Міністерство освіти і науки України

Національний технічний університет України

«Київський політехнічний інститут імені Ігоря Сікорського»

Інститут прикладного системного аналізу

Кафедра математичних методів системного аналізу

**ЗВІТ**

про виконання комп’ютерного практикуму №4

з дисципліни «Системи баз даних»

Виконав:

Студент ІІІ курсу

Групи КА-96

Вітковський Д.О.

Перевірила:

Афанасьєва І.В.

Київ – 2021

**Тема:** Мова керування даними (DCL)

**Завдання:** Відповідно до обраної теми курсового проекту, виконати:

1. Перевірити привілеї облікового запису під яким Ви увійшли:

SHOW GRANTS;

1. 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;

1. Відкликати усі привілеї облікового запису someuser:

REVOKE ALL PRIVILEGES, GRANT OPTION FROM

'someuser'@'localhost';

1. Перевірити привілеї будь-якого облікового запису:

SHOW GRANTS FOR 'someuser'@'localhost';

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

GRANT SELECT (col1, col2) ON db.tbl TO

'someuser'@'localhost';

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

GRANT EXECUTE ON PROCEDURE db.proc TO

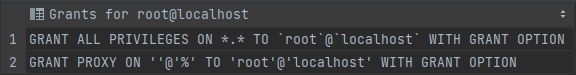
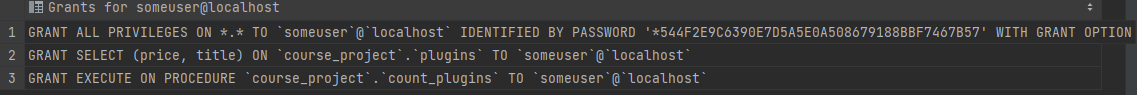
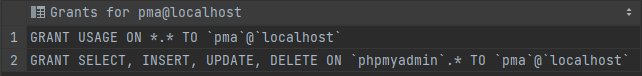
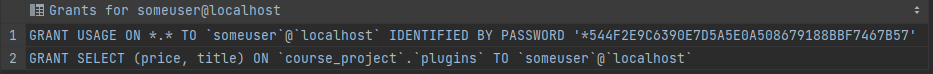
'someuser'@'localhost';

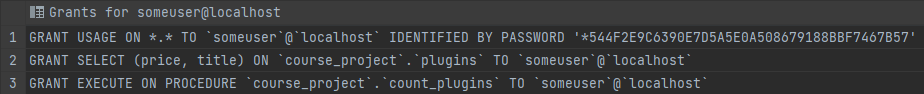
1. Перевірити надані привілеї користувача someuser. Спробувавши виконати процедури та вибрати дані із таблиць на які користувач має та не має привілеїв.
2. Видалити обліковий запис someuser:

DROP USER 'someuser'@'localhost';

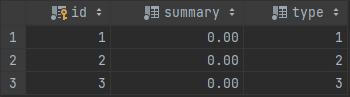
1. Створити транзакцію всередині якої змінити дані таблиці, зафіксувати у випадку, коли записів у таблиці не парна кількість і скасувати, якщо парна.
2. Створити процедуру із транзакцією всередині, яка фіксується або скасовується при виконанні/невиконанні деякої умови.

**Результати роботи**

-- 1  
show grants;  
  
-- 2  
grant all privileges on \*.\* to 'someuser'@'localhost'  
identified by 'p4ssword' with grant option;  
  
show grants FOR 'someuser'@'localhost';  
  
-- 3  
revoke all privileges, grant option from 'someuser'@'localhost';  
  
show grants FOR 'someuser'@'localhost';  
  
-- 4  
show grants FOR 'pma'@'localhost';  
  
-- 5  
grant select (title, price) on course\_project.plugins to 'someuser'@'localhost';  
  
show grants FOR 'someuser'@'localhost';  


-- 6  
create procedure *count\_plugins* () select count(1) from `course\_project`.`plugins`;  
  
grant execute on procedure course\_project.*count\_plugins* to 'someuser'@'localhost';  
  
show grants FOR 'someuser'@'localhost';  
  
-- 7  
show grants FOR 'someuser'@'localhost';  
  
select \* from course\_project.plugins; -- error on column `id`  
select title, price from course\_project.plugins; -- OK  
SELECT \* FROM course\_project.users; -- error on table `users`  
  
-- 8  
drop user 'someuser'@'localhost';  
  
