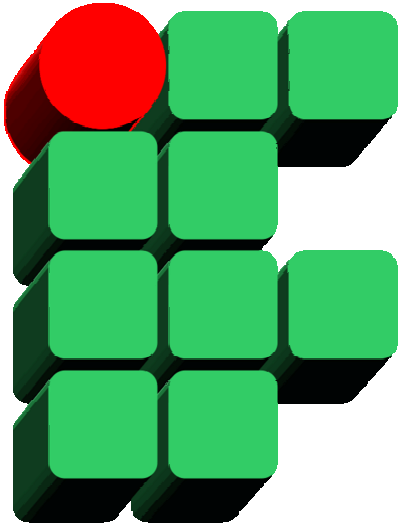
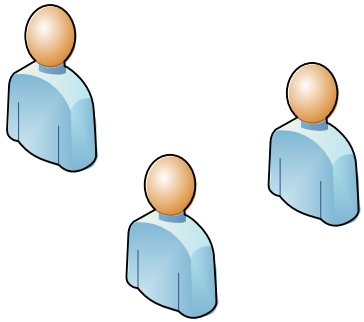


Formação de DBAs SQL Server 2008

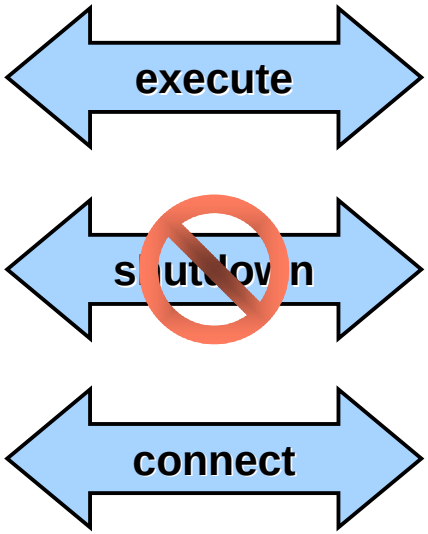
Parte 3: Gerenciamento de Segurança

Material Cedido pelo Professor Leandro Coelho Correia

Arquitetura de Segurança do SQL Server 2008



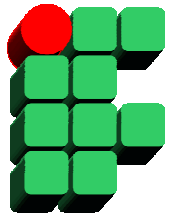
PRINCIPALS
Entidades que solicitam acesso a recursos do SQL Server 2008.



PERMISSIONS
Controles de segurança entre principals e securables.

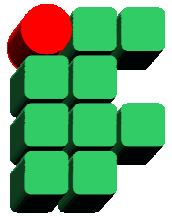
SECURABLES
Objetos que podem ser gerenciados (criados, alterados, excluídos, configurados, etc).

Principals



- O SQL Server 2008 reconhece as seguintes entidades como principals:
  - Login
    - Controla o acesso a uma instância do SQL Server 2008;
    - Utilizado para realizar a autenticação de uma entidade (pessoa, grupo de pessoas, empresa, etc) frente ao SQL Server 2008;
    - Podem ser usadas as credenciais de um usuário do sistema operacional ou do SQL Server a depender do modo de autenticação configurado.
  - User
    - Controla o acesso aos bancos de dados de uma instância;
    - Para obter acesso aos objetos de um banco de dados o usuário deve possuir um login válido na instância desejada e um user no banco de dados em questão.
  - Role:
    - Agrupamento de logins, users ou roles;
    - Usada para facilitar a administração de segurança.

Criando Logins



- Criação de logins para autenticação via SQL Server 2008:

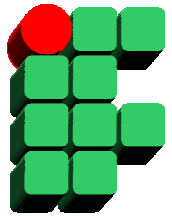
  CREATE LOGIN login_name **WITH PASSWORD =** 'password'
  [HASHED] [MUSTCHANGE]

  – HASHED: Indica que a senha informada já está criptografada;
  – MUST_CHANGE: Força o usuário a trocar a senha no próximo logon. As opções CHECK_EXPIRATION e CHECK_POLICY precisam estar habilitadas.

