

# Администрирование, безопасность

## Лекция №11

Бутенко И.В. 2017 год

# Роль администратора

- Группа технического обслуживания – отвечает за работу аппаратного обеспечения
- Сетевой администратор
- Системный администратор – поддержание работы нескольких серверов
- WEB мастер
- Администратор БД

# Обязанности АБД

- Установка и модернизация SQL Server
  - Определение системных требований и пожеланий пользователей
  - Выбор платформы
  - Архитектура сервера
  - Инсталляция\Модернизация

# Обязанности АБД

- Наблюдение за состоянием сервера БД и его настройка
  - Запуск, приостановка и остановка сервера – Service Manager
  - Management Studio
  - SQL Mail
  - Profiler

# Обязанности АБД

- Правильное использование памяти
- Резервное копирование и восстановление данных
- Управление пользователями и обеспечение безопасности
- Сотрудничество с разработчиками
- Перенос данных
- Репликация данных
- Хранилище данных
- Составление графика обработки событий
- Обеспечение круглосуточного доступа к данным

# Безопасность ИС

- Организационные мероприятия
- Внедрение технических средств защиты
- Дополнительные средства защиты
- Разграничение прав на доступ к ресурсам СУБД

# Организационные мероприятия

- Определение ролей пользователей
- Выделение разных БД с разным уровнем конфиденциальности данных и защищенности доступа

# Технические средства защиты

- Ключи ЭЦП
- Токены
- Смарт-карты
- Криптокалькуляторы
- Биометрическая аутентификация
- Шифрование трафика



# Дополнительные средства

- СМС-авторизация
- СМС-информирование
- Google Authenticator
- FRAUD-Анализ

# Безопасность в SQL Server

Уровни в системе безопасности:

- аутентификация
- проверка разрешений

**Аутентификация** – процесс при котором пользователь в зависимости от результата допускается или не допускается к установлению соединения с SQL Server.

# Типы аутентификации

Windows – используются преимущества системы безопасности Windows и ее механизма учетных записей. Этот режим позволяет использовать имя пользователя и пароль, которые определены в Win и тем самым обходить процесс подключения к SQL Server.

Преимущества: пользователю не нужно запоминать еще один пароль и имя пользователя; в случае изменения пароля, его не нужно менять в SQL Server

# Типы аутентификации

Смешанный режим – в этом режиме задействованы обе системы аутентификации: Windows и SQL Server, необходимо ввести имя пользователя и пароль для сервера.

# Logins

Имена пользователей для входа – **logins**

Login дает право на подключение

- Хранится в БД Master
- Относится к серверу в целом
- Сам по себе не дает прав

*Добавление учетных записей для входа: **sp\_addlogin**  
@loginname, @passwd*

*Удаление учетных записей: **sp\_droplogin** @loginname*

*Изменение пароля: **sp\_password** @old, @new,  
@loginname*

# Logins

```
CREATE LOGIN login_name { WITH <option_list1> |  
  FROM <sources> }
```

```
<sources> ::= WINDOWS [ WITH <windows_options> [ ,...  
  ] ] | CERTIFICATE certname | ASYMMETRIC KEY  
  asym_key_name
```

```
<option_list1> ::= PASSWORD = 'password' [ HASHED ] [  
  MUST_CHANGE ] [ , <option_list2> [ ,... ] ]
```

```
<option_list2> ::= SID = sid | DEFAULT_DATABASE =  
  database | DEFAULT_LANGUAGE = language |  
  CHECK_EXPIRATION = { ON | OFF } | CHECK_POLICY  
  = { ON | OFF } [ CREDENTIAL = credential_name ]
```

```
<windows_options> ::= DEFAULT_DATABASE =  
  database | DEFAULT_LANGUAGE = language
```

# Схема

Схема — это коллекция сущностей базы данных, формирующая единое пространство имен.

Пространство имен — это набор, в котором у каждого элемента есть свое уникальное имя.

```
CREATE SCHEMA schema_name_clause [  
  <schema_element> [ , ...n ] ]
```

```
<schema_name_clause> ::= { schema_name |  
  AUTHORIZATION owner_name | schema_name  
  AUTHORIZATION owner_name }
```

```
<schema_element> ::= { table_definition | view_definition |  
  grant_statement revoke_statement | deny_statement }
```

# Users

Пользователь (user) БД ассоциируется с правами

- С ним ассоциируется схема (коллекция объектов БД)
- Права назначаются пользователям БД, не login'ам
- Работает в рамках конкретной БД

Специальные пользователи:

- dbo – database owner
- Guest – гость



# Users

*Создание пользователей БД:*

- `CREATE USER user_name [ { FOR | FROM } { LOGIN login_name | CERTIFICATE cert_name | ASYMMETRIC KEY asym_key_name } | WITHOUT LOGIN ] [ WITH DEFAULT_SCHEMA = schema_name ]`
- `sp_adduser @loginname, @name_id_db, @grpname`

*Удаление:*

- `DROP USER user_name`
- `sp_dropuser, sp_revokedbaccess`

*Изменение:*

- `ALTER USER user_name WITH <set_item> [ ,...n ]`
- `<set_item> ::= NAME = new_user_name | DEFAULT_SCHEMA = schema_name`

# Roles

С помощью ролей можно логически сгруппировать пользователей, имеющих соответствующие права доступа.

- Роли уровня сервера (sysadmin, dbcreator, diskadmin )
- Роли уровня базы данных (db\_owner, db\_datareader, db\_datawriter)

```
CREATE ROLE role_name [ AUTHORIZATION owner_name ]
```

*Пользовательские роли (sp\_addrole, sp\_addrolemember)*

*Пользовательские роли серверного уровня: (sp\_addsrvrole, sp\_addsrvrolemember)*

Специальная роль public

# Права доступа

- Права доступа (permission) – это выдаваемое конкретному пользователю разрешение сделать, что либо в базе данных. Существует два типа прав доступа: объектное (object) и командное (statement).
- Объектные права доступа определяют, кто может получать доступ и работать с данными в объектах БД.
- Командные права доступа определяют, кто может удалять и создавать объекты БД.

# Права доступа

- GRANT (Выдать) Назначение прав доступа (SELECT, UPDATE, INSERT, DELETE, EXECUTE) и назначение прав на выполнение команд (CREATE TABLE)
- REVOKE (Аннулировать) Аннулирование прав на доступ к данным и выполнение команд
- DENY (Отказать) Отказ в правах
  
- GRANT SELECT ON OBJECT::Person.rates TO bigor; GO