1. Diferença entre GRANT e REVOKE

- **GRANT:** Concede permissões específicas a usuários para executar operações no banco.
- **REVOKE:** Remove permissões previamente concedidas a usuários.

Exemplos:

- Usar `GRANT SELECT ON banco.* TO 'usuario'@'host';` para permitir que um usuário apenas leia dados.
- Usar `REVOKE INSERT ON banco.* FROM 'usuario'@'host';` para retirar permissão de inserir dados.

---

2. Princípio do menor privilégio

É o conceito de conceder aos usuários apenas as permissões necessárias para realizar suas tarefas, nada além disso. É importante para minimizar riscos de uso indevido, erros ou ataques dentro do banco.

---

3. Importância de senhas fortes

Senhas fortes evitam acesso não autorizado, protegendo os dados. Características de uma senha segura:

- Mínimo de 8 caracteres;
- Combinação de letras maiúsculas, minúsculas, números e símbolos;
- Não conter informações pessoais;
- Não ser repetida em diferentes sistemas.

---

4. Evitar usuários compartilhados

Usuários compartilhados dificultam o controle de responsabilidades e auditoria, além de aumentar o risco de vazamento de credenciais e uso indevido.

---

5. Riscos de permissões desnecessárias e como evitá-los

Permissões excessivas podem permitir exclusão, alteração ou vazamento de dados. Evita-se isso aplicando o princípio do menor privilégio e revisando periodicamente os acessos.

---

6. Restrição de acesso por host

Limitar usuários a acessarem o banco apenas de hosts específicos reduz o risco de conexões não autorizadas por máquinas desconhecidas ou externas.

---

7. Permissões comuns em MySQL

- **SELECT:** Ler dados da tabela. Exemplo: usuário que consulta relatórios.
- **INSERT:** Inserir novos dados. Exemplo: cadastro de novos clientes.
- **UPDATE:** Atualizar dados existentes. Exemplo: alterar status de pedidos.
- **DELETE:** Excluir dados. Exemplo: remover registros obsoletos.

---

8. Auditoria de usuários e permissões

Consiste em monitorar e registrar as permissões concedidas e ações dos usuários no banco. É importante para identificar acessos indevidos, manter a segurança e cumprir normas.

---

9. Práticas ao conceder permissões em ambiente multiusuário

- Conceder somente o necessário;
- Usar grupos ou roles para facilitar gerenciamento;
- Revisar permissões regularmente;
- Usar autenticação forte;

- Registrar e monitorar acessos.

---

10. Configuração em ambiente de produção com desenvolvedores e administradores

- Criar usuários com permissões diferenciadas:
    - Desenvolvedores: acesso restrito a bancos de desenvolvimento, permissão para leitura e escrita limitada;
    - Administradores: permissões amplas para gerenciar banco, backups e configurações;
- Usar autenticação forte e restrição por host;
- Monitorar e auditar acessos;
- Segmentar ambientes (produção, teste, desenvolvimento) para evitar interferência.