- Criação de logins para autenticação via Windows:

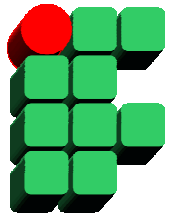
  CREATE LOGIN [domain\login_name] **FROM WINDOWS**

Criando Logins



- Opções de criação dos logins (cláusula WITH):
  - **SID =** sid
    - Especifica com qual ID o login será criado. Caso esse parâmetro seja omitido, o SQL Server 2008 atribue um ID automaticamente.
  - **DEFAULT_DATABASE =** database_name
    - Determina o banco de dados padrão ao efetuar logon.
  - **DEFAULT_LANGUAGE =** language
    - Especifica a linguagem padrão para o login criado.
  - **CHECK_EXPIRATION =** { ON | OFF}
    - Ativa ou desativa a política de expiração de senha para o login criado.
  - **CHECK_POLICY =** { ON | OFF}
    - Ativa ou desativa a política de senha d Windows para o login criado.

Criando Logins



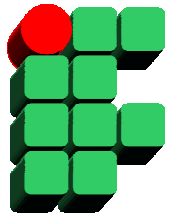
- Exemplos de criação de logins:

```
CREATE LOGIN leandro WITH PASSWORD = 'abCxyZ30'
MUST_CHANGE, CHECK_EXPIRATION = ON, CHECK_POLICY = ON;

CREATE LOGIN [LAB\leandro] FROM WINDOWS
WITH DEFAULT_DATABASE = bd_treinamento ;
```

Obs.: Logins podem ser alterados através do comando ALTER LOGIN e excluídos através do comando DROP LOGIN.

Criando Users



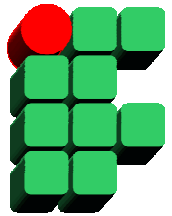
- Criação de user para um login existente:

  CREATE USER user_name **{ FOR | FROM } LOGIN** login_name;

- Criação de user sem login associado:

  CREATE USER user_name **WITHOUT LOGIN**;

Criando Users



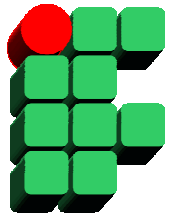
- Exemplos:

```
CREATE USER lcorreia FOR LOGIN leandro;

CREATE USER lcorreia WITHOUT LOGIN;
```

Obs.: Users podem ser alterados através do comando ALTER USER e excluídos através do comando DROP USER.

Roles

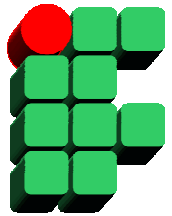


- No SQL Server 2008, existem permissões associadas aos logins que permitem a realização de operações no escopo de uma instância e permissões associadas aos users que permitem a realização de operações no escopo de um banco de dados;

- Para associar um conjunto de permissões de uma instância a um conjunto de logins, são utilizadas fixed server roles;

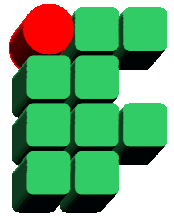
- Para associar um conjunto de permissões de um banco de dados a um conjunto de users, são utilizadas database roles.

Roles



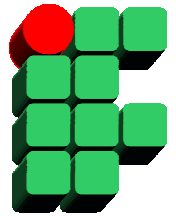
- Fixed Server Roles
  - Controlam permissões sobre operações que interferem no funcionamento de uma instância do SGBD (create database, shutdown, control server, etc);
  - Podem ser associadas a logins;
  - Novas server roles não podem ser criadas e as pré-definidas não podem ser alteradas.
- Database Roles
  - Controlam permissões sobre operações que manipulam objetos de um banco de dados (insert, update, delete, select, connect, control, etc);
  - Podem ser associadas a users;
  - Em cada banco de dados, existe um conjunto pré-definido de roles chamadas fixed database roles que facilitam o gerenciamento de permissões. Fixed database roles não podem ser alteradas;
  - É possível criar novas database roles.

Fixed Server Roles



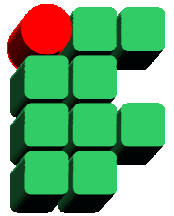
Roles	Permissões Associadas
Bulkadmin	Permite a execução de operações de importação de dados (bulk insert).
Dbcreator	Permite a criação de novos bancos de dados.
Diskadmin	Permite o gerenciamento de arquivos dos bancos de dados.
Processadmin	Permite o encerramento de processos (kill) que estejam executando no escopo do SGBD.
Securityadmin	Permite criar e configurar logins.

Fixed Server Roles



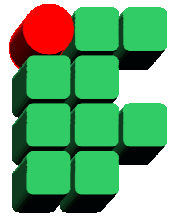
Roles	Permissões Associadas
Serveradmin	Permite modificar configurações da instância e realizar shutdown.
Setupadmin	Permite a criação e exclusão de linked servers.
Sysadmin	Permite a execução de qualquer operação.

