Educação Profissional Paulista

Técnico em
Desenvolvimento de Sistemas

Secretaria da Educação

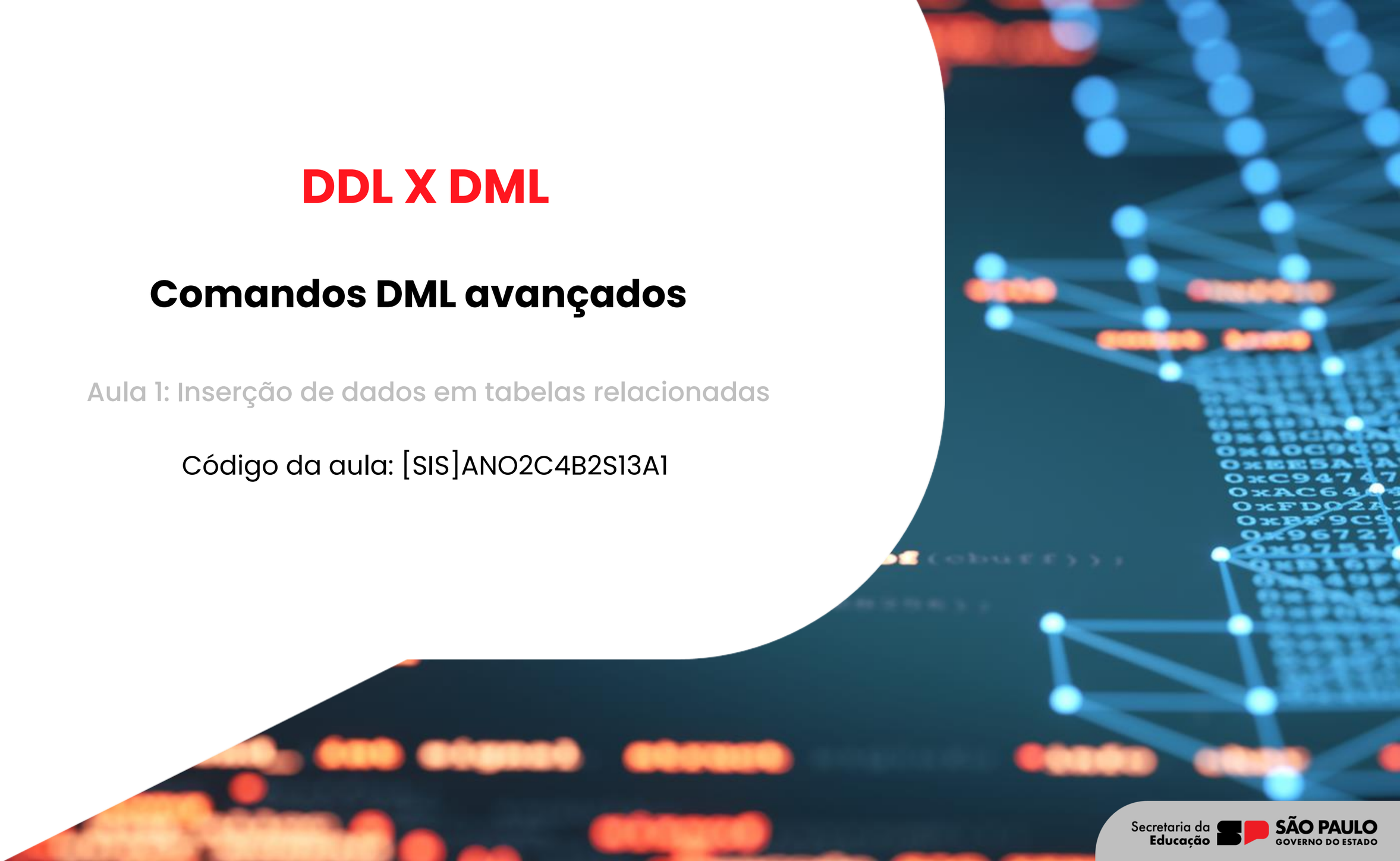
SÃO PAULO
GOVERNO DO ESTADO

DDL X DML

Comandos DML avançados

Aula 1: Inserção de dados em tabelas relacionadas

Código da aula: [SIS]ANO2C4B2S13A1



Mapa da Unidade 5 Componente 4

DDL X DML

semana 12

Diferenças entre DDL e DML

semana 13

Você está aqui!

