**Para saber mais: segurança em Views - Proteção e controle de dados**

Quando estamos lidando com bancos de dados, muitas vezes queremos mostrar informações para os usuários, mas sem expor dados sensíveis ou permitir que eles façam alterações que não deveriam. Aqui entra o conceito de "views", que são como janelas através das quais podemos ver os dados do banco de forma segura e controlada.

**O que são Views?**

Imagine que o banco de dados é uma casa cheia de cômodos (tabelas) com vários objetos (dados). Uma view é como uma janela específica que foi desenhada para permitir que as pessoas de fora vejam apenas uma parte da casa ou apenas alguns objetos específicos, sem poder entrar ou mexer em nada.

No mundo dos bancos de dados, uma view é uma tabela virtual que mostra dados de uma ou mais tabelas. Ela é criada através de uma consulta SQL e pode ser usada para:

* Simplificar consultas complexas;
* Limitar o acesso a dados sensíveis;
* Apresentar uma estrutura de dados personalizada para diferentes usuários.

**Uso de Views para Dados Sensíveis**

As views podem ser configuradas para proteger dados sensíveis. Por exemplo, se tivermos dados como salários ou informações médicas, podemos criar uma view que não inclui essas colunas:

**CREATE** **VIEW** view\_funcionario\_seguro **AS**

**SELECT** id, nome, departamento **FROM** funcionarios;

Isso significa que mesmo que alguém tenha acesso à view view\_funcionario\_seguro, essa pessoa não verá os salários dos funcionários.

**Motivação para Uso de Views**

A motivação para usar views em questões de segurança é simples: proteger informações e garantir que cada usuário acesse apenas o que precisa para suas tarefas. Isso ajuda a prevenir vazamentos de dados, acessos não autorizados e possíveis fraudes.

**Teoria e Boas Práticas**

* **Princípio do Menor Privilégio**: Sempre dê aos usuários o mínimo de acesso necessário.
* **Segregação de Dados**: Use views para separar dados sensíveis dos demais e limitar quem pode vê-los.
* **Views de Auditoria**: Crie views que facilitam o rastreamento de atividades para auditorias de segurança.

**Cuidados ao trabalhar com Views**

* **Use nomes claros:** Dê nomes que façam sentido para a "view", assim como você faz com pastas no seu computador.
* **Não abuse das views:** Criar muitas "views" pode tornar o sistema confuso e lento.
* **Documente as views:** Mantenha uma documentação de quais "views" existem e o que elas mostram, para que as pessoas saibam como usá-las.
* **Teste o desempenho:** Como "views" são baseadas em consultas, elas podem ser lentas se a consulta for complexa. Teste e otimize as consultas quando necessário.
* **Atualize quando necessário:** Às vezes, a estrutura do banco de dados muda. Quando isso acontecer, lembre-se de atualizar as "views" para refletir essas mudanças.

Lembre-se: a segurança dos dados é crucial, e as views são ferramentas poderosas para ajudar a manter essa segurança. Use-as com sabedoria e sempre mantenha-se atualizado sobre as melhores práticas de segurança em bancos de dados.

**Para saber mais: TRIGGER com DELETE e UPDATE**

Uma trigger em um banco de dados é um tipo de procedimento armazenado que é automaticamente executado em resposta a um evento específico, como uma instrução DELETE ou UPDATE sendo executada em uma tabela. As triggers são usadas para automatizar tarefas, impor regras de integridade, auditar ações e realizar outras ações automaticamente em um banco de dados.

Existem dois tipos principais de triggers relacionadas a instruções DELETE e UPDATE:

## Trigger com DELETE:

Uma trigger com DELETE é uma trigger que é acionada automaticamente quando uma instrução DELETE é executada em uma tabela específica.

Ela pode ser usada para realizar ações antes ou após a exclusão de dados da tabela. Por exemplo, você pode criar uma trigger DELETE para registrar informações sobre os registros excluídos em uma tabela de log antes de excluí-los.

A trigger com DELETE pode ser usada para impor regras de integridade referencial, como impedir a exclusão de registros em uma tabela pai que têm filhos relacionados em uma tabela filho.

Exemplo: Suponha que você tenha uma tabela de "Pedidos" e uma tabela de "ItensPedidos". Você pode criar uma trigger DELETE para garantir que os itens de pedido correspondentes sejam excluídos automaticamente quando um pedido for excluído.

## Trigger UPDATE:

Uma trigger com UPDATE é uma trigger que é acionada automaticamente quando uma instrução UPDATE é executada em uma tabela específica.

Ela pode ser usada para realizar ações antes ou após a atualização de dados na tabela. Por exemplo, você pode criar uma trigger UPDATE para registrar as alterações feitas em determinados campos de registro antes de atualizá-los.

A trigger com UPDATE pode ser usada para impor regras de validação ou de negócios, como impedir que um valor específico seja atualizado em uma coluna.

Exemplo: Em uma tabela de "Colaboradores", você pode criar uma trigger UPDATE para rastrear a data da última atualização do registro de funcionário e impedir que o salário de um funcionário seja reduzido em mais de 10%.

Em ambas as situações, as triggers DELETE e UPDATE são usadas para automatizar ações relacionadas às instruções DELETE e UPDATE no banco de dados. Elas podem ser poderosas, mas devem ser usadas com cuidado, pois podem afetar o desempenho e a integridade dos dados do banco de dados. É importante projetá-las e testá-las com atenção para garantir que elas atendam aos requisitos específicos do sistema.