Fixed Database Roles



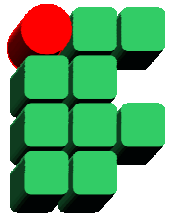
Roles	Permissões Associadas
db_accessadmin	Permite a criação e exclusão de users e schemas.
db_backupoperator	Permite a execução de backups os banco de dados.
db_datareader	Permite consultar (SELECT) todas as tabelas e visões de um banco de dados.
db_datawriter	Permite alterar (DELETE, UPDATE, INSERT) todas as tabelas e visões de um banco de dados.
db_ddladmin	Permite a execução de comandos DDL no banco de dados.

Fixed Database Roles



Roles	Permissões Associadas
db_denydatareader	Bloqueia consultas (SELECT) em todas as tabelas e visões de um banco de dados.
db_denydatawriter	Bloqueia alterações (DELETE, UPDATE, INSERT) em todas as tabelas e visões de um banco de dados.
db_owner	Permite a execução de qualquer comando de manutenção e configuração no banco de dados.
db_securityadmin	Permite gerenciar permissões e participação de users em roles.

Criando Database Roles



- Sintaxe:

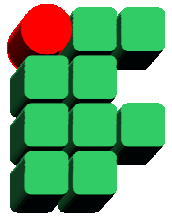
  CREATE ROLE role_name **[AUTHORIZATION** owner_name**]**

- Exemplo:

  CREATE ROLE alunos_treinamento_sql;

Obs.: Roles podem ser alteradas através do comando ALTER ROLE e excluídas através do comando DROP ROLE.

Manutenção de Usuários em Roles



- Adicionando usuários a uma role:
  - Database Role:

    sp_addrolemember database_role_name, user_name
  - Server Role:

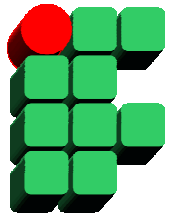
    sp_addsrvrolemember login, server_role_name

- Removendo usuários de uma role:
  - Database Role:

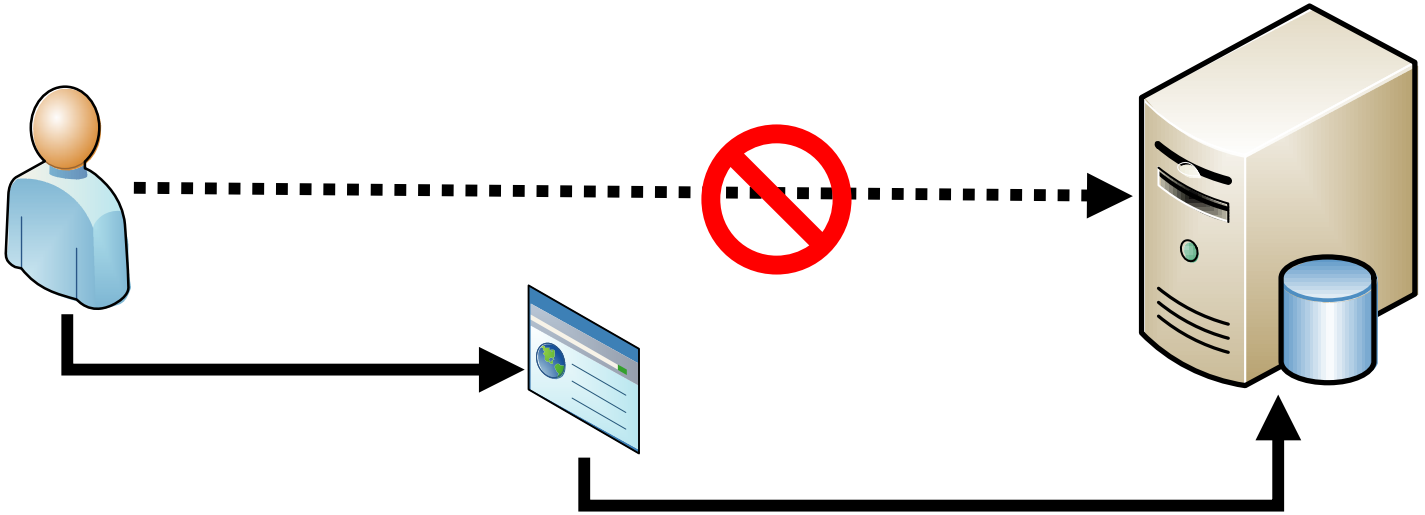
    sp_droprolemember database_role_name, user_name
  - Server Role:

    sp_dropsrvrolemember login, server_role_name

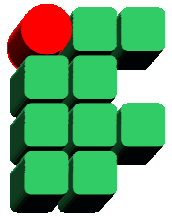
Application Roles



- Permitem que aplicações acessem o banco de dados sem a necessidade criação de users;
- Garante o acesso das aplicações ao banco de dados evitando o acesso direto dos usuários.