Comandos DML avançados

semana 14

Análise e modelagem de dados

semana 19

DDL x DML

DDL X DML

Mapa da Unidade 5 Componente 4

Você está aqui!

13

Comandos DML avançados

Aula 1: Inserção de dados em tabelas relacionadas

Código da aula: [SIS]ANO2C4B2S13A1

Objetivos da aula

- Inserir dados em tabelas relacionadas utilizando comandos DML.

Recursos didáticos

- Recursos audiovisuais para exibição de vídeos e imagens;
- Caderno e caneta;
- Laboratório de informática.

Duração da aula

50 minutos.

Habilidades técnicas

- Inserir dados em tabelas relacionadas utilizando comandos DML.

Habilidades socioemocionais

- Atuar com cooperação ao trabalhar em equipe para gerenciar inserções de dados em tabelas relacionadas.

Colocando em **prática**

Inserção condicional com JOIN em banco de dados – Loja TechSmart

Inserção condicional com JOIN em banco de dados – Loja TechSmart

46 minutos

Em duplas

Materiais necessários

- Computador com acesso à internet;
- Caderno para anotações;
- Uma caneta.

Passo a passo

1. Leia com atenção cada uma das ações propostas no documento;
2. Antes de iniciar, realize esboços no papel para definir sua intenção e seu resultado;
3. Utilize as ferramentas de desenvolvimento disponíveis para a atividade.

Baixe o roteiro dessa atividade

Pause e **responda**

Qual comando SQL é utilizado para inserir dados em uma tabela a partir de outra utilizando JOIN?

- INSERT INTO ... SELECT ... FROM ... JOIN ...
- SELECT INTO ... JOIN ...
- JOIN INTO ... INSERT ... SELECT ...
- UPDATE ... JOIN ... SELECT ...

Pause e **responda**

Qual comando SQL é utilizado para inserir dados em uma tabela a partir de outra utilizando JOIN?

- ✓ **INSERT INTO ... SELECT ... FROM ... JOIN ...**
- ✗ **SELECT INTO ... JOIN ...**
- ✗ **JOIN INTO ... INSERT ... SELECT ...**
- ✗ **UPDATE ... JOIN ... SELECT ...**

Pause e **responda**

Em uma inserção condicional, qual das opções a seguir impede que dados duplicados sejam inseridos em uma tabela?

- WHERE
- GROUP BY
- HAVING
- NOT EXISTS

Registro

Pause e **responda**

Em uma inserção condicional, qual das opções a seguir impede que dados duplicados sejam inseridos em uma tabela?

