

Educação Profissional Paulista

Técnico em
**Desenvolvimento
de Sistemas**

Comandos SQL (CRUD)

Comandos SQL (CRUD)

Aula 2: Atualização de dados (UPDATE)

Código da aula: [SIS]ANO2C4B2S10A2

Mapa da Unidade 4 Componente 4

Comandos SQL
(CRUD)

semana

7

Aplicações práticas
de SQL

semana

10

Você está aqui!

Comandos SQL (CRUD)

semana

6

Comandos SQL
avançados

semana

5

Introdução aos
comandos SQL

semana

11

Comandos
avançados de SQL
(CRUD)

**Comandos SQL
(CRUD)**

**Mapa da
Unidade 4
Componente 4**

Você está aqui!

10

Comandos SQL (CRUD)

Aula 2: Atualização de dados (UPDATE)

Código da aula: [SIS]C4B2S10A2



Objetivos da aula

- Atualizar dados em tabelas utilizando o comando UPDATE.



Recursos didáticos

- Recurso audiovisual para exibição de vídeos e imagens.
- Caderno e lápis.



Duração da aula

50 minutos



Habilidades técnicas

- Inserir dados em tabelas utilizando o comando UPDATE.



Habilidades socioemocionais

- Praticar a cooperação ao trabalhar em equipe para gerenciar inserções de dados.

Ponto de partida

Banco de dados

Você é responsável pelo banco de dados de uma empresa que tem uma tabela chamada Clientes. Nessa tabela, cada cliente contém informações como nome, e-mail e status da conta. Recentemente, a equipe de atendimento pediu que todas as contas inativas há mais de um ano sejam reativadas automaticamente.

Além disso, foi adicionada uma nova coluna na tabela Clientes, chamada `data_ultimo_login`, em que será necessário registrar a última data de login de cada cliente.

Também está em andamento uma campanha especial para oferecer descontos a clientes que fizeram compras nos últimos três meses. Esses descontos devem ser registrados na tabela Clientes, usando os dados da tabela Pedidos.

Agora, como você usaria o comando UPDATE para:

- ▶ reativar todas as contas inativas de clientes que estão há mais de um ano sem usar o serviço?
- ▶ atualizar a tabela Clientes com a data do último login de cada cliente na nova coluna `data_ultimo_login`?

Construindo
o **conceito**

Atualizando dados em tabelas no MySQL com o comando UPDATE

O comando **UPDATE** no MySQL é utilizado para modificar dados já existentes em uma ou mais colunas de uma tabela. Ele pode ser usado para fazer atualizações simples ou mais complexas, como atualizações condicionais ou baseadas em outras tabelas por meio de JOIN.

A sintaxe básica do comando **UPDATE** envolve a atualização de uma ou mais colunas de uma tabela com base em uma condição.

Sintaxe:

```
UPDATE nome_tabela  
SET coluna1 = valor1, coluna2 = valor2,...  
WHERE condição;
```

Continua...

Construindo
o **conceito**

Atualizando dados em tabelas no MySQL com o comando UPDATE

Exemplo: Vamos atualizar o preço do produto "Televisão" na tabela Produtos.

```
UPDATE Produtos  
SET preco = 1600.00  
WHERE nome_produto = 'Televisão';
```

O preço da televisão será atualizado para R\$ 1.600,00.



Tome nota:

A ausência da cláusula WHERE atualiza todas as linhas da tabela, o que pode levar a resultados indesejados.

Construindo
o **conceito**

Atualizando dados em tabelas no MySQL com o comando UPDATE

A atualização condicional no MySQL é realizada, utilizando a cláusula WHERE para especificar as condições que as linhas devem atender para serem atualizadas.

Exemplo: Vamos aumentar o preço de todos os produtos acima de R\$ 1.000,00 em 10%.

```
UPDATE Produtos  
SET preco = preco * 1.10  
WHERE preco > 1000.00;
```

Neste caso, o preço de todos os produtos cujo valor seja maior que R\$ 1.000,00 será atualizado em 10%.

Construindo
o **conceito**

Atualizando dados em tabelas no MySQL com o comando UPDATE

O UPDATE também pode ser combinado com JOIN para atualizar uma tabela com base em informações de outra tabela. Isso é útil quando você precisa atualizar registros que dependem de informações relacionadas entre diferentes tabelas.

