ('Laranja', 500, 2.00);
```

```
INSERT INTO Produto (nome, quantidade, precoVenda)
```

```
VALUES ('Manga', 800, 4.00);
```

```
SELECT * FROM Produto;
```

```
DELETE FROM Produto;
```

```
DELETE FROM MovimentoCompra;
```

```
DELETE FROM MovimentoVenda;
```

```
DELETE FROM Produto;
```

```
SET IDENTITY_INSERT Produto ON;
```

```
INSERT INTO Produto (id, nome, quantidade, precoVenda)
```

```
VALUES (1, 'Banana', 100, 5.00);
```

```
INSERT INTO Produto (id, nome, quantidade, precoVenda)
```

```
VALUES (2, 'Laranja', 500, 2.00);
```

```
INSERT INTO Produto (id, nome, quantidade, precoVenda)
VALUES (3, 'Manga', 800, 4.00);
```

```
SET IDENTITY_INSERT Produto OFF;
```

```
SELECT * FROM Produto;
```

4 – Resultado dos códigos executados:

The screenshot displays the SQL Server Enterprise Manager interface. On the left, the 'Pesquisador de Objetos' (Object Explorer) shows the database structure for 'NOTE_DO_MATHEUS\SQLEXPRESS (SQL Server 16.0.1130 - lojaa)'. The 'Tabelas' (Tables) folder is expanded, showing the 'dbo.Usuarios' table. On the right, the 'SQLQuery53.sql' window shows the executed queries:

```
INSERT INTO Usuarios (nome, senha)
VALUES ('op1', 'op1');

INSERT INTO Usuarios (nome, senha)
VALUES ('op2', 'op2');

SELECT * FROM Usuarios;
```

Below the queries, the 'Resultados' (Results) pane shows the output of the SELECT statement. The data is displayed in a table with columns 'id', 'nome', and 'senha'.

	id	nome	senha
1	1	op1	op1
2	2	op2	op2

The screenshot shows the SQL Server Enterprise Manager interface on the left, displaying the database structure of 'bancoLoja'. The right pane shows a SQL Query window with the following code:

```
INSERT INTO Produto (id, nome, quantidade, precoVenda)
VALUES (1, 'Banana', 100, 5.00);

INSERT INTO Produto (id, nome, quantidade, precoVenda)
VALUES (2, 'Laranja', 500, 2.00);

INSERT INTO Produto (id, nome, quantidade, precoVenda)
VALUES (3, 'Manga', 800, 4.00);

SET IDENTITY_INSERT Produto OFF;

SELECT * FROM Produto;
```

Below the query window, the 'Resultados' (Results) tab is active, showing the data inserted into the 'Produto' table:

id	nome	quantidade	precoVenda
1	Banana	100	5.00
2	Laranja	500	2.00
3	Manga	800	4.00

5 – Análise e Conclusão:

a. Quais as diferenças no uso de sequence e identity?

As principais diferenças entre sequence e identity no SQL Server são:

1 - Definição e Escopo:

- Identity: Propriedade de coluna em uma tabela que gera valores automaticamente.
- Sequence: Objeto independente que gera números sequenciais, podendo ser usado em várias tabelas.

2 - Flexibilidade:

- Identity: Gera valores automaticamente e não permite especificação ao inserir.
- Sequence: Permite obter o próximo valor e definir incrementos personalizados.

3 - Reinicialização:

- Identity: Reinicia ao truncar a tabela, sem controle manual.
- Sequence: Pode ser reiniciada com o comando ALTER SEQUENCE.

4 - Controle de Valores:

- Identity: Não permite controle direto dos valores gerados.
- Sequence: Oferece maior controle sobre os valores, permitindo uso conforme necessário.

5 - Usos Comuns:

- Identity: Usado para chaves primárias.
- Sequence: Usado quando um contador único compartilhado entre tabelas é necessário.

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

As chaves estrangeiras são fundamentais para a consistência do banco de dados porque:

- Integridade Referencial: Garantem que os relacionamentos entre tabelas sejam mantidos, evitando referências a dados inexistentes.
- Validação de Dados: Impedem a inserção de registros inválidos, assegurando que os dados relacionados estejam sempre disponíveis e corretos.
- Facilitam a Manutenção: Ao impor regras de relacionamento, ajudam a manter a estrutura do banco de dados organizada e compreensível.
- Cascata de Ações: Permitem ações automáticas (como DELETE ou UPDATE) em registros relacionados, assegurando que alterações não deixem dados órfãos.

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

- Seleção: Filtra linhas com base em uma condição.
- Projeção: Retorna colunas específicas de uma tabela.
- União: Combina resultados de duas tabelas, eliminando duplicatas.
- Diferença: Retorna linhas de uma tabela que não estão na outra.
- Produto Cartesiano: Combina todas as linhas de duas tabelas.
- Junção: Combina linhas de duas tabelas com base em uma condição.

Operadores do Cálculo Relacional:

- Cálculo Relacional de Tuplas: Utiliza variáveis de tuplas para descrever as condições para a seleção.
- Cálculo Relacional de Domínios: Usa variáveis de domínio para expressar consultas em forma de equações.

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

O agrupamento em consultas SQL é feito usando a cláusula GROUP BY, que junta linhas com valores iguais em colunas que você escolhe. É muito útil quando você quer calcular totais ou médias para grupos de dados, usando funções como SUM(), COUNT(), ou AVG().

Ponto importante: Você sempre precisa incluir no GROUP BY todas as colunas que não estão sendo agregadas na sua consulta. Isso ajuda a garantir que os resultados estejam corretos e que você não tenha confusão sobre quais dados estão sendo contados ou somados.