-- 9  
create table if not exists `table1` (  
 id int not null auto\_increment,  
 salary decimal(10, 2) not null default 5000,  
 type tinyint not null,  
 constraint table1\_pk primary key (id)  
);  
insert into table1 (salary, type)  
values (1200.3, 1),  
 (1000, 1),  
 (3212, 1),  
 (2222.1, 2),  
 (1233.5, 2),  
 (3321, 3),  
 (900, 3),  
 (1500, 2);  
create table if not exists `table2`(  
 id int not null auto\_increment,  
 summary decimal(15, 2) not null default 0,  
 type tinyint not null,  
 constraint table2\_pk primary key (id)  
);  
insert into table2 (type)  
values (1),  
 (2),  
 (3);  
  
select \* from table1;



select \* from table2;  


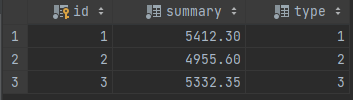
start transaction;  
select @is\_odd := if((select count(\*) from `table1`) % 2, 'COMMIT', 'ROLLBACK');  
select @A:=SUM(salary) from table1 where type=1;  
update table2 set summary=@A where type=1;  
select @A:=SUM(salary) from table1 where type=2;  
update table2 set summary=@A where type=2;  
select @A:=SUM(salary) from table1 where type=3;  
update table2 set summary=@A where type=3;  
prepare act from @is\_odd;  
execute act;



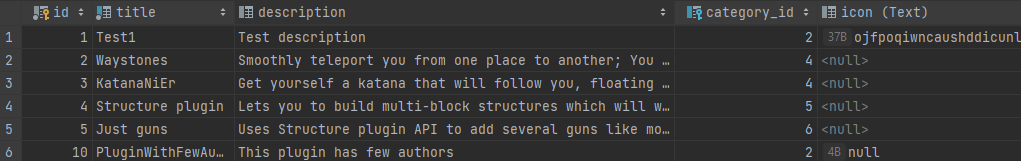
insert into table1 (salary, type) values (1111.35, 3);



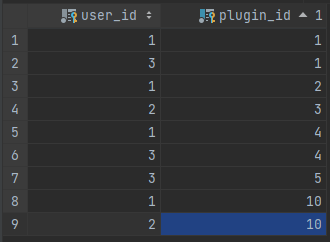
select \* from table2;



-- 10  
delimiter //  
create procedure *publish\_plugin* (in pl\_title varchar(200),  
 in pl\_description text,  
 in pl\_category smallint,  
 in pl\_icon mediumblob,  
 in pl\_version varchar(20),  
 in pl\_file longblob,  
 in pl\_authors json)  
begin  
 declare current\_pl\_id bigint;  
 declare indx bigint;  
 declare author\_id bigint;  
  
 start transaction;  
 savepoint sp;  
 insert into `course\_project`.`plugins` (title, description, category\_id, icon) values(pl\_title, pl\_description, pl\_category, pl\_icon);  
 select id into current\_pl\_id from `course\_project`.`plugins` where title = pl\_title;  
 insert into `course\_project`.`plugin\_versions` (plugin\_id, version\_title, `file`) values(current\_pl\_id, pl\_version, pl\_file);  
 set indx = 0;  
 repeat  
 set author\_id = json\_extract(pl\_authors, concat('$[', indx, ']'));  
 insert into `course\_project`.`plugin\_authors` values(author\_id, current\_pl\_id);  
 set indx = indx + 1;  
 until indx = json\_length(pl\_authors)  
 end repeat;  
 prepare act from if(json\_length(pl\_authors) > 3, 'ROLLBACK TO sp', 'COMMIT');  
 execute act;  
end//  
  
call *publish\_plugin*(  
 'PluginWithFewAuthors',  
 'This plugin has few authors',  
 2,  
 'null',  
 '1.0-SNAPSHOT',  
 'sdfuhasiufoiuefoquwbeouybasdiycbasdhc', -- contents of the file  
 '[1, 2]'  
);  
call *publish\_plugin*(  
 'PluginWithManyAuthors',  
 'This plugin has many authors',  
 2,  
 'null',  
 '1.0-SNAPSHOT',  
 'iufdihasdfihuqwjkhsduihasdasdjiofo', -- contents of the file  
 '[1, 2, 3, 4]'  
);







**Висновки**

Під час виконання комп’ютерного практикуму було створено виконано 10 запитів. Я набув навичок створення користувачів для бази даних, керування їхніми правами. Навчився користуватися транзакціями бази даних, зупиняти та відкочувати їх.