Criando Application Roles



- Sintaxe:

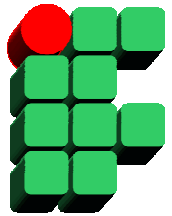
```
CREATE APPLICATION ROLE application_role_name
WITH PASSWORD = 'password'
[ , DEFAULT_SCHEMA = schema_name ]
```

- Exemplo:

```
CREATE APPLICATION ROLE excel_frequencia_treinamento
WITH PASSWORD = ‘abcXYZ'
```

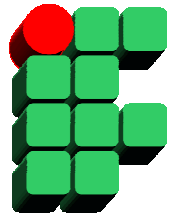
Obs.: Application Roles podem ser alteradas através do comando ALTER APPLICATION ROLE e excluídas através do comando DROP APPLICATION ROLE.

Securables



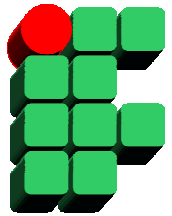
- Securables são objetos que têm seu acesso controlado pelo SQL Server 2008;
- Estão divididos em 3 categorias:
  - Server securables:
    - endpoints, logins, databases;
  - Database securables:
    - users, roles, application roles, assemblies, message types, schemas, etc;
  - Schema securables:
    - types, XML schema collections, object.

Securables



- Schemas
  - Coleções de objetos que formam um único espaço de nomes (namespace);
  - No SQL Server 2000 os conceitos de schema e user eram acoplados, o que dificultava a manutenção de users;
  - No SQL Server 2008, os conceitos de schema e user são desassociados. Isso permite que roles sejam proprietárias de schemas, simplificando a manutenção de users.

Criando Schemas



- Sintaxe:

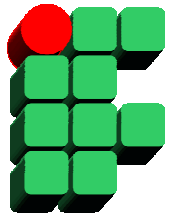
  CREATE SCHEMA schema_name **[AUTHORIZATION** owner_name**];**

- Exemplo:

  CREATE SCHEMA fabricasw;

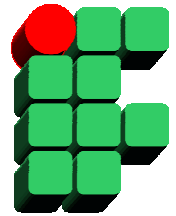
Obs.: Schemas podem ser alterados através do comando ALTER SCHEMA e excluídos através do comando DROP SCHEMA.

Permissions



- Controlam o acesso aos objetos do banco de dados e determinam que usuários podem executar determinadas operações;
- Estabelecem as relações entre principals e securables;
- Estão classificadas em duas categorias:
  - Server permissions:
    - Controlam o acesso dos DBAs a tarefas administrativas;
    - São estabelecidas através de fixed server roles ou através de comandos DCL.
  - Database permissions:
    - Controlam o acesso a objetos e a execução de comandos em um banco de dados;
    - Podem ser estabelecidas através de database roles ou diretamente através de comandos DCL (Database Control Language).

Concedendo Permissões



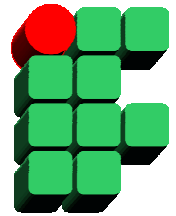
- Sintaxe:

  GRANT ALL | permission [,...n] ON securable **TO** principal **[,...n]**
  [WITH GRANT OPTION]

- Exemplos:

  GRANT ALL ON Employee TO Joe, Mary WITH GRANT OPTION;
  GRANT INSERT, DELETE ON Employee TO Paul;
  GRANT CREATE PROCEDURE, CREATE VIEW TO Mike;
  GRANT SHUTDOWN TO Jack;

Negando Permissões



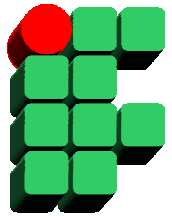
- Sintaxe:

  DENY ALL | permission [,...n] ON securable **TO** principal **[,...n]**
  [CASCADE]

- Exemplos:

