Тема лекції 9:

Керування правами доступу

- Користувачі бази даних
- Створення користувачів
- Надання прав доступу
- Ролі і групи
- Повноваження надавати права
- Відміна прав доступу

Категорії користувачів бази даних

- адміністратор БД
- власник об’єктів БД
- користувач, який має право надавати повноваження
- користувач, який не має права надавати повноваження
- рядові користувачі

Адміністратор бази даних

- При інсталяції СУБД необхідно ввести реєстраційне ім’я (user) або обліковий запис користувача (login) та пароль (password). Таким чином ви автоматично стаєте адміністратором бази даних, тобто користувачем з повним об’ємом повноважень.
- Якщо адміністратору необхідно виконати деяку роботу, для якої не потрібні повноваження, то йому краще увійти у систему під іменем користувача з мінімальними повноваженнями, які дозволяють вирішити цю задачу.

Повноваження адміністратора бази даних

- має усі права на будь-які дії з базою даних
- несе велику відповідальність за порушення правил роботи з базою даних та за зіпсуті дані
- створює інших користувачів бази даних, визначаючи для них імена і права (може створити ще одного адміністратора)
- має повноваження надати та анулювати права доступу для інших користувачів

Власник об’єкта БД

- Будь-який користувач, який створив деякий об’єкт бази даних (таблицю чи віртуальну таблицю), стає власником (owner) цього об’єкта
- Це користувач БД з повноваженнями, який може призначити іншого власника цього ж об’єкта
- Власник таблиці володіє усіма повноваженнями відносно цієї таблиці, включаючи керування доступом до неї
- Власник віртуальної таблиці може і не бути власником базових таблиць (створивши віртуальну таблицю, можна захистити базові таблиці, власниками яких є інші користувачі)

Інші користувачі (PUBLIC)

- Користувачі, крім адміністраторів та власників, називаються публікою (public):
  - користувач, який має право надавати повноваження
  - користувач, який не має права надавати повноваження
  - рядові користувачі
- Якщо уповноважений користувач надає права доступу типу PUBLIC, то їх отримують усі користувачі бази даних

Створення і видалення користувачів

- CREATE USER <ім’я користувача> [WITH
  [SYSID <ідентифікатор користувача>]
  [PASSWORD ‘<пароль>’]]
  [CREATEDB | NOCREATEDB]
  [CREATEUSER | NOCREATEUSER]
  [IN GROUP <ім’я групи> [,…]]
  [VALID UNTIL ‘<час>’];

- DROP USER <ім’я користувача>;

Рекурсія надання прав доступу

- Звичайний користувач не має прав дoти, поки вони йому не нададуться спеціально тим користувачем, у якого вже є ці права і який має повноваження надавати права іншим користувачам.
  - Спочатку адміністратор створює користувачів і надає їм деякі права.
  - Якщо хтось із створених користувачів має повноваження передавати права, то створивши таблицю чи віртуальну таблицю, він може передати права на неї іншим користувачам.
  - І так далі.

Інструкція надання прав доступу

GRANT <список прав> ON <об’єкт>
TO <список користувачів>
[WITH GRANT OPTION]

Інструкція надання прав доступу, <список прав>:

- Права у списку прав інструкції GRANT розділяються комами.
- Якщо необхідно надати усі права, то вказують ключові слова

  ALL PRIVILEGES (усі повноваження).

Інструкція надання прав доступу, <список прав>:

В SQL:2003 права можуть приймати наступні значення:

- SELECT – право перегляду;
- DELETE – право видалення записів;
- INSERT[(<список стовпців>)] – право додавання нових записів з вставкою значень для вказаних стовпців;
- UPDATE[(<список стовпців >)] – право зміни значень вказаних стовпців; якщо імена стовпців не вказані, то маються на увазі усі стовпці;
- REFERENCES[(<список стовпців >)] – право доступу до таблиці, на які посилається дана таблиця;
- USAGE – право на домени, набори символів, співставлення і трансляції;
- UNDER – право на структуровані типи даних;
- TRIGGER – право на використання тригерів;
- EXECUTE – право на виконання зовнішньої програми.

Інструкція надання прав доступу, <об’єкт>:

Може приймати наступні значення:

- [TABLE] <ім’я таблиці>;
- DOMAIN <ім’я домену>;
- COLLATION <ім’я співставлення>;
- CHARACTER SET <ім’я символьного набору>;
- TRANSLATION <ім’я трансляції>;
- TYPE <користувацький тип>.

Інструкція надання прав доступу, <список користувачів>

- Складається з імен користувачів, розділеними комами
- Замість списку можна вказати ключове слово PUBLIC (публіка). У цьому випадку права, вказані в інструкції GRANT, отримують усі користувачі БД.

Приклад 1. Надання права перегляду

- Надати право перегляду таблиці Table_1 усім користувачам БД :

```
GRANT SELECT ON Table_1
  TO PUBLIC;
```

Приклад 2. Надання права змінювати значення стовпця

- Надати право змінювати значення стовпця Col1 таблиці Table_1 користувачеві SalesManager:

```
GRANT UPDATE ON Table_1(Col1)
  TO SalesManager;
```

- Якщо менеджеру потрібно змінювати значення декількох стовпців, то в інструкції GRANT необхідно перелічити їх імена через кому.

```
GRANT UPDATE ON Table_1(Col1,…,Col_n)
  TO SalesManager;
```

- Якщо потрібно дозволити змінювати усі стовпці таблиці, то використовується інструкція:

```
GRANT UPDATE ON Table_1
  TO SalesManager;
```

Приклад 3. Надання права вставляти записи та ін.

