Universidade Estácio

Juazeiro da Bahia

Desenvolvimento Full Stack

Vamos manter as informações!

Relatório discente de acompanhamento

Nome: Gerson José de Almeida Júnior

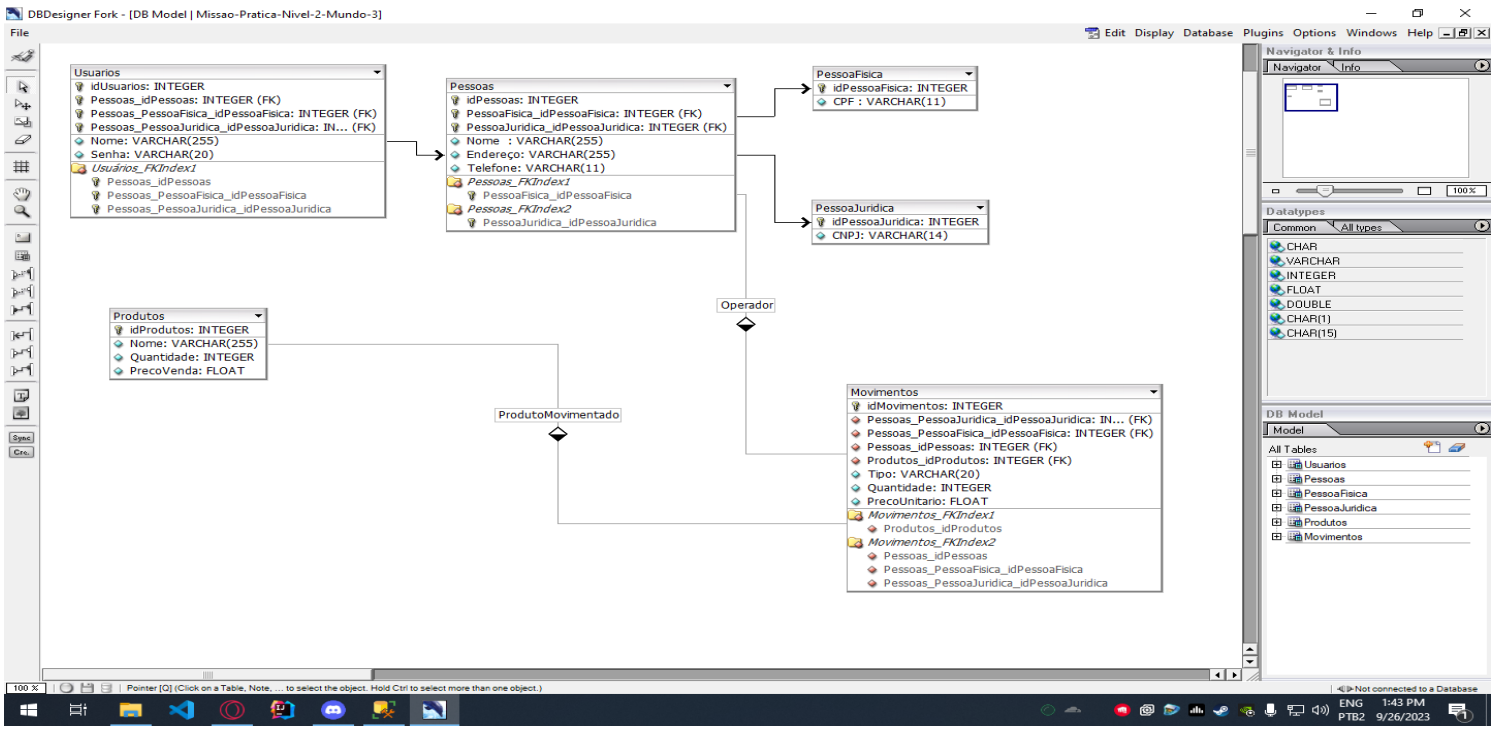
Número da Turma: 23.3

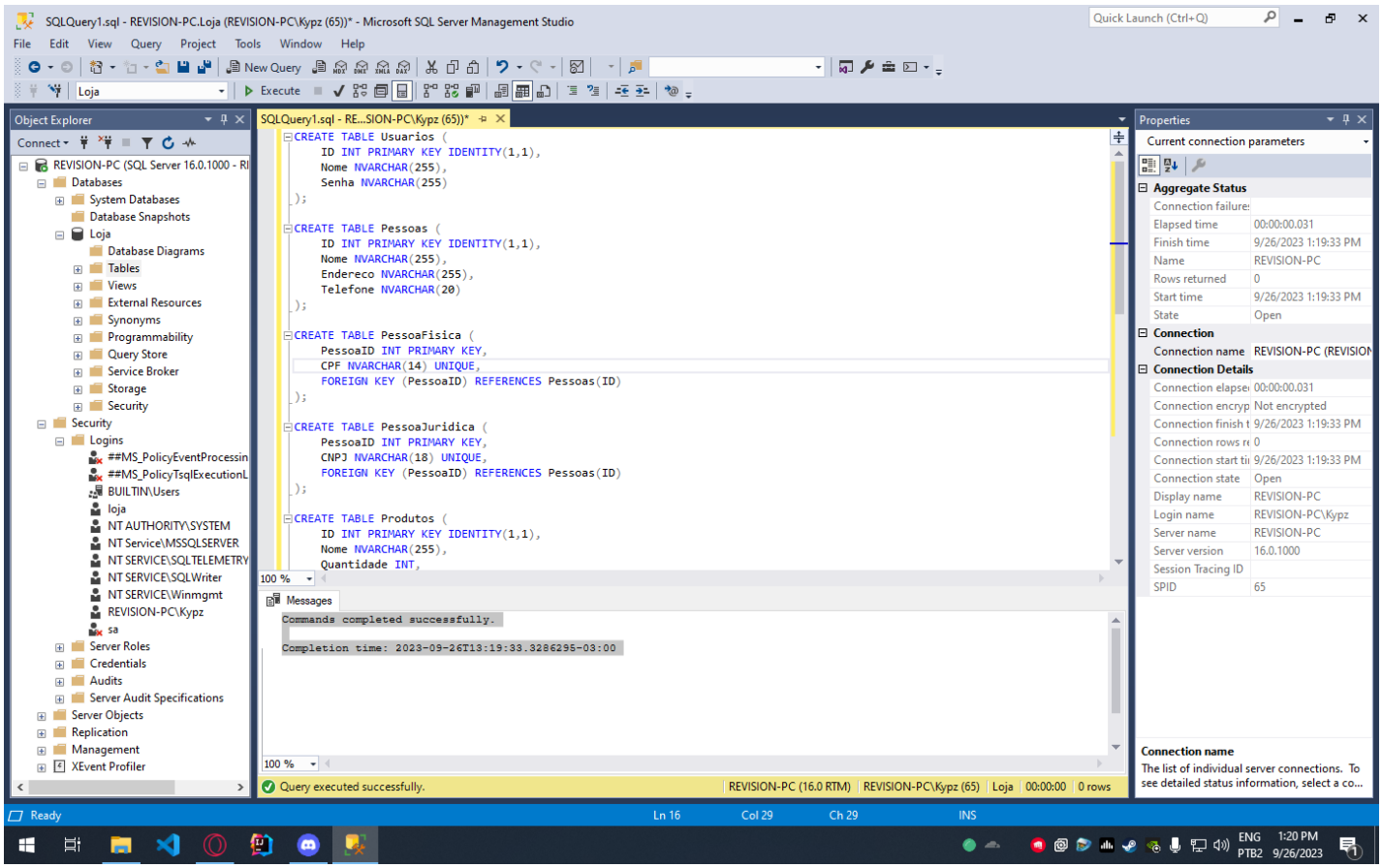
Semestre Letivo: 3° Semestre

Github: https://github.com/gersonjose9713/Missao-Pratica-Nivel-2-Mundo-3

Criando o Banco de Dados

Códigos Desenvolvidos:





Análise e Conclusão:

Como são implementadas as diferentes cardinalidades, basicamente 1X1, 1XN ou NxN, em um banco de dados relacional?

