

Projekt studencki - Bazy danych

Kamil Polit

Nr. 168159

Temat projektu: Baza danych do zapisu notatek tekstowych.

Celem projektu jest zaprojektowanie schematu umożliwiającego przechowywanie notatek tekstowych w formacie txt.

Repozytorium: <https://github.com/KamilZemo/BD-Projekt>

Spis Treści:

Spis treści	1
1.Temat i założenia projektu	1
2.Funkcje projektu	2
3.Wybrane środowisko: MySQL oraz MySQL Workbench.....	3
4.Proces tworzenia bazy danych	4
5.Kod projektu	5
6.Do wypełnienia tabel używam funkcji „INSERT”	6
7.Przykładowy „SELECT”	8
8.Tworzenie pliku txt zawierającego notatki	8
9.Prosta komenda SELECT	11
10.Rozbudowana komenda SELECT.....	12
11.Funkcja update.....	12
12.Skrypt zliczający ilość notatek	13
13.Funkcja tworzenia użytkowników w bazie danych	14
14.Tworzenie wariantu testowego bazy danych	16

1. Funkcje projektu

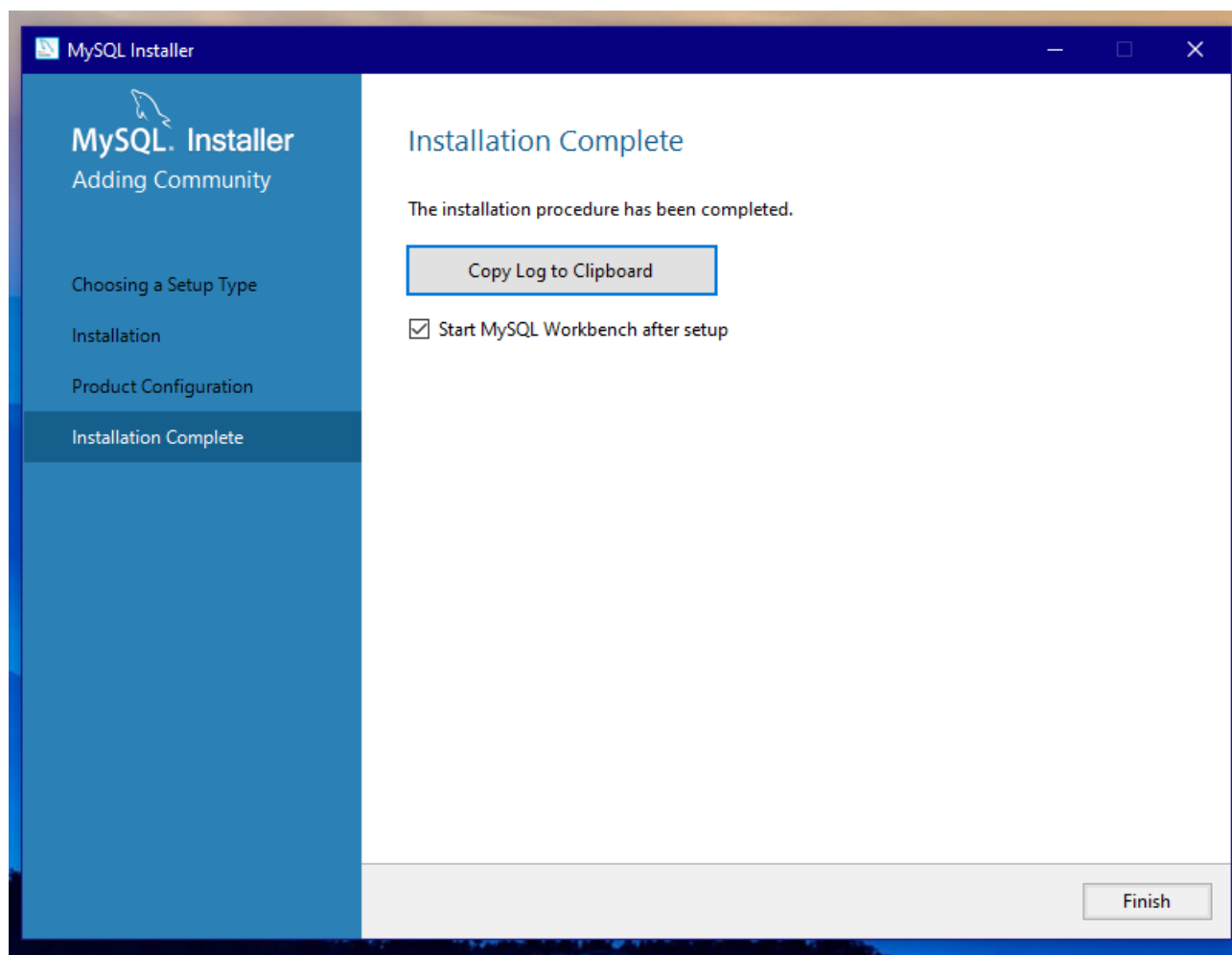
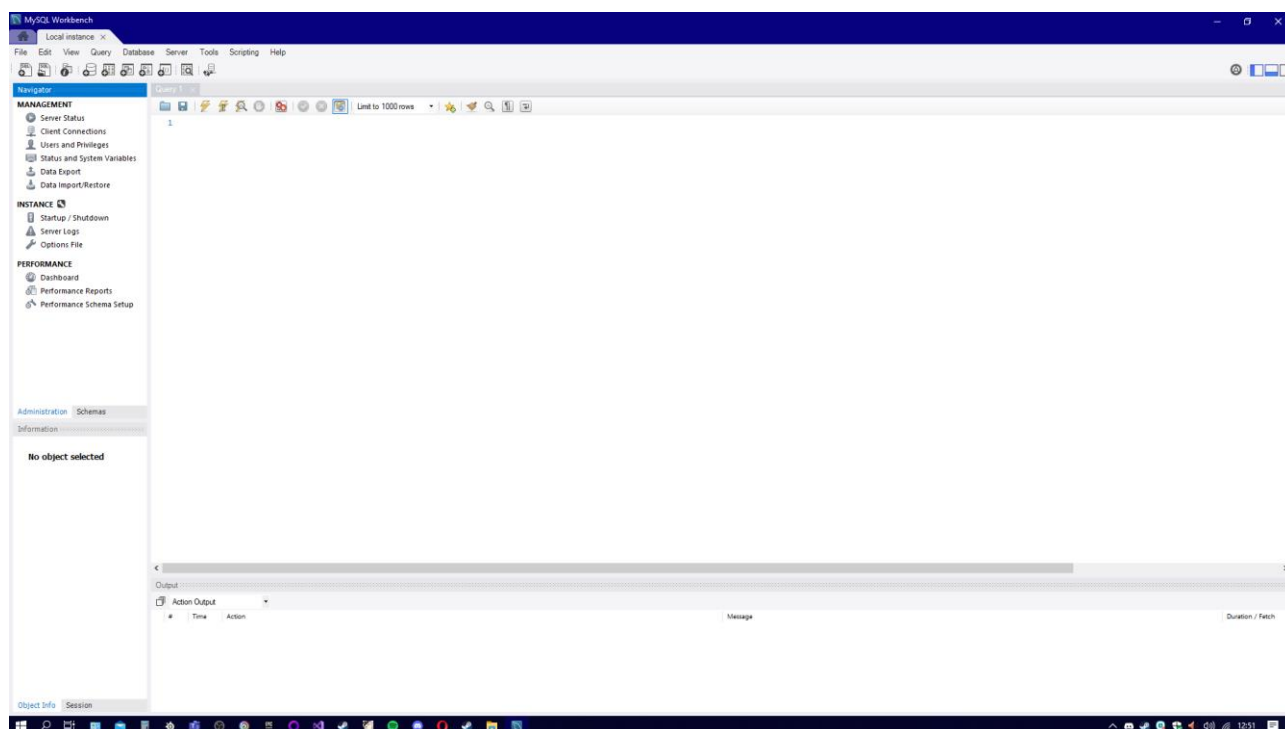
- **Zapis autora** – wykonywany będzie poprzez tabelę „User“, która zawierać będzie niezbędne informacje odnośnie użytkownika (np. imię, nazwisko, rola).
- **Zapis tematu** – umożliwiać to będzie tabela „Note_Content“, pozwala ona na zapis tematu oraz tekstu konkretnej notatki.
- **Zapis czasu dodania i usunięcia z bazy** – obie te funkcje są możliwe dzięki tabeli „Note_Info“, która pozwala na zapis dokładnych informacji dotyczących konkretnej notatki takich jak: twórca notatki, czas dodania i usunięcia oraz status notatki (publiczna czy niepubliczna).
- **Dostęp do wybranych danych historycznych** – ta funkcja umożliwiać ma przepatrywanie notatek dodanych przez użytkowników, pozwala na to tabela „Note_Content“, której zawartość zapisywana jest do pliku txt.

