Заняття 12. Керування правами доступу

ЛОГОС
Центр комп'ютерного навчання

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, перейменування і видалення користувачів

- CREATE USER user_specification [, user_specification] ...
- user_specification:

  user [IDENTIFIED BY [PASSWORD] 'password']

- RENAME USER old_user TO new_user

  [, old_user TO new_user] ...

- DROP USER user [, user] ...

Рекурсія надання прав доступу

- Звичайний користувач не має прав доти, 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

  priv_type [(column_list)]

  [, priv_type [(column_list)]] ...

  ON [object_type] priv_level

  TO user_specification [, user_specification] ...

  [REQUIRE {NONE | ssl_option [[AND] ssl_option] ...}]

  [WITH with_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).

- 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;
- USAGE – право на домени, набори символів, співставлення і трансляції;
- UNDER – право на структуровані типи даних;
- CREATE – право на створення об'єктів;
- EXECUTE – право на виконання зовнішньої програми.

Інструкція надання прав доступу. Об’єкт

- TABLE
- FUNCTION
- PROCEDURE
- USER
- VIEW

Інструкція надання прав доступу. Список користувачів

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

Приклад 1. Створення користувача та надання йому прав перегляду бази даних

- CREATE USER 'someuser'@'localhost' IDENTIFIED BY 'somepass';
- GRANT SELECT ON mydb.* TO 'someuser'@'localhost';

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.

Відміна прав доступу

- REVOKE

  priv_type [(column_list)]

  [, priv_type [(column_list)]] ...

  ON [object_type] priv_level

  FROM user [, user] ...

- REVOKE ALL PRIVILEGES, GRANT OPTION

  FROM user [, user] ...

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 2. 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 2. 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 ''@'localhost';

GRANT INSERT, DELETE ON Table_1 TO 'User_1'@'localhost', 'User_2'@'localhost';

GRANT UPDATE (Col1,Col2,Col3,Col4) ON Table_1 TO 'User_1'@'localhost', 'User_2'@'localhost';
```

Приклад 2. Рішення 2

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

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

Перегляд стану бази даних

- Для перегляду об'єктів бази даних, прав і т.д. існує команда SHOW.
- Основні типи команди SHOW:
  - SHOW CREATE DATABASE db_name
  - SHOW CREATE FUNCTION func_name
  - SHOW CREATE PROCEDURE proc_name
  - SHOW CREATE TABLE tbl_name
  - SHOW DATABASES [like_or_where]
  - SHOW GRANTS FOR user
  - SHOW INDEX FROM tbl_name [FROM db_name]
  - SHOW PRIVILEGES
  - SHOW TRIGGERS [FROM db_name] [like_or_where]

Завдання 1. Надання прав на використання власної бази даних

- Створити 3 користувачі для сервера, 1 з яких встановити пароль.
- 1 користувачу (із паролем) надати права адміністратора для власної бази даних.
- 1 користувачу надати права перегляду всіх таблиць бази даних та можливості додання та модифікації даних для 1 таблиці.
- 1 користувачу надати права перегляду усієї бази даних та можливість створення об'єктів в цій базі (збережених процедур, віртуальних таблиць і т. д.).
- Останнім 2-м користувачам заборонити перегляд 1 із таблиць бази даних.