1X1 (Um-para-Um): Isso ocorre quando uma linha em uma tabela está relacionada a uma única linha em outra tabela e vice-versa. Para implementar isso, você pode adicionar uma chave estrangeira em uma das tabelas que referencia a outra tabela. Por exemplo, se você tiver uma tabela de "Pessoa" e outra tabela de "Endereço", cada pessoa pode ter um único endereço e cada endereço pode estar associado a uma única pessoa. A chave estrangeira seria na tabela de "Endereço", referenciando a tabela de "Pessoa" com uma coluna como "PessoaID".

1XN (Um-para-Muitos): Isso ocorre quando uma linha em uma tabela está relacionada a várias linhas em outra tabela, mas cada linha nesta última tabela está relacionada a apenas uma linha na primeira tabela. Isso é implementado com uma chave estrangeira na tabela "muitos" que faz referência à tabela "um". Por exemplo, em um sistema de pedidos, uma única ordem (tabela "Ordem") pode conter vários itens (tabela "ItemPedido"). A tabela "ItemPedido" teria uma chave estrangeira que aponta para a tabela "Ordem".

NxN (Muitos-para-Muitos): Isso ocorre quando várias linhas em uma tabela estão relacionadas a várias linhas em outra tabela. Para implementar isso, você geralmente precisa de uma terceira tabela intermediária, chamada de tabela de associação ou tabela de junção. Por exemplo, em um sistema de gerenciamento de cursos, várias disciplinas (tabela "Disciplina") podem ser frequentadas por vários estudantes (tabela "Estudante"). Para representar isso, você cria uma tabela de associação "Matrícula" que contém duas chaves estrangeiras, uma que referencia a tabela "Disciplina" e outra que referencia a tabela "Estudante".

Que tipo de relacionamento deve ser utilizado para representar o uso de herança em bancos de dados relacionais?

Para representar o uso de herança em bancos de dados relacionais, geralmente é recomendado usar uma abordagem conhecida como "Tabelas de Herança" ou "Tabelas Filhas". Nesse método, você cria uma tabela base que contém os campos e atributos comuns a todas as entidades envolvidas na herança. Em seguida, cria tabelas filhas que herdam da tabela base e adicionam campos específicos daquela entidade. Aqui está um exemplo:

Suponhamos que você tenha uma hierarquia de entidades: "Pessoa" (base), "PessoaFisica" e "PessoaJuridica" (filhas). A tabela base "Pessoa" conteria campos comuns a todas as pessoas, como nome, endereço e telefone. As tabelas filhas "PessoaFisica" e "PessoaJuridica" adicionariam campos específicos, como CPF e CNPJ, respectivamente.

Para representar esse relacionamento de herança, você cria uma chave estrangeira nas tabelas filhas que referencia a tabela base, usando uma coluna que serve como chave primária da tabela base.

Como o SQL Server Management Studio permite a melhoria da produtividade nas tarefas relacionadas ao gerenciamento do banco de dados?

O SQL Server Management Studio (SSMS) é uma ferramenta poderosa que melhora significativamente a produtividade nas tarefas relacionadas ao gerenciamento de banco de dados. Algumas maneiras pelas quais o SSMS melhora a produtividade incluem:

Interface Gráfica Amigável: O SSMS oferece uma interface gráfica intuitiva que permite criar, modificar e visualizar objetos de banco de dados de forma eficiente, sem a necessidade de escrever comandos SQL manualmente.

