**25. Администрирование и безопасность SQL сервера. Роли, права и привилегии**

**В системе SQL-сервер организована двухуровневая настройка ограничения доступа к данным**. На первом уровне необходимо создать так называемую учетную запись пользователя (login), что позволяет ему подключиться к самому серверу, но не дает автоматического доступа к базам данных. На втором уровне для каждой базы данных SQL-сервера на основании учетной записи необходимо создать запись пользователя. На основе прав, выданных пользователю как пользователю базы данных (user), его регистрационное имя (login) получает доступ к соответствующей базе данных. В разных базах данных login одного и того же пользователя может иметь одинаковые или разные имена user с разными правами доступа. Иначе говоря, с помощью учетной записи пользователя осуществляется подключение к SQL-серверу, после чего определяются его уровни доступа для каждой базы данных в отдельности.

**В системе SQL-сервер существуют дополнительные объекты – роли**, которые определяют уровень доступа к объектам SQL-сервера. Они разделены на две группы: назначаемые для учетных записей пользователя сервера и используемые для ограничения доступа к объектам базы данных.

**Итак, на уровне сервера система безопасности оперирует следующими понятиями:**

**аутентификация; учетная запись; встроенные роли сервера**.

На уровне базы данных применяются следующие понятия;

**пользователь базы данных; фиксированная роль базы данных; пользовательская роль базы данных.**

**Режимы аутентификации**

SQL Server предлагает два режима аутентификации пользователей:

режим аутентификации средствами Windows NT/2000;

смешанный режим аутентификации (Windows NT Authentication and SQL Server Authentication).

**Администрирование системы безопасности**

Для создания пользователя в среде MS SQL Server следует предпринять следующие шаги:

Создать в базе данных учетную запись пользователя, указав для него пароль и принятое по умолчанию имя базы данных (процедура sp\_addlogin).

Добавить этого пользователя во все необходимые базы данных (процедура sp\_adduser).

Предоставить ему в каждой базе данных соответствующие привилегии (команда GRANT) .

**Аутентифика́ция (англ. Authentication) — проверка принадлежности субъекту доступа предъявленного им идентификатора; подтверждение подлинности.**

**Роль – механизм, с помощью которого можно назначать права той или иной группе пользователей**

создание новой роли:

sp\_addrole

[@rolename=] 'имя\_роли'

[, [@ownername=] 'имя\_владельца']

добавление пользователя к роли:

sp\_addrolemember

[@rolename=] 'имя\_роли',

[@membername=] 'имя\_пользователя'

удаление пользователя из роли:

sp\_droprolemember

[@rolename=] 'имя\_роли',

[@membername=] 'имя\_пользователя'

удаление роли:

sp\_droprole

[@rolename=] 'имя\_роли'

**Администрирование системы безопасности**

**Для создания пользователя в среде MS SQL Server следует предпринять следующие шаги:**

**Создать в базе данных учетную запись** пользователя, указав для него пароль и принятое по умолчанию имя базы данных (процедура sp\_addlogin).

**Добавить этого пользователя** во все необходимые базы данных (процедура sp\_adduser).

**Предоставить ему в каждой** базе данных соответствующие привилегии (команда GRANT) .

Создание новой учетной записи может быть произведено с помощью системной хранимой процедуры:

sp\_addlogin

[@login=] 'учетная\_запись'

[, [@password=] 'пароль']

[, [@defdb=] 'база\_данных\_по\_умолчанию']

После завершения аутентификации и получения идентификатора учетной записи (login ID) пользователь считается зарегистрированным, и ему предоставляется доступ к серверу. Для каждой базы данных, к объектам которой он намерен получить доступ, учетная запись пользователя (login) ассоциируется с пользователем (user) конкретной базы данных, что осуществляется посредством процедуры:

sp\_adduser

[@loginame=] 'учетная\_запись'

[, [@name\_in\_db=] 'имя\_пользователя']

[, [@grpname=] 'имя\_роли']

Отобразить учетную запись Windows NT в имя пользователя позволяет хранимая процедура:

sp\_grantdbaccess

[@login=] ‘учетная\_запись’

[, [@name\_in\_db=]‘имя\_пользователя’]

Привилегиями, или правами, называются действия, которые пользователь имеет право выполнять в отношении данной таблицы базы данных или представления. В стандарте SQL определяется следующий набор привилегий:

SELECT – право выбирать данные из таблицы;

INSERT – право вставлять в таблицу новые строки;

UPDATE – право изменять данные в таблице;

DELETE – право удалять строки из таблицы;

REFERENCES – право ссылаться на столбцы указанной таблицы в описаниях требований поддержки целостности данных;

USAGE – право использовать домены, проверки и наборы символов.

**Предоставление привилегий пользователям**

Оператор GRANT применяется для предоставления привилегий в отношении поименованных объектов базы данных указанным пользователям. Обычно его использует владелец таблицы с целью предоставления доступа к ней другим пользователям. **Оператор GRANT имеет следующий формат:**

<предоставление\_привилегий>::=

GRANT {<привилегия>[,...n] |

ALL PRIVILEGES}

ON имя\_объекта

TO {<идентификатор\_пользователя>

[,...n]| PUBLIC}

[ WITH GRANT OPTION]

**Параметр <привилегия> представляет собой:**

<привилегия>::=

{SELECT | DELETE | INSERT

[(имя\_столбца[,...n])]

| UPDATE [(имя\_столбца[,...n])]}

| REFERENCES [(имя\_столбца[,...n])] |

В языке SQL для отмены привилегий, предоставленных пользователям посредством оператора GRANT, используется оператор REVOKE. С помощью этого оператора могут быть отменены все или некоторые из привилегий, полученных указанным пользователем раньше. Оператор REVOKE имеет следующий формат:

<отмена\_привилегий>::=

REVOKE[GRANT OPTION FOR]

{<привилегия>[,...n]

| ALL PRIVILEGES}

ON имя\_объекта

FROM {<идентификатор\_пользователя>

[,...n]| PUBLIC}

[RESTRICT | CASCADE]