* + - 1. Показать и объяснить, какой режим аутентификации используется для экземпляра SQL Server.

Сущ-ет 2 режима аутентификации:

* + - * + Виндовс (для доменных БД – внутри организации)
        + Смешанная аутентификация скл сервер (не исп.домены, доступ сторонних юзеров)
      1. Создать необходимые учетные записи, роли и пользователей. Объяснить назначение привилегий.

Есть 2 типа учетных записей логины и пользователи:

* + - * + Логин – понятие всего sql. У них мб права на какие-то административные задачи (бэкап). Чтобы у логина были права на БД на основе него создается юзер
        + Пользователь – понятие в рамках одной бд. Им нельзя дать права по админ.части, но можно дать права на работу с бд.

ЛОГИН:

--у этого логина уже будет право на вход.

-- проверить: Безопасность => имена для входа => Julia\_C

СОЗДАНИЕ ПОЛЬЗОВАТЕЛЯ ПРИВЯЗАННОГО К ЛОГИНУ:

-- проверить: БД => Sklad => безопасность => пользователи => Julia\_User

-- пользователь дб в той базе, откуда выполняем запрос

-- здесь же пользователь JustUser

Пользователи создаются без каких=либо прав на базу.

Привилегии – те действия, которые может выполнять пользователь. Т.е.они нужны для того, чтобы разрешить либо отменить какие-либо действия пользователя.

* + - 1. Продемонстрируйте заимствование прав для любой процедуры в базе данных.

Имперсонификация – разрешение заимствовать права у другого пользователя для выполнения определенного блока кода.

SETUSER остается в силе, пока не будет выполнен другой оператор SETUSER или пока текущая база данных не будет изменена с помощью оператора USE.

* + - 1. Создать для экземпляра SQL Server объект аудита.

Аудит – отслеживание и протоколирование событий, происходящих на сервере.

Аудит SQL Server:

\*аудит сервера для событий на уровне сервера

\*аудит бд для событий на уровне бд

Журнал аудита:

\*файл файловой системы

\*журнал windows application log

\*windows security log

СОЗДАНИЕ АУДИТА CAusit:

* max\_rollover\_files – макс количество файлов кроме текущего
* reserve\_disk\_space – заранее размещает на диске файл в соответствии с maxsize
* queue\_delay – задержка очереди сброса на диск (0 – синхронно, 1000 – 1 сек)
* on\_failure – действия в случае невозможности продолжения аудита (continue, fail operation, shutdown)
* MAX\_FILES - максимальное число файлов аудита, которые могут быть созданы. При достижении предела переключение на первый файл не производится. При достижении предела MAX\_FILES любое действие, которое вызывает создание дополнительных событий аудита, завершится ошибкой

--проверить: безопасность => аудит (через св-ва посмотреть где созданы можно)

* + - 1. Задать для серверного аудита необходимые спецификации.

Типы действия аудита:

|  |  |
| --- | --- |
| database\_change\_group | Изменение баз данных |
| database\_object\_change\_group | Изменения над объектами базы данных |
| login\_change\_password\_group | Изменения паролей |
| server\_state\_change\_group | Модификация служб |
| successful\_login\_group | Успешный вход |
| failed\_login\_group | Неудачная попытка входа |
| logout\_group | Выход пользователя из SQL Server |

* + - 1. Запустить серверный аудит, продемонстрировать журнал аудита.

--запуск: безопасность => аудит => правой конпкой мыши (включить аудит)

--журнал аудита: безопасность => аудит => правой конпкой мыши (посмотреть журналы)

* + - 1. Создать необходимые объекты аудита БД + создать спецификацию + запустить аудит + журнал

--сделать: БД lab10 => безопасность => спецификации аудита бд => создать

(insert, object, выбираем имя объекта (таблица), выбираем имя участника (dbo))

10. Остановить аудит БД и сервера.

Все откючить

11. Создать для экземпляра SQL Server ассиметричный ключ шифрования.

Шифрование:

\* Service Master Key – шифрует мастер-ключи БД

\* Database Master Key – шифрует симметричные ключи

\* Симметричные ключи – шифруют данные

Асимм.ключи – пара ключей public и private;

\* алгоритма rsa

\* create, alter, drop

--проверить что ключ создан: БД Sklad => безопасность => ассиметричные ключи

13. Создать для экземпляра SQL Server сертификат.

Сертификат – подписанная цифр. подписью инструкция, кот. связывает значение откр. ключа с id юзера, у-ва или службы, кот. имеет соотв. закрытый ключ.

Содержит:

* Откр. ключ субъекта
* Идентификационные данные субъекта + поставщика (имя и адрес эл.почты)
* Срок действия (valid from / valid to)
* Цифр. подпись поставщика

SQL м.генрировать автоподписанные сертификаты

\*create, alter, drop

--проверить: БД Sklad => безопасность => сертификаты

15. Создать для экземпляра SQL Server симметричный ключ шифрования данных.

1. Создаем симм. ключ SKey для шифрования симм. ключа  
2. Открыть его

3. Создаем симм. ключ SData для шифрования данных

4. Открыть его

Create, alter, drop, open, close, close all symmetric keys;

17. Продемонстрировать прозрачное шифрование базы данных.

Tde – transparent data encryption – прозрачное шифрование д-х

исп. для шифрования данных и файлов журналов перед их записью на диск и расшифр. перед их возвратом в приложение.

Это вып. на уровне [SQL](https://ru.wikipedia.org/wiki/SQL). В последующих резервных копиях файлов БД данные будут зашифрованы.

Чтобы выполнить прозрачное шифрование необходимо:

* + - * + Создать главный ключ
        + Создать или получить сертификат, защищенный главным ключом
        + Создать ключ шифрования бд и защитить его с помощью сертификата
        + Задайте ведение шифрования бд Sklad

=> обнуляется оставшаяся часть вирт. журнала транзакций и принудительно начинается новый

Это гарантирует, что после включения шифрования бд в журналах транзакций не останется простого текста.

19. Продемонстрировать применение ЭЦП при помощи сертификата.

SignByCert (id сертификата, текст, пароль)

VerifySignedByCert (certificate\_ID, plaintext, signature)

SignByAsymKey (asym\_key\_ID, plaintext, password)

VerifySignedByAsymKey (asym\_key\_ID, plaintext, signature)