



Trabalho 05

Table of contents

Introdução

Definição de controle de segurança e acesso

Importância do controle de acesso aos dados

Tipos de Controle de Acesso

Controle de acesso baseado em funções

Controle de acesso baseado em papéis

Controle de acesso baseado em políticas

DAC, MAC e RBAC

DCL (Data Control Language)

Definição e função do DCL

Como o DCL é utilizado para controlar o acesso aos dados

Comandos do DCL

Concessão de Privilégios

Comando GRANT para conceder privilégios

Exemplo

Revogação de Privilégios

Comando REVOKE para conceder privilégios

Exemplo

Controle de Acesso em Tabelas

DCL para controlar o acesso a tabelas específicas

Exemplos práticos de uso do GRANT e REVOKE para tabelas

Controle de Acesso em Colunas

DCL para controlar o acesso a colunas específicas dentro de uma tabela

Exemplos práticos de uso do GRANT e REVOKE para colunas

RLS (Row-Level-Security)

 Conceito de RLS

 Como controlar o acesso a linhas específicas dentro de uma tabela

 Benefícios

 Exemplo de uso

Melhores Práticas de Segurança e Controle de Acesso

 Apresentação de melhores práticas

 Ações possíveis

Conclusão

  Recapitação dos principais pontos

Introdução

▼ Definição de controle de segurança e acesso

Antes de mergulharmos no DCL, é importante entender alguns conceitos básicos. A segurança em bancos de dados refere-se às medidas e práticas adotadas para proteger os dados contra acesso não autorizado, modificação indevida ou vazamento. O controle de acesso, por sua vez, é o mecanismo que determina quem pode acessar os dados e quais ações eles podem realizar. É aqui que o DCL entra em cena, permitindo que você controle o acesso aos dados e as ações que os usuários podem realizar.

▼ Importância do controle de acesso aos dados

Agora, vamos falar sobre a importância do controle de acesso aos dados em bancos de dados. O controle de acesso desempenha um papel fundamental na segurança das informações armazenadas, garantindo que apenas usuários autorizados tenham acesso aos dados relevantes.

Pontos-chave sobre a importância do controle de acesso:

1. **Proteção contra acesso não autorizado:** impede o acesso de usuários não autorizados, protegendo dados confidenciais e garantindo segurança.
2. **Preservação da integridade dos dados:** evita modificações não autorizadas ou acidentais, mantendo a integridade dos dados.
3. **Cumprimento de regulamentações e leis de privacidade:** essencial para cumprir regulamentações, como o GDPR e LGPD.
4. **Mitigação de riscos internos:** limita acesso e monitora atividades para impedir danos e ações inapropriadas.

5. **Garantia da confidencialidade dos dados:** evita o acesso de pessoas não autorizadas a informações sensíveis, protegendo segredos comerciais e informações estratégicas.

Tipos de Controle de Acesso

▼ Controle de acesso baseado em funções

Existem diferentes abordagens para o controle de acesso em bancos de dados. O primeiro é o controle de acesso baseado em funções, no qual as permissões são atribuídas a funções específicas e os usuários são associados a essas funções.

▼ Controle de acesso baseado em papéis

Outra abordagem é o controle de acesso baseado em papéis, onde os usuários são atribuídos a papéis específicos, que possuem determinados privilégios.

▼ Controle de acesso baseado em políticas

E temos também o controle de acesso baseado em políticas, onde as permissões são definidas com base em políticas de segurança predefinidas.

▼ DAC, MAC e RBAC

Existem modelos de controle de acesso amplamente utilizados, tais como:

- **DAC (Discretionary Access Control):** neste modelo, os proprietários dos recursos têm controle sobre quem tem acesso a eles e quais ações podem ser executadas. As permissões são concedidas ou revogadas pelos proprietários dos recursos.
- **MAC (Mandatory Access Control):** neste modelo, o controle de acesso é determinado por políticas e regras predefinidas pelo sistema. As permissões são atribuídas com base em níveis de segurança, classificações ou categorias de dados.
- **RBAC (Role-Based Access Control):** neste modelo, o acesso é controlado com base nos papéis que os usuários desempenham em uma organização ou sistema. Os usuários são atribuídos a funções ou papéis específicos, e as permissões são definidas para cada papel.

Além dos modelos já mencionados (DAC, MAC e RBAC) e do RuBAC, existem outros modelos de controle de acesso, como o ABAC (Attribute-Based Access Control), que se baseia em atributos dos usuários, recursos e ambiente para

determinar o acesso. Outro modelo é o PBAC (Policy-Based Access Control), onde as políticas são usadas para controlar o acesso.

