Julia Ruszer 247775

Dominik Gałkowski 247659

Jan Śladowski 247806

Wiktor Żelechowski 247833

Microsoft SQL Server

Zadanie 1.

1. Utwórz nowe loginy o nazwach login1, login2 i login3 z domyślną bazą danych HR.

```
create login login1 with password = 'login1', DEFAULT_DATABASE = HR;
create login login2 with password = 'login2', DEFAULT_DATABASE = HR;
create login login3 with password = 'login3', DEFAULT_DATABASE = HR;
```

4	login1	2025-01-09 14:47:00.720	HR
5	login2	2025-01-09 14:47:00.723	HR
6	login3	2025-01-09 14:47:00.723	HR

2. Utwórz nowych użytkowników w bazie danych HR o nazwach user1, user2 i user3 przypisanych do loginów login1, login2 i login3.

```
create user user1 for login login1;
create user user2 for login login2;
create user user3 for login login3;
```

6	user1	SQL_USER	2025-01-09 14:47:24.680
7	user2	SQL_USER	2025-01-09 14:47:24.683
8	user3	SQL_USER	2025-01-09 14:47:24.683

3. Nadaj użytkownikom user1, user2 i user3 uprawnienia do wyświetlania departamentów.

grant select on departments to user1, user2, user3;

1 2 3	UserName	UserName ObjectName F		PermissionState	
1	user1	departments	SELECT	GRANT	
2	user2	departments	SELECT	GRANT	
3	user3	departments	SELECT	GRANT	

4. Utwórz nową rolę o nazwie role1 i uczyń użytkownika user1 jej właścicielem.

create role role1 authorization user1;

	RoleName	RoleType	CreateDate	OwnerName
1	role1	DATABASE_ROLE	2025-01-09 14:56:19.273	user1

5. Nadaj użytkownikowi user1 uprawnienia do dodawania departamentów z możliwością dalszego przekazywania uprawnień.

grant insert on departments to user1 with grant option;

ш.,,	Es Messa	yes		
	GranteeName	ObjectName	PermissionName	PermissionState
1	user1	departments	INSERT	GRANT_WITH_GRANT_OPTION

6. Nadaj roli role1 uprawnienia do usuwania departamentów.

grant delete on departments to role1;

	GranteeName	ObjectName	PermissionName	PermissionState
1	role1	departments	DELETE	GRANT

7. Jako user1 nadaj użytkownikowi user2 uprawnienia do dodawania departamentów.

execute as user = 'user1';
grant insert on departments to user2;

	GranteeName	ObjectName	PermissionName	PermissionState
1	user2	departments	INSERT	GRANT

8. Jako user1 nadaj użytkownikowi user2 rolę role1.

alter role role1 add member user2;



9. Pozbaw użytkownika user1 uprawnień do dodawania departamentów.

revert; revoke insert on departments to user1 cascade;



10. Pozbaw użytkownika user1 roli role1.

exec sp_droprolemember 'role1', 'user1';



11. Czy jako użytkownik user2 możesz nadać użytkownikowi user3 uprawnienia do dodawania departamentów?

```
execute as user = 'user2';
grant insert on departments to user3;
--nie mozna
```

12. Czy jako użytkownik user2 możesz nadać użytkownikowi user3 role role1?

```
alter role role1 add member user3;
--nie mozna

Msg 15151, Level 16, State 1, Line 60

Cannot find the object 'departments', because it does not exist or you do not have permission.

Completion time: 2025-01-09T15:06:11.5852709+01:00
```

13. Czy jako użytkownik user1 możesz nadać użytkownikowi user3 rolę role1?

```
revert;
execute as user = 'user1';
alter role role1 add member user3;
--mozna
```

```
Msg 15151, Level 16, State 1, Line 64

Cannot alter the role 'role1', because it does not exist or you do not have permission.

Completion time: 2025-01-09T15:06:20.9130724+01:00
```

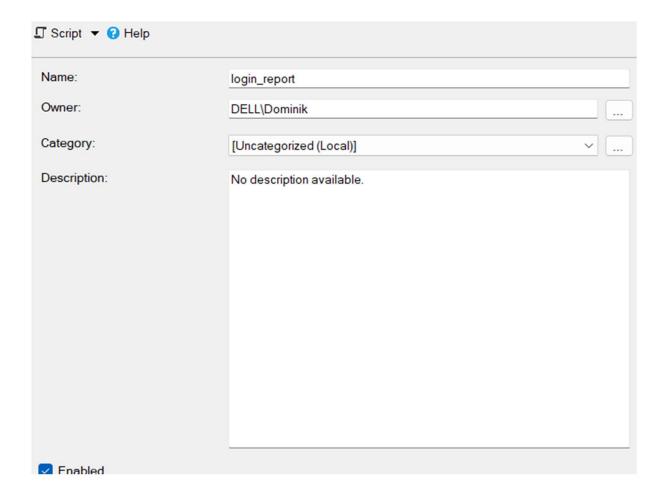
7adanie 2.

