

Complementos de Bases de Dados

2021/2022

Licenciatura em Engenharia Informática



Laboratório 5 – Backup e Segurança

Objetivos:

- *Backups*
 - Criação de *backups*.
 - Simulação de falhas no sistema e restauro de dados.
- *Segurança*
 - Criação de utilizadores
 - Definição de papéis/permisões para utilizadores
 - Encriptação

Anexos:

- [Tutorial Backup and Restore using SQL Server Management Studio 2019.](#)

Etapa 1: Backup & Restore

1.1) Criar uma base de dados de teste e definir modo de recuperação

1. Criar uma base de dados com o nome *Lista_Cliente* e com um *primary filegroup* de 50MB e um ficheiro de *Log* com 20MB.
2. Definir o modelo de recuperação como *Full*.
3. Criar a seguinte tabela:

```
CREATE TABLE Cliente (  
    ClienteID          INTEGER NOT NULL IDENTITY(1, 1),  
    CliPrimeiroNome    VARCHAR(50) NOT NULL,  
    CliUltimoNome      VARCHAR(100) NOT NULL,  
    CliEmail           VARCHAR(255) NOT NULL,  
    CliDataNasc        DATETIME NULL,  
    CliTelem           VARCHAR(20) NULL,  
    CliEmpresa         VARCHAR(150) NULL,  
    PRIMARY KEY (ClienteID)  
);
```

4. Crie o seguinte *Stored Procedure*:

```
CREATE PROCEDURE InserirVarios (@Inicio int, @Fim int)
AS
BEGIN

    DECLARE @Contador INT = @Inicio
    WHILE (@Contador <= @Fim)
    BEGIN
        INSERT INTO
            Cliente
        (
            [CliPrimeiroNome], [CliUltimoNome], [CliEmail],
            [CliDataNasc], [CliTelem], [CliEmpresa]
        )
        VALUES
        (
            'Primeiro' + convert(varchar, @Contador),
            'Ultimo' + convert(varchar, @Contador),
            'email' + convert(varchar, @Contador) + '@dominio.pt',
            '19850611',
            '919191919',
            'Empresa' + + convert(varchar, @Contador)
        )
        SET @Contador = @Contador + 1
    END
END
GO
```

5. Recorrendo ao procedimento anterior, insira 100 registos na base de dados:

Etapa 2: Backup/Restore (Cenários)

1.2) Cenário 1

6. Proceda a um *backup* completo da base de dados.
7. Insira mais 20 registos na base de dados.
8. Faça agora um *backup* diferencial à base de dados.
 - a. Verifique o espaço ocupado por cada um dos *backups*.
 - b. O que pode concluir? Perante os dois tipos de *backup*, os resultados obtidos são os esperados?
9. Faça a simulação de uma falha no sistema, da seguinte forma:
 - a. Desligue o servidor e serviço do SQL Server.
 - b. Mude o nome do ficheiro definido no *primary file group*.
 - c. Ligue novamente o SQL Server e confirme a ausência dos dados.
10. Teste o restauro do sistema e verifique a informação recuperada. Comente os dados que foram recuperados.

1.3) Cenário 2

11. Insira mais 20 registos na base de dados.
12. Faça um *backup* do *transational log*.
13. Insira mais 20 registos na base de dados.
14. Faça a simulação de uma falha no sistema (ver passo 9).
15. Repita o restauro do sistema.
 - a. Comente os dados que foram recuperados.
 - b. O que deve fazer para recuperar toda a informação?
16. Repita o restauro do sistema, mas recuperando o *tail de logs*. Comente os dados que foram recuperados.
17. **Exercício adicional:** Faça novas experiências de *backup* combinando os 3 tipos.

Obs.: tipicamente, os *backups full* são feitos com menor regularidade, seguidos de *backups* diferenciais com uma maior frequência e, finalmente, os backups *transactional log* são efetuados com uma frequência ainda maior.

Etapa 3: Segurança & Encriptação

Crie 2 novas bases de dados. Considere que a primeira base de dados pertence ao departamento de faturação de um Operador de Telecomunicações e que a segunda base de dados pertence ao departamento telemarketing, dentro da mesma empresa. Considerações:

- O departamento de faturação faculta a tabela dos clientes da sua base de dados ao departamento de telemarketing.
- Na primeira base de dados (departamento de faturação), crie uma tabela de clientes, idêntica à criada na etapa 1, insira alguns registos de teste.

3.1) Definição de Utilizadores (USERS)

1. Tendo presente as duas bases de dados do Operador de Telecomunicações. Crie o schema *[Developer_Schema]*.
2. Crie 2 tabelas, *[table1]* e *[table2]*, com as duas 2 colunas indicadas abaixo:
 - *[id]* *int IDENTITY(1,1) NOT NULL*
 - *[name]* *varchar(50) NULL*Tenha em atenção que a *[table2]* deverá pertencer ao schema *[Developer_Schema]*.
3. Crie 3 logins com a seguinte designação: *UserS1*, *UserS2*, *UserS3*.