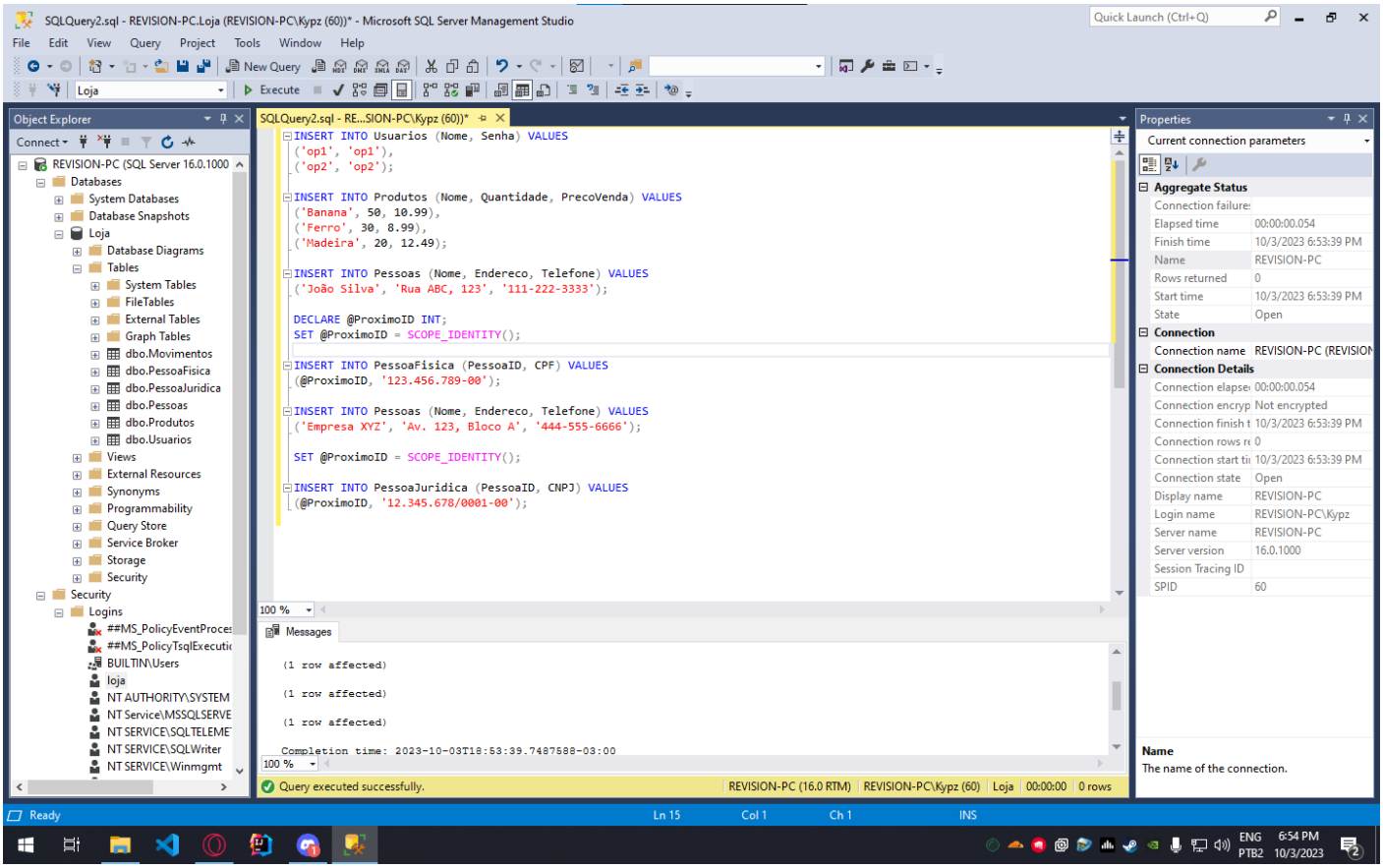
Editor de Consultas: O SSMS inclui um editor de consultas robusto com realce de sintaxe e recursos de auto-completar que facilitam a escrita de consultas SQL complexas.