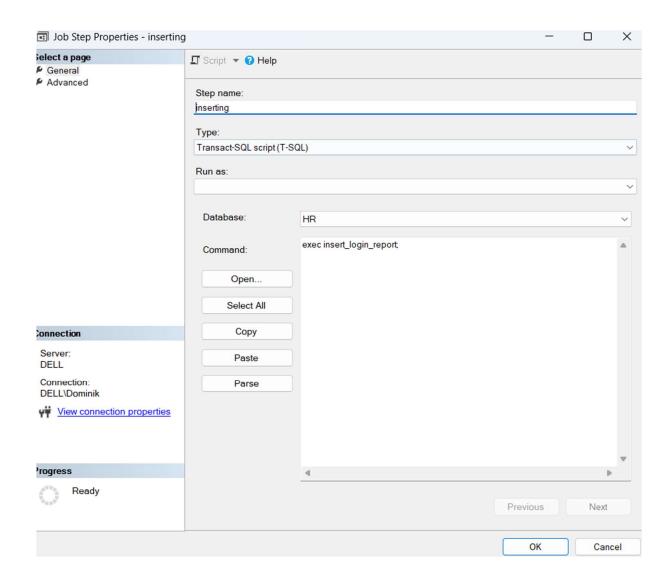
- 1. Utwórz tabelę o nazwie admin_logs, która zawierać będzie następujące pola:
 - log_id typu numerycznego będące kluczem głównym,
 - log_desc typu znakowego,
 - log_data typu datowego (z uwzględnieniem czasu).

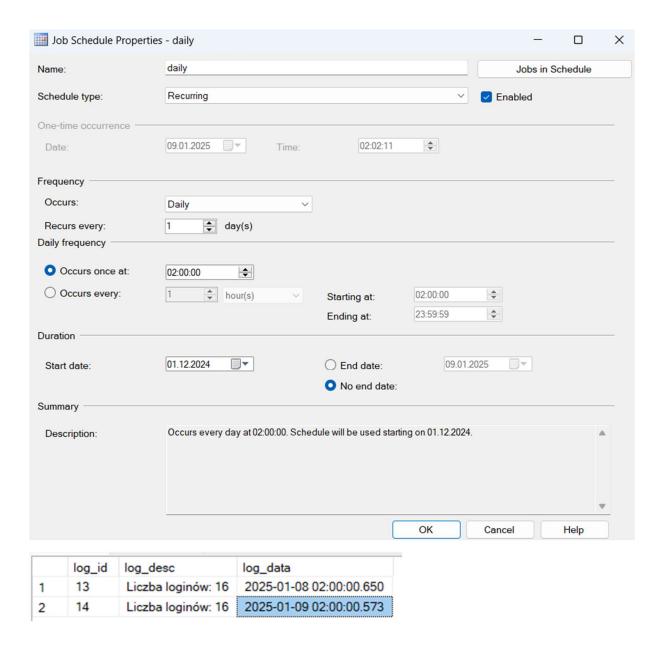
```
CREATE TABLE admin_logs ( log_id INT IDENTITY(1,1) PRIMARY KEY, log_desc NVARCHAR(255), log_data DATETIME );
```

- 2. Utwórz nowe zadanie o nazwie login_report, które:
 - przy każdym uruchomieniu doda do tabeli admin_logs wpis zawierający liczbę wszystkich loginów,
 - począwszy od 1 grudnia 2024 roku będzie wykonywane codziennie o godzinie 2:00.

```
CREATE PROCEDURE insert_login_report
AS
BEGIN
INSERT INTO admin_logs (log_desc, log_data)
SELECT CONCAT('Liczba loginów: ', COUNT(*)), GETDATE()
FROM sys.syslogins;
END;
```







PostgreSQL

Zadanie 1.