  DENY SELECT ON Employee TO Mary CASCADE;
  DENY ALL ON Employee TO Paul;
  DENY CREATE VIEW TO Mike;
  DENY SHUTDOWN TO Joe;

Removendo Permissões



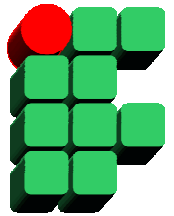
- Sintaxe:

  REVOKE GRANT OPTION FOR principal
  REVOKE ALL | permission [,...n] ON securable
  TO | **FROM** principal **[,...n] [CASCADE]**

- Exemplos:

  REVOKE GRANT OPTION FROM Mary;
  REVOKE ALL ON Employee FROM Joe CASCADE;
  REVOKE CREATE PROCEDURE FROM Mike;
  REVOKE SHUTDOWN FROM Jack;

Composição de Permissões

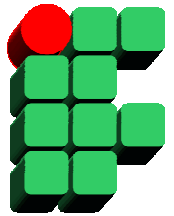


- O GRANT é propagado de escopo maior para o menor. Isso significa que se um user pertence a uma role e esta role possui uma determinada permissão, o user herda essa permissão;

- O DENY prevalece em quaquer escopo. Isso significa que se um user possuir um DENY para uma determinada permissão, mesmo que ele pertença a uma role que possui GRANT para a mesma permissão, o DENY prevalece;

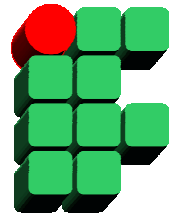
- O REVOKE é utilizado para remoção de GRANT ou DENY.

Composição de Permissões

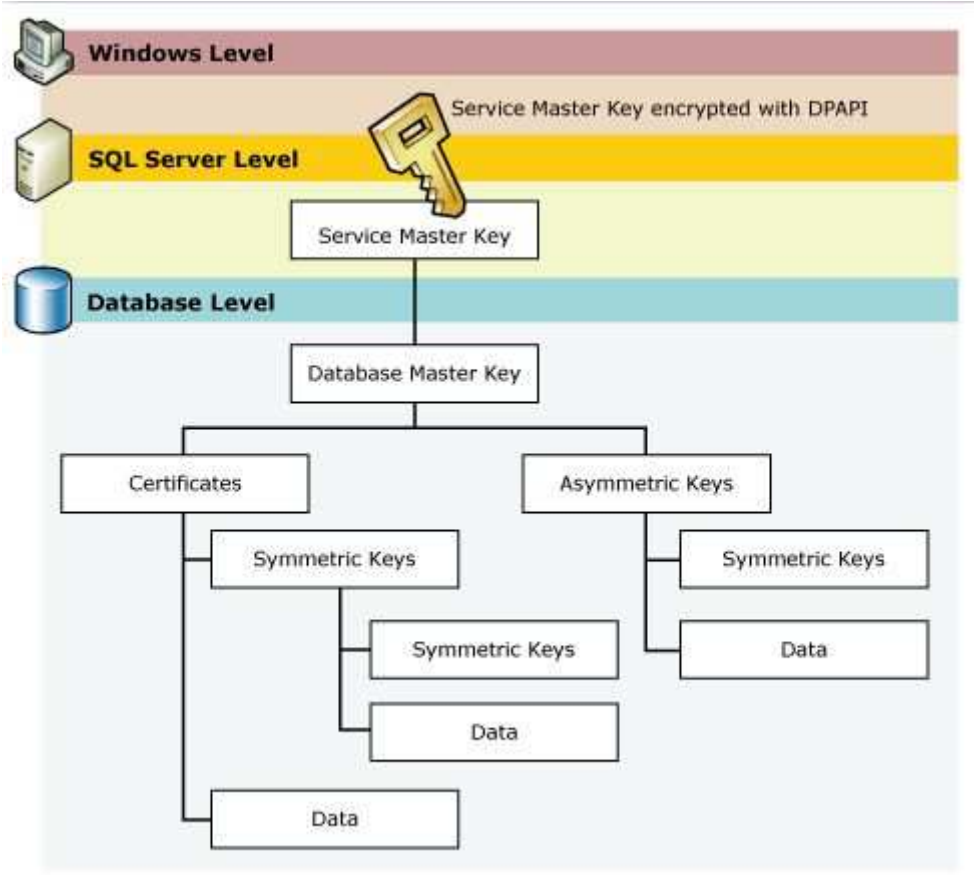


Permissão da Role	Permissão do Usuário	Situação do Usuário
grant select em uma tabela	nenhuma	Com permissão de leitura na tabela.
nenhuma	grant select em uma tabela	Com permissão de leitura na tabela.
nenhuma	nenhuma	Sem permissão de leitura na tabela.
deny select em uma tabela	nenhuma	Sem permissão de leitura na tabela.
nenhuma	deny select em uma tabela	Sem permissão de leitura na tabela.
grant select em uma tabela	deny select em uma tabela	Sem permissão de leitura na tabela.
deny select em uma tabela	grant select em uma tabela	Sem permissão de leitura na tabela.