- ✗ WHERE
- GROUP BY ✗
- ✗ HAVING
- NOT EXISTS ✓

Pause e **responda**

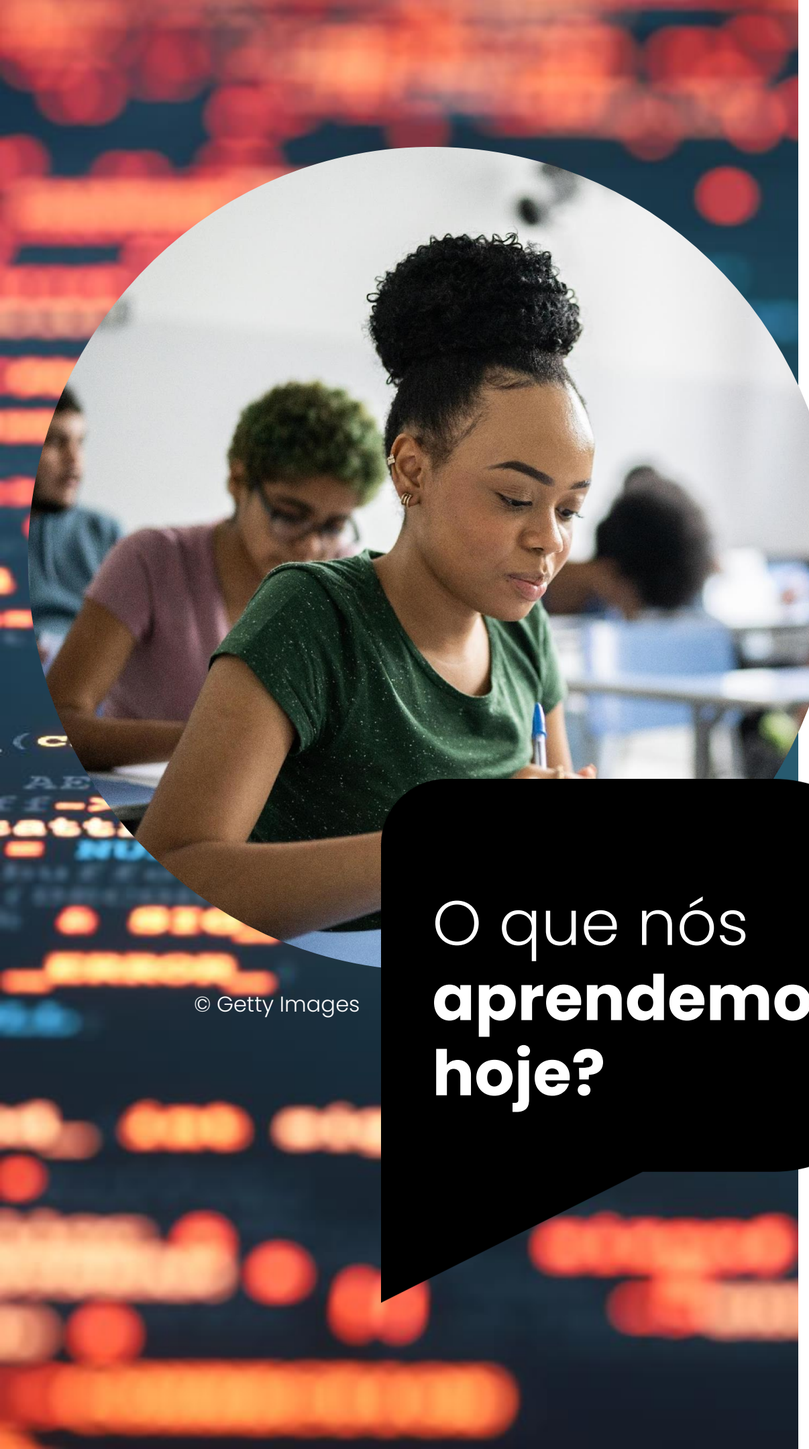
Qual comando é ideal para garantir que apenas clientes com pelo menos três pedidos sejam inseridos na tabela de premiados?

- WHERE COUNT(p.id_pedido) >= 3
- GROUP BY c.id_cliente HAVING COUNT(p.id_pedido) >= 3
- HAVING COUNT(c.id_pedido) >= 3
- WHERE p.id_pedido >= 3

Pause e **responda**

Qual comando é ideal para garantir que apenas clientes com pelo menos três pedidos sejam inseridos na tabela de premiados?

- ✗ WHERE COUNT(p.id_pedido) >= 3
- ✓ GROUP BY c.id_cliente HAVING COUNT(p.id_pedido) >= 3
- ✗ HAVING COUNT(c.id_pedido) >= 3
- ✗ WHERE p.id_pedido >= 3



O que nós **aprendemos hoje?**

Então, ficamos assim:

1. O comando INSERT INTO ... SELECT ... JOIN permite inserir dados em uma tabela a partir de outra utilizando junções para combinar informações;

2. Com o uso de NOT EXISTS, podemos evitar a inserção de dados duplicados, garantindo que os registros já presentes na tabela não sejam inseridos novamente;

3. GROUP BY, combinado com HAVING, é essencial para realizar inserções condicionais com base em agregações, como contar pedidos de clientes antes de inseri-los.

Referências da aula

Identidade visual: imagens © Getty Images

ALURA. **Comandos DML: manipulação de dados com MySQL**, [s.d.]a. Disponível em: https://www.alura.com.br/conteudo/mysql-dml-manipulacao-de-dados. Acesso em: 20 dez. 2024.