- Надати право вставляти записи і змінювати значення стовпців таблиці Table_1 користувачеві SalesManager:

```
GRANT UPDATE, INSERT ON Table_1
  TO SalesManager;
```

Приклад 4. Право на видалення

- Надати право на видалення рядків таблиць Table_1 і Table_2 користувачеві SuperManager:

```
GRANT DELETE ON Table_1, Table_2
    TO SuperManager;
```

Ролі і групи

- Поняття ролі (role) використовується в SQL:2003
- Поняття ролі (role) немає в стандарті SQL2, тому може і не бути в деяких СУБД.
- Деякі СУБД використовують поняття групи користувачів, яке є аналогічне поняттю ролі.
- При використанні ролі (групи) права призначаються для ролі (групи), а потім для кожного користувача вказується, яку роль він грає (або в яку групу входить).
- Роль (група) характеризується своїм ім’ям.

Ролі і групи. Інструкції.

- Створення ролі :
  CREATE ROLE <ім’я ролі>;
- Після створення роль призначається користувачам:
  GRANT <ім’я ролі> TO <список користувачів>;
- Призначення прав для ролі :
  GRANT <список прав> ON <об’єкт>
  TO <ім’я ролі>
  [WITH GRANT OPTION]

- Створення групи (якщо її підтримує реалізація SQL):
  CREATE GROUP <ім’я групи>
  WITH <список користувачів>;

Повноваження надавати права

- На практиці права доступу може надавати:
  - адміністратор бази даних,
  - власник об’єктів бази даних. (При цьому власники надають права лише на об’єкти, якими володіють)
- Треті особи, які отримали права доступу від адміністратора і/або власника, вже не можуть надавати права. Таке обмеження дозволяє адміністратору і власникам зберегти контроль на базою даних.
- Однак, бувають ситуації, коли необхідно делегувати повноваження надавати свої права іншим користувачам (наприклад, помічникам, або заміні під час відпустки).
- У цьому випадку в інструкції GRANT використовується фраза WITH GRANT OPTION.

Приклад 5. Повноваження надавати права

- Право оновлювати і додавати записи у таблиці Table_1 надається користувачу SalesManager. Причому йому надається право передавати отримані права доступу іншому користувачу:

```
GRANT UPDATE, INSERT ON Table_1
  TO SalesManager
  WITH GRANT OPTION;
```

- Після цього користувач SalesManager може передати свої права помічнику з іменем AssistantManager:

```
GRANT UPDATE, INSERT ON Table_1
  TO AssistantManager;
```

Відміна прав доступу

- Відміна вказаних у списку прав для перелічених користувачів:

REVOKE [GRANT OPTION FOR] <спис. прав>
ON <об’єкт>
FROM <список користувачів>
[RESTRICT | CASCADE];

- При цьому можна анулювати деякі або усі права.

Ключові фрази в інструкції REVOKE

- CASCADE (каскадно) – відміняються вказані права у перелічених користувачів, а також у тих, кому ці користувачі передали повноваження;
- RESTRICT (обмежити) – відміняються вказані права у користувачів, які нікому іншому їх не надавали. Однак, якщо користувач встиг надати права, вказані в цьому операторі REVOKE, то оператор не виконається і з’явиться повідомлення про помилку;
- GRANT OPTION FOR (право надавати для) – застосовується, щоб відмінити у користувача повноваження передавати вказані права, але залишити їх за користувачем.

Фрази в інструкції REVOKE

- Якщо оператор REVOKE містить GRANT OPTION FOR та CASCADE, то відміняють усі повноваження, надані користувачем, а також право цього користувача на надання повноважень.
- Якщо в операторі REVOKE включено GRANT OPTION FOR та RESTRICT, то є можливими наступні два варіанта.
  - якщо користувач ще не надав іншому користувачеві повноважень, які у нього відміняються, то оператор REVOKE виконається, відміняючи також право цього користувача надавати повноваження;
  - якщо користувач вже надав кому-небудь хоча б одне з зазначених в операторі REVOKE повноважень, то права не відміняються і повертається код помилки.

Інше використання інструкції REVOKE (оптимізація SQL-коду)

- Пов’язане не з прямою задачею відміни повноважень, а навпаки – надання повноважень.

- Як правило, надання прав багатьом користувачам на велику кількість об’єктів пов’язане з великим об’ємами SQL-коду.
- Комбінуючи оператори GRANT та REVOKE, надаючи спочатку широкі повноваження для багатьох користувачів, а потім обмежуючи їх для деяких користувачів, можна скоротити загальний об’єм SQL-коду.

Приклад 6. Інструкції керування доступом

- Користувачі User_1 і User_2 мають права переглядати, додавати і видаляти записи таблиці Table_1(Id_Col,Col1,Col2,Col3,Col4). Також вони можуть змінювати значення стовпців, крім стовпця Id_Col. Усі інші користувачі можуть лише переглядати записи.

  (Такий розподіл прав можна виконати двома способами)

Приклад 6. Рішення 1

- Надати відповідні права на усі операції. У випадку надання прав на зміну, треба перелічити усі стовпці, дозволені для цього.

```
GRANT SELECT ON Table_1 TO PUBLIC;
GRANT INSERT, DELETE ON Table_1
     TO User_1,User_2;
GRANT UPDATE ON
   Table_1(Col1,Col2,Col3,Col4)
   TO User_1,User_2;
```

Приклад 6. Рішення 2

- Спочатку надати право оновлювати усі стовпці, а потім відізвати право оновлювати заборонений стовпець :

```
GRANT SELECT ON Table_1 TO PUBLIC;
GRANT INSERT, UPDATE, DELETE
   ON Table_1 TO User_1,User_2;
REVOKE UPDATE ON Table_1(Id_Col)
      TO User_1,User_2;
```

Дякую за увагу

Опрацювати: Д.Петковіч «Microsoft SQL Server 2012. Руководство для начинающих» ***ст. 323-367***