

Комп'ютерний практикум №5

до теми «Мова керування даними (DCL)»

Теоретичні відомості

Команди DCL (Data Control Language) контролюють рівень доступу, користувача на об'єкти бази даних (БД).

GRANT – надає користувачам дозвіл на читання/запис об'єктів БД [1].

REVOKE – позбавляє користувачів дозволу читання/запису об'єктів БД [2].

Аутентифікація. «Хто Ви?»

MySQL аутентифікує користувача по:

- імені;
- паролю;
- комп'ютеру, з якого встановлюється з'єднання.

Знання особистості користувача – обов'язкова умова для призначення йому тих чи інших привілеїв.

Авторизація. «Що дозволено?»

В MySQL авторизація застосовується до глобальних привілеїв, не пов'язаним з якимись конкретними об'єктами схеми.

Контроль доступу. «Які дані дозволено бачити і/або змінювати?»

При спробі прочитати або модифікувати інформацію MySQL перевіряє, чи є у користувача дозвіл на перегляд або зміну запитаних стовпців. На відміну від глобальних привілеїв, правила контролю доступу застосовуються до конкретних даних, наприклад, до окремої БД, таблиці або колонки.

Привілеї та дозволи.

Привілей або дозвіл – це спосіб представлення в MySQL права доступу.

Рекомендований спосіб створювати облікові записи, а також додавати і видаляти привілеї – команди **GRANT** і **REVOKE**.

Щоб додати новий обліковий запис або привілей, використовуйте **GRANT**, команда **REVOKE** може видалити тільки привілей, але не обліковий запис. Для видалення облікового запису призначена команда **DROP USER**. Для перегляду привілеїв користувача існує команда **SHOW GRANTS** [3, стор.642].

Порядок виконання роботи

Відповідно до обраної теми курсового проекту, виконати:

1. Перевірити привілеї облікового запису під яким Ви увійшли:

```
SHOW GRANTS ;
```

2. Створити обліковий запис із ім'ям *someuser* та паролем *p4ssword* і надати йому усі привілеї:

```
GRANT ALL PRIVILEGES ON *.* TO 'someuser'@'localhost'  
IDENTIFIED BY 'p4ssword' WITH GRANT OPTION;
```

3. Відкликати усі привілеї облікового запису *someuser*:

```
REVOKE ALL PRIVILEGES, GRANT OPTION FROM  
'someuser'@'localhost' ;
```

4. Перевірити привілеї будь-якого облікового запису:

```
SHOW GRANTS FOR 'someuser'@'localhost' ;
```

5. Надати привілеї на перегляд стовпців *col1*, *col2* таблиці *tbl*, БД *db* обліковому запису *someuser*:

```
GRANT SELECT (col1, col2) ON db.tbl TO  
'someuser'@'localhost' ;
```

6. Надати привілеї на виконання збереженої процедури *proc*, БД *db* обліковому запису *someuser*:

```
GRANT EXECUTE ON PROCEDURE db.proc TO  
'someuser'@'localhost' ;
```

7. Перевірити надані привілеї користувача *someuser*. Спробувавши виконати процедури та вибрати дані із таблиць на які користувач має та не має привілеїв.

8. Видалити обліковий запис *someuser*:

```
DROP USER 'someuser'@'localhost' ;
```

Завдання на підвищення балів.

До теми «Мова керування транзакціями (TLC – Transaction Control Language)»

Транзакція. «Все або нічого»

Транзакцією називається атомарна група запитів SQL, які розглядаються як єдине ціле. Якщо підсистема бази даних може виконати всю групу запитів, вона робить це, але якщо будь-який із запитів не може бути виконаний в результаті збою або з якоїсь іншої причини, не буде виконаний жоден запит групи

Транзакцій недостатньо, якщо система не проходить тест **ACID**. Аббревіатура **ACID** розшифровується як **Atomicity**, **Consistency**, **Isolation** і **Durability** (атомарність, несуперечливість, ізолюваність і довговічність). Це тісно пов'язані критерії, яким повинна відповідати правильно функціонуюча система обробки транзакцій:

Atomicity – Атомарність

Транзакція має функціонувати як єдина неподільна одиниця роботи таким чином, щоб вся транзакція була або виконана, або скасована. Коли транзакції є атомарними, не існує такого поняття, як частково виконана транзакція: «**все або нічого**»

Consistency – Несуперечливість

БД повинна завжди переходити з одного несуперечливого стану в інший.

Isolation – Ізолюваність

Результати транзакції *зазвичай* невидимі іншим транзакціям, поки вона не закінчена. Рівні ізоляції: **READ UNCOMMITTED**, **READ COMMITTED**, **REPEATABLE READ**, **SERIALIZABLE**.

Durability – Довговічність

Зафіксовані, внесені в ході транзакції зміни стають постійними. Це означає, що зміни повинні бути записані так, щоб дані не могли бути втрачені в разі збою системи.

Транзакція починається командою **START TRANSACTION**, а потім або фіксується командою **COMMIT**, або скасовується командою **ROLLBACK** [4].

9. Створити транзакцію всередині якої змінити дані таблиці, зафіксувати у випадку, коли записів у таблиці не парна кількість і скасувати, якщо парна.
10. Створити процедуру із транзакцією всередині, яка фіксується або скасовується при виконанні/невиконанні деякої умови.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. MySQL 5.7 Reference Manual :: GRANT Syntax – Керівництво користувача MySQL 5.7. Синтаксис команди GRANT [Електронний ресурс] – Режим доступу: <https://dev.mysql.com/doc/refman/5.7/en/grant.html>
2. MySQL 5.7. Reference Manual :: REVOKE Syntax – Керівництво користувача MySQL 5.7. Синтаксис команди REVOKE [Електронний ресурс] – Режим доступу: <https://dev.mysql.com/doc/refman/5.7/en/revoke.html>
3. Шварц Б., Зайцев П., Ткаченко В., Заводны Дж., Ленц А., Бэллинг Д. MySQL. Оптимизация производительности, 2-е издание. – Пер. с англ. – СПб.: Символ-Плюс, 2010. – 832 с., ил. ISBN 978-5-93286-153-0.
4. MySQL 5.7. Reference Manual :: START TRANSACTION, COMMIT, and ROLLBACK Syntax – Керівництво користувача MySQL 5.7. Синтаксис START TRANSACTION, COMMIT, та ROLLBACK команд [Електронний ресурс] – Режим доступу: <https://dev.mysql.com/doc/refman/5.7/en/commit.html>