DCL (Data Control Language)

▼ Definição e função do DCL

DCL é uma linguagem utilizada para definir e controlar as permissões de acesso aos dados em um banco de dados. Sua função principal é garantir que apenas usuários autorizados tenham permissão para realizar operações específicas, como leitura, gravação, atualização ou exclusão de dados.

▼ Como o DCL é utilizado para controlar o acesso aos dados

O DCL é usado para controlar o acesso aos dados de diversas maneiras:

- Permissões de tabela: Com o DCL, é possível conceder ou revogar privilégios de tabela para usuários ou papéis. Isso permite controlar quais operações podem ser realizadas em determinadas tabelas.
- Permissões de coluna: Além do controle de acesso em tabelas, o DCL também permite controlar o acesso a colunas específicas dentro de uma tabela. Isso é útil quando se deseja limitar a visualização ou modificação de determinadas informações sensíveis.

▼ Comandos do DCL

O DCL possui dois principais comandos: GRANT e REVOKE, os quais serão detalhados a seguir.

Concessão de Privilégios

▼ Comando GRANT para conceder privilégios

O comando GRANT é usado para conceder permissões de acesso a usuários ou papéis. Por exemplo, podemos conceder o privilégio SELECT em uma tabela específica para um usuário, permitindo que ele consulte os dados.

▼ Exemplo

Para conceder acesso de leitura (SELECT) para o usuário "joao" em uma tabela chamada "clientes", utiliza-se o comando GRANT. A sintaxe básica do comando

é a seguinte:

```
GRANT <privilégio> ON <objeto> TO <usuário>;
```

Para o exemplo citado, a sintaxe específica seria:

```
GRANT SELECT ON clientes TO joao;
```

Com esse comando, o usuário "joao" terá permissão para executar consultas de leitura na tabela "clientes".

Revogação de Privilégios

▼ Comando REVOKE para conceder privilégios

O comando REVOKE é usado para revogar permissões previamente concedidas. Isso é útil quando é necessário remover o acesso de um usuário ou papel a determinados dados.

▼ Exemplo

Para revogar um privilégio de um usuário específico, basta utilizar o comando REVOKE, seguido do nome do privilégio (SELECT, UPDATE, DELETE, etc.), nome da tabela e nome do usuário. A sintaxe completa do comando é a seguinte:

```
REVOKE <privilégio> ON <objeto> FROM <usuário>;
```

Por exemplo, para revogar o privilégio SELECT da tabela "clientes" do usuário "maria", a sintaxe seria:

```
REVOKE SELECT ON clientes FROM maria;
```

Com esse comando, o usuário "maria" não terá mais permissão para realizar consultas de leitura na tabela "clientes".

Controle de Acesso em Tabelas

▼ DCL para controlar o acesso a tabelas específicas

Esses comandos permitem conceder e revogar privilégios de acesso a tabelas para usuários ou papéis no banco de dados. Aqui estão alguns exemplos práticos de como usar o GRANT e REVOKE para controlar o acesso a tabelas:

▼ Exemplos práticos de uso do GRANT e REVOKE para tabelas

- Concessão de privilégios para uma tabela:
 - Suponha que temos uma tabela chamada "produtos" e queremos conceder ao usuário "vendas" os privilégios de SELECT, INSERT e UPDATE nessa tabela.
- ```
GRANT SELECT, INSERT, UPDATE ON produtos TO vendas;
```
- Com esse comando GRANT, estamos concedendo ao usuário "vendas" os privilégios de SELECT (leitura), INSERT (inserção) e UPDATE (atualização) na tabela "produtos". Agora, o usuário "vendas" terá permissão para executar essas operações na tabela.
- Revogação de privilégios para uma tabela:
    - Suponha que tenhamos concedido previamente ao usuário "financeiro" o privilégio de DELETE (exclusão) na tabela "transacoes" e agora desejamos revogar esse privilégio.

```
REVOKE DELETE ON transacoes FROM financeiro;
```

- Com esse comando REVOKE, estamos revogando do usuário "financeiro" o privilégio de DELETE na tabela "transacoes". Após a execução do comando, o usuário "financeiro" não terá mais permissão para excluir registros nessa tabela.

# Controle de Acesso em Colunas

## ▼ DCL para controlar o acesso a colunas específicas dentro de uma tabela

Vamos explorar o controle de acesso em colunas específicas dentro de uma tabela usando os comandos GRANT e REVOKE no DCL. Esses comandos permitem conceder e revogar privilégios de acesso a colunas para usuários ou papéis no banco de dados. Aqui estão alguns exemplos práticos de como usar o GRANT e REVOKE para controlar o acesso a colunas:

### ▼ Exemplos práticos de uso do GRANT e REVOKE para colunas

Exemplo 1: Concessão de privilégios para colunas específicas

```
GRANT SELECT (nome, telefone) ON clientes TO atendimento;
```

Com esse comando, o usuário "atendimento" terá permissão para visualizar apenas as colunas "nome" e "telefone" da tabela "clientes".