Zaimplementowane zostaną trzy role:

Za funkcję ról w tym projekcie odpowiadać będzie tabela „Role“.

- **Gość** – mogący jedynie przeglądać notatki ustawione przez autorów jako publiczne.
- **Edytor** – pozwalać ma użytkownikowi na dodawanie i usuwanie własnych notatek oraz przeglądanie publicznych notatek innych.
- **Administrator** – ta rola pozwala na przeglądanie wszystkich notatek oraz jeżeli zajdzie taka potrzeba usuwanie notatek użytkowników.

2. Wybrane środowisko: MySQL oraz MySQL Workbench



3. Proces tworzenia bazy danych

Projektowanie tabel – wykonany przy pomocy strony **my.vertabelo.com**

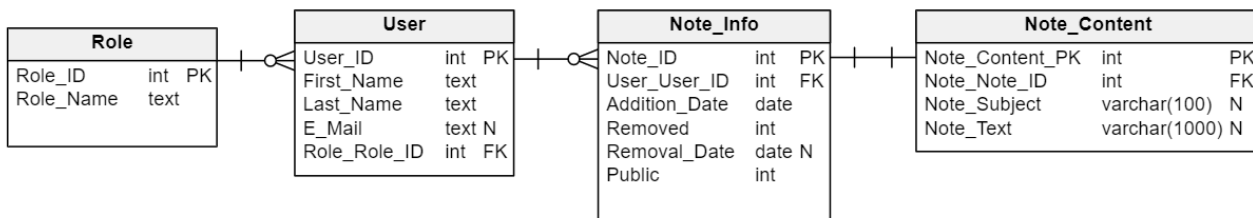


Tabela „**Role**” - odpowiada za funkcję użytkowników o różnych dostępie do funkcji bazy danych. „Role_ID” pozwala odróżnić poszczególne role. „Role_Name” jest to nazwa roli, która służy jedynie dla cech estetycznych przy wyświetlaniu. Połączona jest z tabelą „User” połączeniem 1 do wielu.

Tabela „**User**” - zawiera dokładne informacje odnośnie użytkownika w tym imię (First_Name), nazwisko (Last_Name), role (Role_Role_ID) oraz ewentualny e-mail (E_Mail). Kluczem głównym tej tabeli jest „User_ID” jest to unikalny kod dla każdego użytkownika. Tabela połączona jest po łączeniu 1 do wielu z tabelą „Note_Info”

Tabela „**Note_Info**” - znajdują się w niej ważne informacje na temat notatek, które nie są możliwe go edytowania. Kluczem głównym tej tabeli jest „Note_ID”, czyli unikalny kod każdej notatki. Znajduje się w niej także klucz obcy „User_ID”, pozwala to na szybkie rozróżnienie do kogo należy dana notatka. „Addition_Date” oraz „Remocal_Date” służą do zapisu dat utworzenia i usunięcia notatki z bazy danych. „Removed” oraz „Public” służyć będą do tego aby sprawdzić czy notatka została usunięta lub czy jest ona widoczna dla innych użytkowników. Połączona jest ona połączeniem 1 do 1 z tabelą „Note_Content”.

Tabela „**Note_Content**” – zawierają się w niej informacje, które użytkownik może edytować (poza „Note_Content_PK”, który jest kluczem głównym tabeli). Są to „Note_Subject” czyli temat notatki wpisany przez użytkownika oraz „Note_Text” zawierający treść notatki. Całość tej tabeli używana jest do zapisywania notatek do pliku w formacie txt. Obie kolumny ograniczone są odnoście ilości znaków jakie mogą zawierać.