Sintaxe de UPDATE com JOIN:

```
UPDATE tabela_destino  
JOIN tabela_origem ON  
    tabela_destino.coluna_relacionada =  
    tabela_origem.coluna_relacionada  
SET tabela_destino.coluna = novo_valor  
WHERE condição;
```

Construindo
o **conceito**

Atualizando dados em tabelas no MySQL com o comando UPDATE

Exemplo: Vamos atualizar o preço dos produtos de um fornecedor específico.

Suponha que o fornecedor "Fornecedor XYZ" ofereceu descontos, e você quer reduzir o preço dos produtos fornecidos por ele em 5%.

```
UPDATE Produtos p
JOIN Fornecedores_Produtos fp ON p.id_produto = fp.id_produto
JOIN Fornecedores f ON fp.id_fornecedor = f.id_fornecedor
SET p.preco = p.preco * 0.95
WHERE f.nome_fornecedor = 'Fornecedor XYZ';
```


Ser
sempre +

Situação

Desenvolvimento socioemocional em MySQL

Você e seus colegas foram contratados para gerenciar o banco de dados de uma empresa de tecnologia que trabalha com produtos, vendas e clientes. No entanto, o banco de dados precisa de atualizações e ajustes importantes para otimizar as operações e garantir a precisão dos dados.

A equipe será dividida em grupos para realizar essas atualizações. Cada grupo terá tarefas específicas envolvendo o comando UPDATE no MySQL, que devem ser concluídas em conjunto, promovendo discussões e tomadas de decisões colaborativas.

Situação fictícia produzida pela SEDUC-SP.

Ser
sempre +

Ação

Cada grupo terá que discutir e propor uma solução para os seguintes cenários:

- ▶ **Cenário 1:** O departamento de vendas identificou que alguns produtos antigos estão com preços desatualizados. Atualize, na tabela **Produtos**, o preço de todos os itens que estão há mais de dois anos no estoque (usando a coluna **data_entrada**), aumentando o valor em 5%.
- ▶ **Cenário 2:** Um fornecedor anunciou o recall de um lote de produtos. Atualize o status de todos os produtos desse fornecedor (usando a tabela **Fornecedores_Produtos**) para "em recall".
- ▶ Depois de realizar essas tarefas no MySQL, cada grupo deve discutir os desafios enfrentados durante a comunicação entre os membros e apresentar as soluções técnicas, explicando as escolhas feitas e os problemas superados.

Então ficamos assim...

- 1** Aprendemos a importância de atualizações de dados no MySQL, utilizando o comando UPDATE tanto em situações simples quanto em atualizações com JOIN e condicionais;
- 2** Entendemos que a cláusula WHERE é essencial para aplicar atualizações condicionais, evitando modificar todos os registros da tabela;
- 3** Vimos que a atualização com JOIN permite a atualização de dados em uma tabela com base nas informações relacionadas a outra tabela, usando JOINS para combinar dados e aplicando a lógica de atualização necessária.

O que nós
**aprendemos
hoje?**

© Getty Images

Saiba mais

Quer conhecer mais sobre este assunto?

Assista ao vídeo:

OTÁVIO MIRANDA. **UPDATE com vários JOINS (many-to-many) – Curso de SQL com MySQL – Aula 21.** Disponível em: <https://www.youtube.com/watch?v=cMwentKB2iE>. Acesso em: 26 nov. 2024.

Referências da aula

Identidade visual: imagens © Getty Images.

DEVMEDIA. **Uso de JOINS em instruções UPDATE e DELETE**, 2008. Disponível em: <https://www.devmedia.com.br/uso-de-joins-em-instrucoes-update-e-delete/9444>. Acesso em: 25 nov. 2024.

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

OTÁVIO MIRANDA. **UPDATE com vários JOINS (many-to-many) – Curso de SQL com MySQL – Aula 21**. Disponível em: <https://www.youtube.com/watch?v=cMwentKB2iE>. Acesso em: 26 nov. 2024.

Orientações ao professor

Slide 6



Orientações: Professor, a seção **Ponto de partida** aparece no início de cada aula, e tem como objetivos ativar o conhecimento prévio dos estudantes sobre o tema da aula e estimular seu pensamento crítico e suas habilidades comunicativas. Por meio de uma situação-problema ou exemplo próximo da realidade do estudante, pretende-se sair da abstração conceitual e promover um diálogo dinâmico para explorar hipóteses, soluções e compartilhar eventuais experiências que os estudantes já possam ter com os tópicos a serem abordados na aula. Também é um momento de engajá-los em relação ao tema da aula.