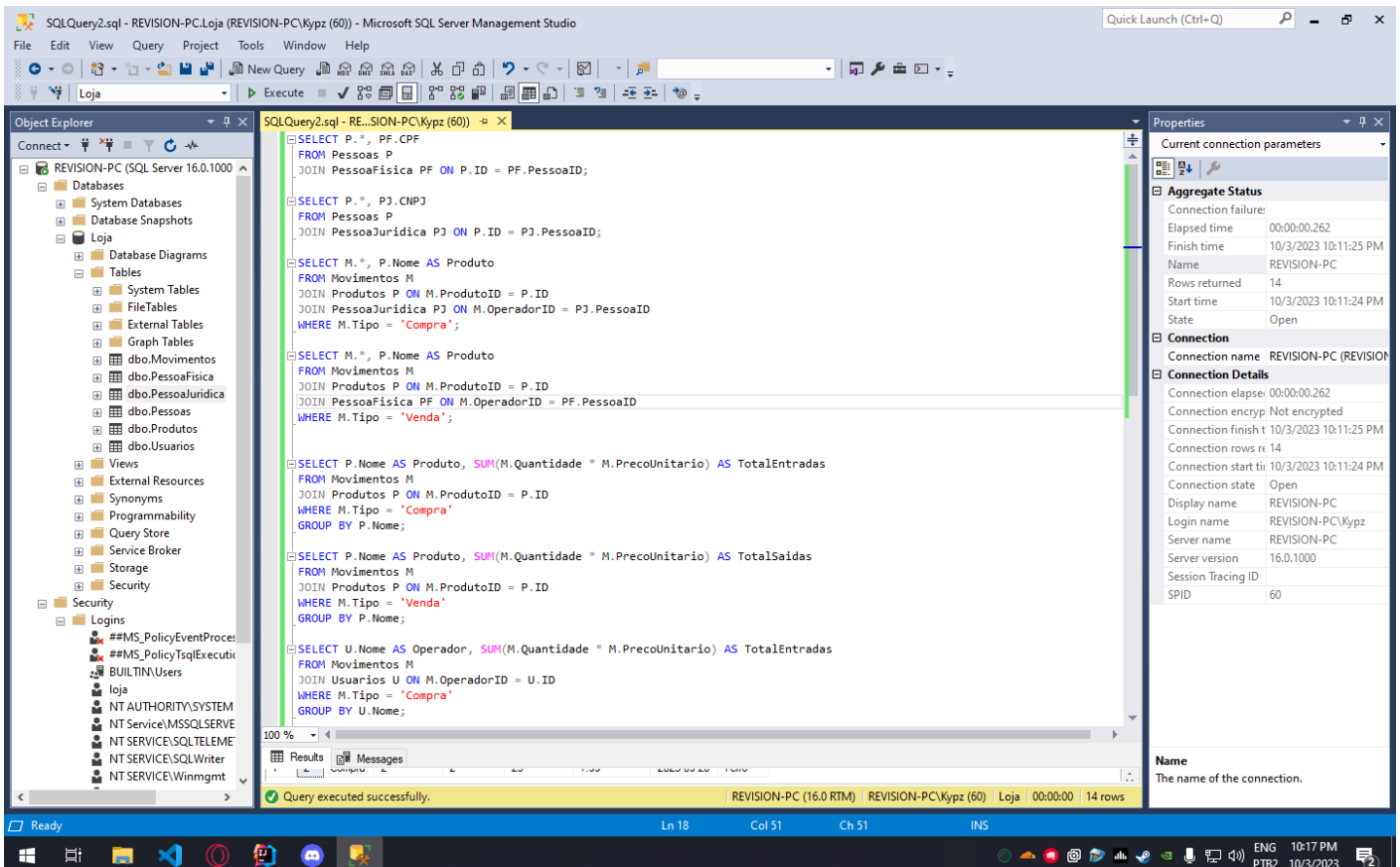
Administração e Manutenção: Você pode gerenciar facilmente a segurança, o desempenho, as cópias de segurança e outras tarefas administrativas do banco de dados usando as ferramentas fornecidas pelo SSMS.