ALURA. **Juntando dados de várias tabelas**, [s.d.]b. Disponível em: https://www.alura.com.br/apostila-sql-e-modelagem-com-banco-de-dados/Juntando-dados-de-varias-tabelas?srsltid=AfmBOoovV_IpX-HVZOEfs8uc2ZsUV15nIzhxPr2hkHBXpwr3WyzgpYnb. Acesso em: 20 dez. 2024.

ELMASRI, R.; NAVATHE, S. B. **Sistemas de banco de dados**. São Paulo: Pearson, 2011.

OLIVEIRA, D.; DIAS, G. **Saiba tudo sobre SQL – A linguagem padrão para trabalhar com banco de dados relacionais!**, 29 nov. 2023. Disponível em: https://www.alura.com.br/artigos/o-que-e-sql?srsltid=AfmBOoqEtg3JKy_GMj7ePg3tuXniEHAfnw0xh-Acuy7e8Vnb4b0XxmxY. Acesso em: 20 dez. 2024.

Orientações ao professor

Slide 6

Orientações: professor, a seção **Colocando em prática** tem como objetivo aplicar os conhecimentos construídos durante a aula, incentivando os estudantes a pensar criticamente e de forma prática.

Tempo: 46 minutos.

Gestão de sala de aula: divida os alunos em duplas.

Condução da dinâmica: apresente a situação em que a loja precisa adicionar novos registros de clientes ou produtos, mas somente se certas condições forem atendidas, como verificar se o cliente já existe ou se o produto não está em duplicidade. Faça perguntas para gerar curiosidade, por exemplo: “Como garantir que não estamos inserindo dados duplicados ou desnecessários na loja?”.

O professor deve orientar os alunos a baixar o material de apoio, refletir, discutir e resolver os problemas propostos.

- **Metodologia ativa:**

Durante a atividade, os alunos deverão trabalhar em grupos para discutir como estruturar o comando de inserção condicional, explorando a utilização de JOIN e verificações condicionais com HAVING e NOT EXISTS. Após a execução, cada grupo deverá explicar o funcionamento do código e justificar as escolhas feitas.

Continua

Slide 6

Expectativas de respostas:

O comando SQL deve atender aos seguintes pontos principais:
1. Uso do JOIN para combinar as tabelas de **clientes** e **pedidos**, com base no campo id_cliente;
2. Condicional que verifica se o cliente é do tipo **online** e já fez pelo menos **três pedidos**;
3. Evitar duplicatas verificando se o cliente já está na tabela **clientes_premiados** utilizando o **NOT EXISTS**;
4. Inserir a data atual como data_premiacao para os clientes premiados.

Exemplo:

```
INSERT INTO clientes_premiados (id_cliente, nome, data_premiacao) SELECT c.id_cliente, c.nome, NOW() AS data_premiacao FROM clientes c JOIN
pedidos p ON c.id_cliente = p.id_cliente WHERE c.tipo_cliente = 'online' GROUP BY c.id_cliente HAVING COUNT(p.id_pedido) >= 3 AND NOT EXISTS
( SELECT 1 FROM clientes_premiados cp WHERE cp.id_cliente = c.id_cliente );
```

Explicação da resposta:

- **JOIN:** a consulta combina as tabelas de **clientes** e **pedidos** por meio do campo comum id_cliente. Isso permite contar os pedidos de cada cliente;
- **WHERE c.tipo_cliente = 'online':** essa condição filtra apenas os clientes do tipo **online**, conforme exigido no enunciado;
- **GROUP BY c.id_cliente:** agrupa os registros por cliente para contar a quantidade de pedidos de cada um;
- **HAVING COUNT(p.id_pedido) >= 3:** verifica se o cliente fez pelo menos **três pedidos**;
- **NOT EXISTS:** essa subconsulta verifica se o cliente já está na tabela de **clientes_premiados**. Caso o cliente já tenha sido premiado, ele será excluído da inserção para evitar duplicatas;
- **NOW():** insere a data e a hora atuais como a data de premiação no campo data_premiação.

Slide 7

Tempo: 3 minutos.

Slide 8

Feedback AVA:
Correto, pois o comando INSERT INTO ... SELECT ... FROM ... JOIN ... é usado para inserir dados em uma tabela a partir de uma consulta que utiliza JOIN para combinar múltiplas tabelas.