Tempo previsto: 8 minutos



Gestão de sala de aula:

Assegure que todos os estudantes tenham a oportunidade de participar. Se necessário, faça rodízio ou direcione perguntas a estudantes que estejam menos ativos, para garantir a participação de todos. Mantenha um ambiente de respeito, em que todas as opiniões são valorizadas, garantindo que todos se sintam confortáveis para expressar seus pontos de vista. Conclua a atividade resumindo as principais ideias discutidas e vinculando-as aos objetivos de aprendizagem da aula.



Expectativas de respostas:

```
1 – UPDATE Clientes  
SET status_conta = 'Ativo'  
WHERE DATEDIFF(CURDATE(), ultima_data_login) > 365;
```

- A função DATEDIFF calcula a diferença entre a data atual (CURDATE()) e a data do último login. Se essa diferença for maior que 365 dias, o campo status_conta será atualizado para 'Ativo'. Assim, todas as contas inativas há mais de um ano serão reativadas.

```
2 – UPDATE Clientes  
SET data_ultimo_login = CURDATE();
```

- Aqui, o comando UPDATE atualiza todos os registros da tabela **Clientes**, inserindo a data atual (obtida com a função CURDATE()) na nova coluna data_ultimo_login. Este é um exemplo de uma atualização simples.

Slide 7



Tempo previsto: 12 minutos



Expectativas de respostas:
Professor:

nome_tabela: O nome da tabela que terá seus dados atualizados.

coluna1, coluna2, ...: As colunas que serão atualizadas.

valor1, valor2, ...: Os novos valores a serem atribuídos.

condição (WHERE): Define quais linhas serão afetadas pela atualização.

Slide 10



Expectativas de respostas: Professor:

tabela_destino: A tabela que terá os dados atualizados.

tabela_origem: A tabela que será usada para comparar os dados.

coluna_relacionada: As colunas que relacionam as duas tabelas (normalmente chaves).

novo_valor: O valor que será aplicado na atualização.

condição (WHERE): Define quais linhas serão afetadas pela atualização.

Slide 11



Expectativas de respostas:

Professor: Aqui, a tabela **Produtos** será atualizada com base na relação de produtos e fornecedores, reduzindo o preço em 5% para todos os produtos fornecidos por "Fornecedor XYZ".

Slide 13



Tempo previsto: 25 minutos



Condução da dinâmica:

Na condução da dinâmica, o professor deve adotar uma postura facilitadora, orientando os alunos de maneira a promover tanto o desenvolvimento técnico quanto as competências socioemocionais. A seguir, algumas orientações para o professor:

O professor deve incentivar os alunos a discutir abertamente suas abordagens para resolver os problemas em MySQL. Perguntar diretamente: "Como vocês pensam que podemos resolver este problema?" ou "Alguém tem uma outra sugestão?" pode estimular a participação de todos.

A orientação deve destacar que não há respostas únicas no processo de tomada de decisões. O professor pode sugerir que os alunos explorem diferentes soluções, comparando os prós e contras de cada uma.

O professor pode orientar os grupos a distribuir responsabilidades, por exemplo, um aluno fica encarregado de organizar a documentação e outro, de revisar o código SQL. Isso promove o trabalho em equipe e assegura que todos participem ativamente.

Caso perceba que alguns alunos estão se destacando muito enquanto outros estão retraídos, o professor pode intervir de forma suave, perguntando a opinião daqueles que ainda não participaram. Por exemplo: "O que você acha dessa solução? Você tem alguma ideia adicional?"



Expectativas de respostas:

Cenário 1:

- **Competências socioemocionais trabalhadas:** Comunicação e colaboração. Os estudantes deverão debater como aplicar o UPDATE com base na data e ajustar o preço de maneira eficiente.

Continua ...

Slide 13



Expectativas de respostas:

Exemplo de consulta: UPDATE Produtos SET preco = preco * 1.05 WHERE DATEDIFF(CURDATE(), data_entrada) > 730;

Cenário 2:

• **Competências socioemocionais trabalhadas:** Tomada de decisões e resolução de problemas. Os estudantes deverão identificar os produtos afetados e decidir a melhor maneira de proceder.

Exemplo de consulta:

```
UPDATE Produtos p
JOIN Fornecedores_Produtos fp ON p.id_produto = fp.id_produto
JOIN Fornecedores f ON fp.id_fornecedor = f.id_fornecedor
SET p.status = 'em recall'
WHERE f.nome_fornecedor = 'Fornecedor XYZ';
```

Slide 14



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: 3 minutos



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 em forma de frases completas.
- Para cada conceito, faça uma breve comparação com as palavras-chave citadas pelos estudantes na dinâmica de nuvem de palavras.
- 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 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**