Criptografia no SQL Server 2008

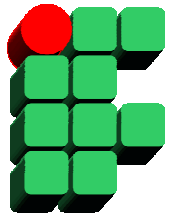


- Arquitetura de Criptografia do SQL Server 2008



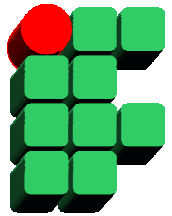
Fonte: Microsoft SQL Server 2008 Books Online

Criptografia no SQL Server 2008



- Service Master Key:
  - Criada automaticamente pelo SQL Server 2008 quando a primeira solicitação de criptografia é gerada;
  - Derivada das credenciais da conta que executa o serviço do SQL Server 2008 e cifrada usando a DPAPI (Windows Data Protection API);
  - Para restaurar um backup de um banco de dados cifrado é necessário conhecer a service master key da instância onde o backup foi realizado.

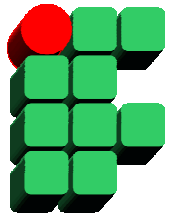
Criptografia no SQL Server 2008



- Database Master Key:
  - Utilizada para cifrar chaves assimétricas e certificados no banco de dados onde foi criada;
  - Cada banco de dados que utiliza criptografia precisa ter uma database master key criada;
  - A database master key é criada manualmente através do comando abaixo:

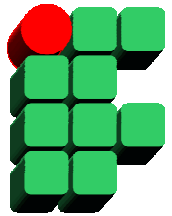
```
CREATE MASTER KEY
ENCRYPTION BY PASSWORD = 'abcXYZ'
```

Criptografia no SQL Server 2008



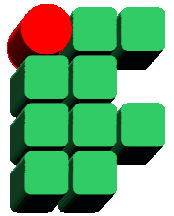
- Symmetric Keys (Chaves Simétricas):
  - Utilizam a mesma chave para cifrar e decifrar os dados;
  - Implementam um mecanismo de criptografia com menor custo de processamento, mas não é tão eficaz quanto as chaves assimétricas e os certificados.

Criptografia no SQL Server 2008

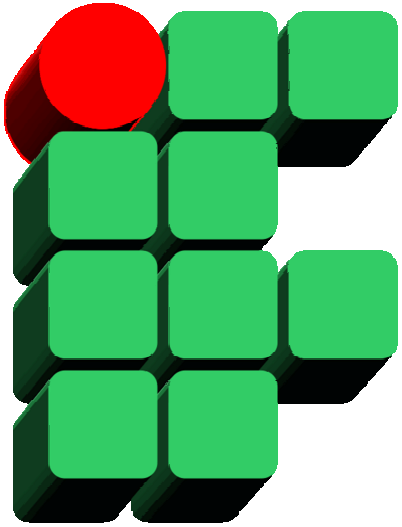


- Asymmetric Keys (Chaves Assimétricas):
  - Utilizam um modelo de chave pública e privada;
    - Os dados podem ser criptografados com a chave privada e todo mundo que possuir a chave pública poderá decriptografar a informação. (autenticidade do emissor);
    - Os dados podem ser criptografados com a chave pública e somente aquele que possuir a chave privada poderá decriptografar a informação(sigilo da informação).
  - Implementam um mecanismo de criptografia eficiente, mas com alto custo de processamento;
  - Podem ser usadas para cifrar dados ou chaves simétricas.

Criptografia no SQL Server 2008



- Certificates (Certificados):
  - Documento digital assinado com a chave pública de uma entidade (pessoa, empresa, etc);
  - Emitido por uma autoridade certificadora que garante a autenticidade da entidade que possui o certificado;
  - O SQL Server 2008 reconhece certificados no padrão X.509 e permite a criação de certificados próprios ou a importação de certificados previamente criados.



Formação de DBAs SQL Server 2008

Parte 3: Gerenciamento de Segurança