1. Utwórz nowych użytkowników o nazwach user1, user2 i user3.

```
create user user1;
create user user2;
create user user3;
```

	name 6
1	postgres
2	user1
3	user2
4	user3

2 Nadaj użytkownikom user1, user2 i user3 uprawnienia do wyświetlania departamentów.

	grant select	on	table	departments	to	user1,	user2,	user3;
	name 🔓							
1	postgres							
2	user1							
3	user2							
4	user3							

3. Utwórz nową rolę o nazwie role1 i uczyń użytkownika user1 jej właścicielem.

create role role1 with admin user1;

	role_name name	admin_role name
1	role1	user1

4. Nadaj użytkownikowi user1 uprawnienia do dodawania departamentów z możliwością dalszego przekazywania uprawnień.

grant insert on table departments to user1 with grant option;
--grant option umozliwia nadanie przywileju obiektowego innym
uzytkownikom (z grant lub bez)
--lub rolom

	grantee name	privilege_type character varying	is_grantable character varying (3)
7	postgres	TRIGGER	YES
8	user1	INSERT	YES

5. Nadaj roli role1 uprawnienia do usuwania departamentów.

grant delete on table departments to role1;

	grantee name	privilege_type character varying
1	role1	DELETE

6. Jako user1 nadaj użytkownikowi user2 uprawnienia do dodawania departamentów.

```
set role user1;
grant insert on table departments to user2;
```

	grantee name	privilege_type character varying
1	user2	INSERT

7. Jako user1 nadaj użytkownikowi user2 rolę role1.

```
set role user1;
grant role1 to user2;
```

	granted_role name	grantee name	grantor name
1	role1	user2	user1

8. Pozbaw użytkownika user1 uprawnień do dodawania departamentów.

reset role;
revoke insert on table departments from user1 CASCADE;

	grantee name	privilege_type character varying	table_name name
1	user1	SELECT	departments

9. Pozbaw użytkownika user1 roli role1.

revoke role1 from user1 CASCADE;



10. Czy jako użytkownik user2 możesz nadać użytkownikowi user3 uprawnienia do dodawania departamentów?

```
set role user2;
grant insert on table departments to user3;
--nie mozna
```

OSTRZEŻENIE: nie przyznano żadnych uprawnień do "departments" GRANT

11. Czy jako użytkownik user2 możesz nadać użytkownikowi user3 role role1?

grant role1 to user3;

```
--nie mozna
```

ERROR: Only roles with the ADMIN option on role "role1" may grant this role.permission denied to grant role "role1"

BŁAD: permission denied to grant role "role1"

12. Czy jako użytkownik user1 możesz nadać użytkownikowi user3 rolę role1?

```
reset role;
set role user1;
grant role1 to user3;
--nie mozna
```

```
ERROR: Only roles with the ADMIN option on role "role1" may grant this role.permission denied to grant role "role1"

BŁĄD: permission denied to grant role "role1"
```

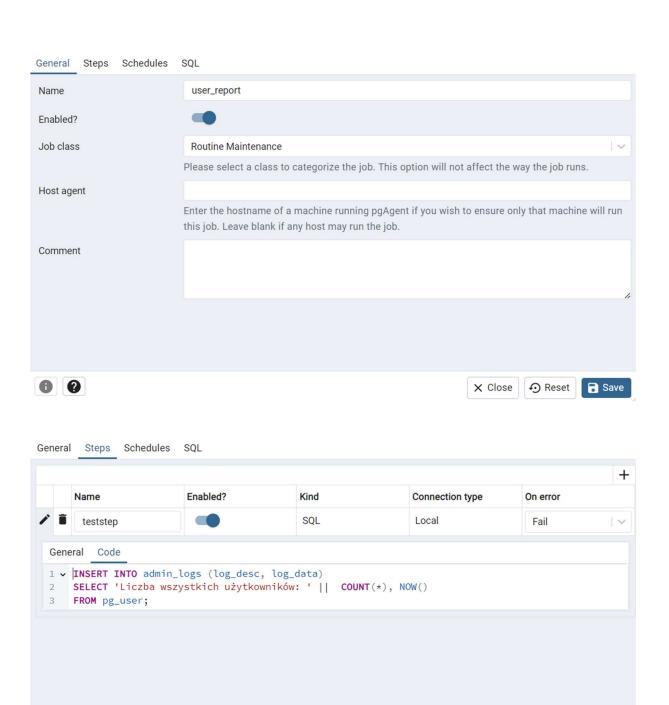
Zadanie 2.

- 1. Utwórz tabelę o nazwie admin_logs, która zawierać będzie następujące pola:
 - log_id typu numerycznego będące kluczem głównym,
 - log_desc typu znakowego,
 - log_data typu datowego (z uwzględnieniem czasu).

```
create table admin_logs(
    log_id serial primary key,
    log_desc varchar,
    log_data timestamp
)
```

- 2. Utwórz nowe zadanie o nazwie user_report, które:
 - przy każdym uruchomieniu doda do tabeli admin_logs wpis zawierający liczbę wszystkich użytkowników,
 - począwszy od 1 grudnia 2024 roku będzie wykonywane codziennie o godzinie 2:00.

```
INSERT INTO admin_logs (log_desc, log_data)
SELECT 'Liczba wszystkich użytkowników: ' || COUNT(*), NOW()
FROM pg_user;
```



X Close

• Reset

Save