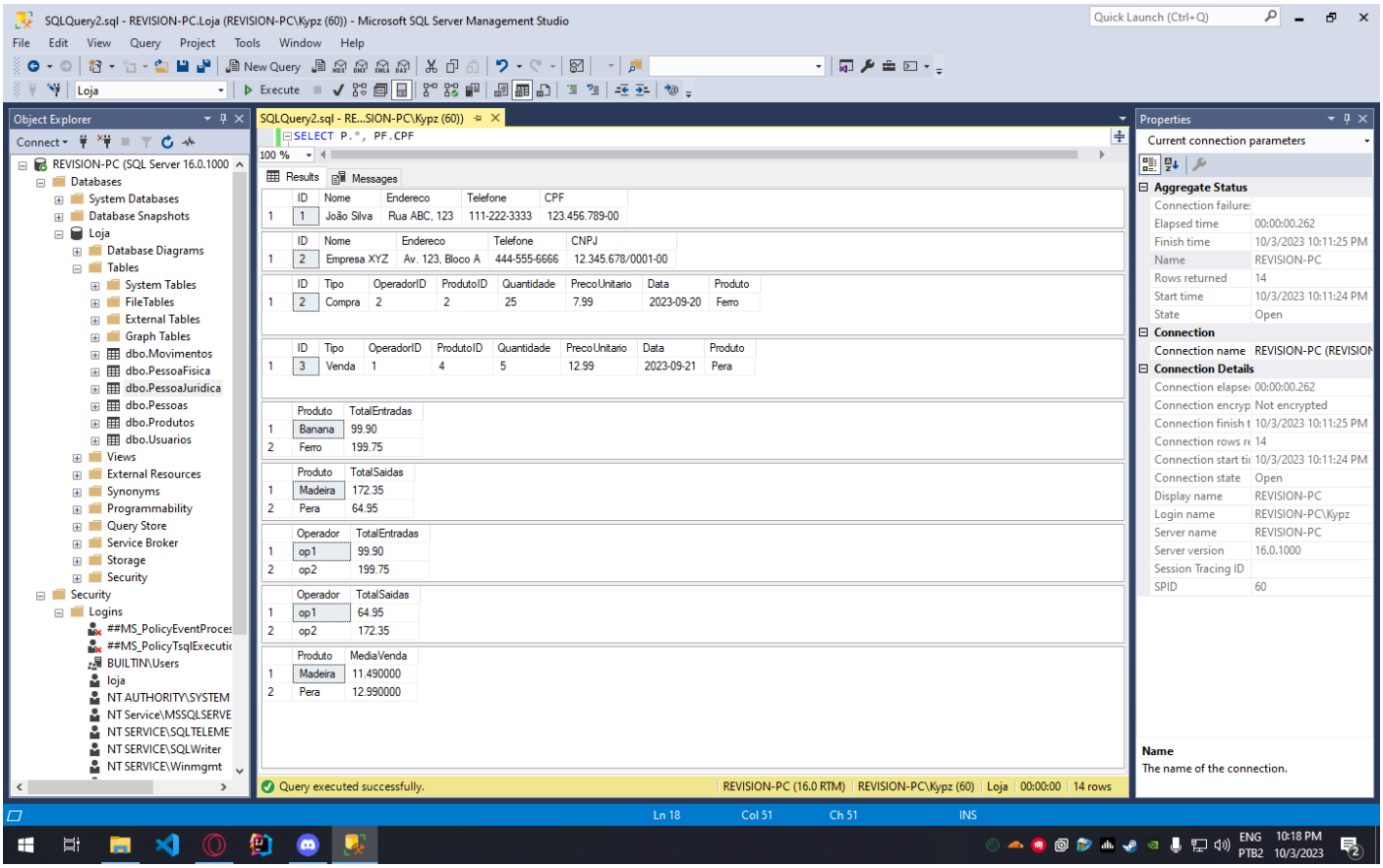
Monitoramento de Desempenho: O SSMS permite monitorar o desempenho do banco de dados, identificar gargalos e otimizar consultas e índices.