Feedback geral:

INSERT INTO ... SELECT ... FROM ... JOIN ... – Correto, pois o comando INSERT INTO ... SELECT ... FROM ... JOIN ... é usado para inserir dados em uma tabela a partir de uma consulta que utiliza JOIN para combinar múltiplas tabelas.

SELECT INTO ... JOIN ... – Incorreto. O comando SELECT INTO não é utilizado para inserção condicional em combinação com JOIN.

JOIN INTO ... INSERT ... SELECT ... – Incorreto. Não existe comando JOIN INTO em SQL.

UPDATE ... JOIN ... SELECT ... – Incorreto. O comando UPDATE é usado para modificar registros existentes, e não para inserir novos dados.
.

Slide 10

Feedback AVA:
Correto! O NOT EXISTS é utilizado para garantir que os dados que já estão na tabela de destino não sejam inseridos novamente, prevenindo duplicatas.

Feedback geral:

WHERE – Incorreto. O WHERE filtra dados em uma consulta, mas não impede a inserção de duplicatas.

GROUP BY – Incorreto. O GROUP BY agrupa resultados, mas não é suficiente para impedir duplicações durante a inserção.

HAVING – Incorreto. O HAVING funciona como uma extensão do GROUP BY para filtrar resultados agregados, mas não previne inserções duplicadas.

NOT EXISTS – Correto! O NOT EXISTS é utilizado para garantir que os dados que já estão na tabela de destino não sejam inseridos novamente, prevenindo duplicatas.

Slide 12

Feedback AVA:
Correto! GROUP BY com HAVING é a combinação correta para agrupar os registros por cliente e depois filtrar aqueles que têm três ou mais pedidos.

Feedback geral:
WHERE COUNT(p.id_pedido) >= 3 – Incorreto. O WHERE não funciona com agregações como COUNT(). O correto é usar HAVING para filtrar resultados agregados.

GROUP BY c.id_cliente HAVING COUNT(p.id_pedido) >= 3 – Correto! GROUP BY com HAVING é a combinação correta para agrupar os registros por cliente e depois filtrar aqueles que têm três ou mais pedidos.

HAVING COUNT(c.id_pedido) >= 3 – Incorreto. Embora HAVING seja necessário, a função de contagem deve se referir aos pedidos, e não aos clientes.

WHERE p.id_pedido >= 3 – Incorreto. Essa condição não está verificando a quantidade de pedidos de forma correta.

Slide 13

Orientações: professor, a seção **O que nós aprendemos hoje?** tem o objetivo de reforçar e esclarecer os conceitos principais discutidos na aula. Essa revisão pode ser uma ferramenta de avaliação informal do aprendizado dos estudantes, identificando áreas que podem precisar de mais atenção em aulas futuras.

Tempo: 1 minuto.

Gestão de sala de aula:

- Mantenha um tom positivo e construtivo, reforçando o aprendizado em vez de focar em correções;
- Seja direto e objetivo nas explicações para manter a atividade dentro do tempo estipulado;
- Engaje os estudantes rapidamente, pedindo confirmações ou reações breves às definições apresentadas.

Condução da dinâmica:

- Explique que esta parte da seção, “Então, ficamos assim”, é um momento de reflexão e esclarecimento sobre os conceitos abordados na aula;
- Informe que será uma rápida revisão para assegurar que os entendimentos dos estudantes estão alinhados com as definições corretas dos conceitos;
- Apresente o slide com a definição sintética de cada conceito principal discutido na aula, ampliando-os em forma de frases completas;
- Finalize resumindo os pontos principais e reiterando a importância de cada conceito e como ele se encaixa no contexto maior da aula;
- Reforce a ideia de que essa revisão ajuda a solidificar o entendimento dos estudantes e a prepará-los para aplicar esses conceitos em situações práticas.

Expectativas de respostas:

Os estudantes devem sair da aula com um entendimento claro e preciso dos conceitos principais. A atividade serve como uma verificação rápida do entendimento dos estudantes e uma oportunidade para corrigir quaisquer mal-entendidos.

Educação Profissional Paulista

Técnico em
Desenvolvimento de Sistemas

Secretaria da Educação

SÃO PAULO
GOVERNO DO ESTADO