Exemplo 2: Revogação de privilégios de colunas específicas

```
REVOKE SELECT (salario) ON funcionarios FROM gerente;
```

Com esse comando, o usuário "gerente" não terá mais permissão para visualizar o salário dos funcionários na tabela "funcionarios".

## RLS (Row-Level-Security)

### ▼ Conceito de RLS

RLS (Row-Level Security) ou Segurança por Nível de Linha é um recurso utilizado em bancos de dados para controlar o acesso a linhas específicas dentro de uma tabela com base em condições definidas. Esses critérios podem ser definidos por meio de expressões lógicas que envolvem colunas da tabela ou informações do usuário.

### ▼ Como controlar o acesso a linhas específicas dentro de uma tabela

Ao implementar o RLS, é possível definir condições para filtrar as linhas visíveis para cada usuário. Por exemplo, um administrador de vendas pode visualizar apenas as linhas relacionadas aos produtos que ele está autorizado a vender,

enquanto um gerente de departamento pode visualizar apenas as linhas relacionadas aos funcionários de seu departamento.

## ▼ Benefícios

O uso do RLS traz diversos benefícios, incluindo:

- Segurança granular: O RLS permite controlar o acesso a nível de linha, garantindo que cada usuário veja apenas os dados relevantes para suas responsabilidades.
- Conformidade com regulamentações: O RLS auxilia no cumprimento de regulamentações de privacidade e segurança de dados, como o GDPR, ao restringir o acesso apenas às informações necessárias.
- Simplificação da lógica de aplicação: O RLS facilita a implementação de políticas de segurança, reduzindo a complexidade da lógica de controle de acesso na aplicação.

## ▼ Exemplo de uso

```
ALTER TABLE contas ENABLE ROW LEVEL SECURITY;

CREATE POLICY gerentes_de_contas ON contas TO gerentes
 USING (gerente = current_user);
```

- O exemplo apresentado mostra a criação de uma tabela chamada "contas" com três colunas: "gerente", "empresa" e "email\_contato".
- Em seguida, é habilitado o recurso de Row-Level Security (RLS) na tabela "contas".
- É criada uma política de segurança chamada "gerentes\_de\_contas" que permite que somente os gerentes tenham acesso às linhas da tabela onde o valor da coluna "gerente" é igual ao nome do gerente atualmente logado no sistema.
- Assim, cada gerente só pode visualizar as linhas de "contas" que correspondem aos seus próprios clientes ou contas gerenciadas.

# Melhores Práticas de Segurança e Controle de Acesso

## ▼ Apresentação de melhores práticas

Além de conhecer as funcionalidades do DCL, é importante seguir algumas melhores práticas para garantir a segurança e o controle de acesso adequados em bancos de dados.

## ▼ Ações possíveis

Para manter a segurança dos dados armazenados, é importante implementar uma série de práticas recomendadas.

Aqui estão algumas das melhores práticas para garantir a segurança e o controle de acesso adequados em bancos de dados:

- **Mantenha o software atualizado:** É fundamental manter o software atualizado. Isso inclui não apenas o sistema operacional, mas também todos os softwares instalados, como navegadores, editores de texto e planilhas eletrônicas. A maioria dos softwares tem atualizações regulares que corrigem erros e falhas de segurança, por isso é importante instalá-las assim que estiverem disponíveis.
- **Implemente auditorias regulares:** É importante implementar auditorias regulares para monitorar atividades suspeitas. Isso pode ajudar a identificar tentativas de invasão ou acesso não autorizado aos dados armazenados. As auditorias também podem ajudar a identificar problemas de segurança e a tomar medidas preventivas antes que ocorram violações.
- **Faça backup regular dos dados:** Outra prática importante é o backup regular dos dados. Isso garante que, em caso de perda de dados devido a problemas técnicos ou violações de segurança, os dados possam ser recuperados com facilidade. Os backups devem ser armazenados em locais seguros e fora do alcance de pessoas não autorizadas.

Ao implementar essas práticas recomendadas, é possível reduzir significativamente o risco de violações de segurança e manter a integridade dos dados armazenados. Além disso, a implementação dessas práticas pode ajudar a garantir a conformidade com as regulamentações de segurança de dados aplicáveis.

# Conclusão

## ▼ Recapitação dos principais pontos

Chegamos ao fim da apresentação sobre segurança e controle de acesso em bancos de dados. Espero que tenham compreendido a importância desses aspectos na proteção dos dados e como o DCL, junto com os modelos de controle de acesso, pode ser utilizado para garantir a segurança adequada. Lembrem-se de implementar as melhores práticas de segurança e controle de acesso em suas organizações para proteger seus bancos de dados.

Agradecemos pela atenção!

---