Alimentando a Base

Códigos Desenvolvidos:







Análise e Conclusão:

Quais as diferenças no uso de sequence e identity?

As principais diferenças no uso de sequências (sequence) e identidades (identity) em bancos de dados são:

Sequence: Uma sequência é um objeto de banco de dados que gera valores sequenciais, que podem ser usados em várias colunas de diferentes tabelas. Você precisa criar explicitamente uma sequência e, em seguida, referenciá-la em suas inserções. As sequências são mais flexíveis e podem ser compartilhadas entre várias tabelas. Elas são suportadas por sistemas de gerenciamento de banco de dados como o Oracle e o PostgreSQL.

Identity: A identidade (identity) é uma propriedade de coluna que é especificada ao criar a tabela. A coluna com identidade gera automaticamente valores incrementais à medida que as linhas são inseridas na tabela. O sistema de gerenciamento de banco de dados, como o SQL Server, gerencia a geração desses valores automaticamente. As identidades são mais simples de usar e geralmente estão vinculadas a uma única coluna em uma única tabela.

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

As chaves estrangeiras (foreign keys) são fundamentais para garantir a consistência e integridade dos dados em um banco de dados relacional. Sua importância reside nos seguintes pontos:

Manutenção de Relacionamentos: As chaves estrangeiras estabelecem relacionamentos entre tabelas, permitindo que os dados em diferentes tabelas estejam interconectados de maneira lógica e significativa.

Impedimento de Dados Órfãos: As chaves estrangeiras evitam a criação de registros "órfãos" em que não há correspondência em uma tabela relacionada. Isso mantém a integridade referencial dos dados.

Consistência de Dados: Elas garantem que os dados atendam a regras de negócios, como garantir que um cliente só possa fazer um pedido se existir como um registro na tabela de clientes.

Manutenção de Integridade: As chaves estrangeiras ajudam a manter a integridade dos dados durante operações de atualização ou exclusão, garantindo que as ações em cascata ocorram conforme necessário.

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

Na linguagem SQL, os operadores que pertencem à álgebra relacional incluem SELECT, PROJECT, JOIN (INNER, OUTER, etc.), UNION, INTERSECT, EXCEPT, etc. Esses operadores são usados para recuperar e manipular dados em tabelas de maneira relacional.

O cálculo relacional, por outro lado, não possui operadores correspondentes no SQL. Ele é uma linguagem que se concentra em descrever o que deve ser obtido, em vez de como obtê-lo. Os operadores no cálculo relacional incluem os operadores sigma (σ) para seleção e pi (π) para projeção, entre outros. Esses operadores não são diretamente usados em SQL, mas o SQL realiza essas operações de forma implícita.

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

O agrupamento em consultas SQL é realizado usando a cláusula GROUP BY. O objetivo do agrupamento é agregar dados e calcular valores resumidos com base em uma ou mais colunas. Quando você utiliza GROUP BY, as linhas são agrupadas com base nos valores das colunas especificadas e você pode usar funções de agregação, como SUM, COUNT, AVG, etc., para calcular valores agregados.

Requisito Obrigatório para GROUP BY: Quando você usa a cláusula GROUP BY em uma consulta SQL, todas as colunas que não estão sendo agregadas devem estar presentes na cláusula GROUP BY ou usadas em funções de agregação. Isso garante que a consulta seja logicamente válida, pois cada linha resultante terá um valor definido para cada coluna não agregada. É uma regra importante para evitar resultados ambíguos.

Por exemplo, se você está agrupando por "Produto" para calcular o total de vendas por produto, a coluna "Produto" deve estar na cláusula GROUP BY ou ser usada em funções de agregação, enquanto as colunas como "Quantidade" e "Preço Unitário" podem ser usadas em funções de agregação, como SUM.