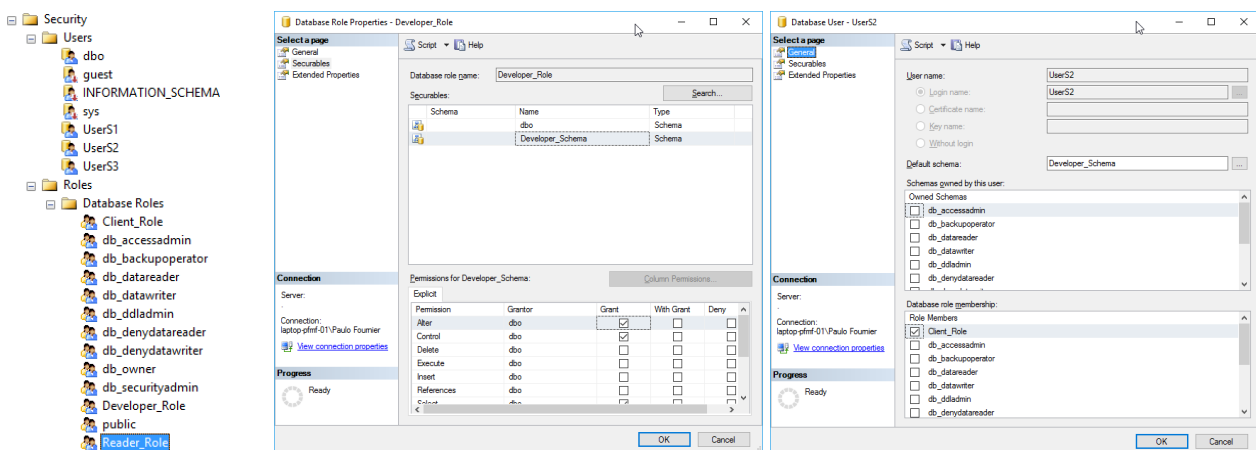
Sintaxe da instrução SQL: `CREATE LOGIN [UserS1] (...)`
4. Crie 3 utilizadores, um para cada login, com o schema *[Developer_Schema]* pré-definido.

3.2) Definição de Permissões (Roles)

5. Defina 3 tipos de permissão (através de script) para acesso à BD (schema *dbo*), da seguinte forma:
 - **Developer:** Tem todo o tipo de acesso aos dados (ver, inserir, modificar, etc.), e pode criar/apagar tabelas no schema *Developer_Role*.
 - **Client:** Tem acesso para consulta/inserção/modificação de dados sobre tabelas existentes.
 - **Reader:** Só pode consultar os dados existentes.
6. Atribua cada uma das roles ao respetivo utilizador definido no ponto 3.1.3)

3.3) Acessos e Restrições

7. Verifique graficamente a configuração das Roles, bem com a associação das mesmas a cada utilizador



8. Faça login com o utilizador **UserS1** e:
 - a. Insira uma linha na tabela [table1] e outra na [table2]
 - b. Crie a tabela [table3] com este utilizador com o schema [*dbo*]
 - c. Crie a tabela [table3] com este utilizador com o schema [*Developer_Schema*]
9. Faça login com o utilizador **UserS2** e:
 - a. Insira uma linha na tabela [table1] e outra na [table2]
 - b. Crie a tabela [table4] com este utilizador com o schema [*dbo*]
 - c. Crie a tabela [table4] com este utilizador com o schema [*Developer_Schema*]

10. Faça login com o utilizador **UserS3** e:

- a. Insira uma linha na tabela [table1] e outra na [table2]
- b. Liste os valores inseridos em cada tabela

Comente, de forma fundamentada, os resultados obtidos nas 3 alíneas anteriores.

Etapa 4: Encriptação

Utilizando o seu *user* por defeito (i.e., *user default*).

11. Insira algumas linhas na tabela [table1]
12. Crie um *master key* com uma chave à sua escolha
13. Crie um certificado de encriptação
14. Crie uma chave simétrica com o algoritmo AES_256 utilizando o certificado definido na alínea anterior
15. Altere a tabela [table1] e adicione a coluna [*EncryptName*] do tipo VARBINARY(256)
16. Atualize a nova coluna com os dados encriptados da coluna [*name*] criados com o certificado e chave definidos anteriormente
17. Liste a tabela [table1] e comente o resultado obtido
18. Liste a tabela [table1] descriptando a coluna [*EncryptName*] de forma a devolver o valor original

(fim de enunciado)