4. Kod projektu

Funkcje tworzące tabele :

```
6 • CREATE TABLE Note_Content (  
7     Note_Content_PK int NOT NULL,  
8     Note_Note_ID int NOT NULL,  
9     Note_Subject varchar(100) NULL,  
10    Note_Text varchar(1000) NULL,  
11    CONSTRAINT Note_Content_pk PRIMARY KEY (Note_Content_PK)  
12 );  
13  
14 -- Table: Note_Info  
15 • CREATE TABLE Note_Info (  
16     Note_ID int NOT NULL,  
17     User_User_ID int NOT NULL,  
18     Addition_Date date NOT NULL,  
19     Removed int NOT NULL,  
20     Removal_Date date NULL,  
21     Public int NOT NULL,  
22     CONSTRAINT Note_Info_pk PRIMARY KEY (Note_ID)  
23 );  
24  
25 -- Table: Role  
26 • CREATE TABLE Role (  
27     Role_ID int NOT NULL,  
28     Role_Name text NOT NULL,  
29     CONSTRAINT Role_pk PRIMARY KEY (Role_ID)  
30 );
```

```

30  -- Table: User
31  ● CREATE TABLE User (
32      User_ID int NOT NULL,
33      First_Name text NOT NULL,
34      Last_Name text NOT NULL,
35      E_Mail text NULL,
36      Role_Role_ID int NOT NULL,
37      CONSTRAINT User_pk PRIMARY KEY (User_ID)
38  );
39
40  -- foreign keys
41  -- Reference: Note_Content_Note (table: Note_Content)
42  ● ALTER TABLE Note_Content ADD CONSTRAINT Note_Content_Note FOREIGN KEY Note_Content_Note (Note_Note_ID)
43      REFERENCES Note_Info (Note_ID);
44
45  -- Reference: Note_User (table: Note_Info)
46  ● ALTER TABLE Note_Info ADD CONSTRAINT Note_User FOREIGN KEY Note_User (User_User_ID)
47      REFERENCES User (User_ID);
48
49  -- Reference: User_Role (table: User)
50  ● ALTER TABLE User ADD CONSTRAINT User_Role FOREIGN KEY User_Role (Role_Role_ID)
51      REFERENCES Role (Role_ID);
52
53  -- End of file.
54

```

Powyższe zdjęcia przedstawiają fragmenty kodu odpowiedzialne za tworzenie tabel pokazanych na pierwszym zdjęciu.

5. Do wypełnienia tabel używam funkcji „INSERT”

```

1  ● INSERT role VALUES (10,'Edytor');
2  ● INSERT role VALUES (30,'Admin');
3
4  ● INSERT user VALUES (101, 'Kamil', 'Polit', NULL, 10);
5
6  ● INSERT note_info VALUES (1, 101, '2022-11-17', 0, NULL, 0, NULL);
7  ● INSERT note_info VALUES (2, 101, '2022-11-17', 0, NULL, 0, NULL);
8  ● INSERT note_info VALUES (3, 101, '2022-11-17', 0, NULL, 0, NULL);
9
10 ● INSERT note_content VALUES (223, 1,'Test 1', 'Przykład');
11 ● INSERT note_content VALUES (224, 2,'Test 2', 'Przykładowy tekst notatki');
12 ● INSERT note_content VALUES (301, 3,'Czy dzila?', 'Tu powinno byc cos napisane');

```

Po wykonaniu powyższego skryptu tabele wyglądają następująco:

Tabela Role

	Role_ID	Role_Name
▶	10	Edytor
	30	Admin
*	NULL	NULL

Tabela User

	User_ID	First_Name	Last_Name	E-Mail	Role_Role_ID
▶	101	Kamil	Polit	NULL	10
*	NULL	NULL	NULL	NULL	NULL

Tabela Note_Info

	Note_ID	User_User_ID	Addition_Date	Removed	Removal_Date	Public	Public_URL
▶	1	101	2022-11-17	0	NULL	0	NULL
	2	101	2022-11-17	0	NULL	0	NULL
	3	101	2022-11-17	0	NULL	0	NULL
*	NULL	NULL	NULL	NULL	NULL	NULL	NULL

Tabela Note_Content

	Note_Content_PK	Note_Note_ID	Note_Subject	Note_Text
▶	223	1	Test 1	Przykład
	224	2	Test 2	Przykładowy tekst notatki
	301	3	Czy działa?	Tu powinno byc cos napisane
*	NULL	NULL	NULL	NULL

6. Przykładowy „SELECT“

```
1 • USE kp_projekt_bazy_danych;
2
3 • SELECT
4     u.First_Name as Name,
5     u.Last_Name as Surname,
6     r.Role_Name,
7     ni.Note_ID,
8     nc.Note_Subject as Subject,
9     nc.Note_Text as Text
10  FROM note_info ni
11  JOIN note_content nc
12      ON ni.Note_ID = nc.Note_Note_ID
13  JOIN user u
14      ON ni.User_User_ID = u.User_ID
15  JOIN role r
16      ON u.Role_Role_ID = r.Role_ID
```

< Result Grid | Filter Rows: | Export: | Wrap Cell Content: |

	Name	Surname	Role_Name	Note_ID	Subject	Text
▶	Kamil	Polit	Edytor	1	Test 1	Przykład
	Kamil	Polit	Edytor	2	Test 2	Przykładowy tekst notatki
	Kamil	Polit	Edytor	3	Czy działa?	Tu powinno byc cos napisane

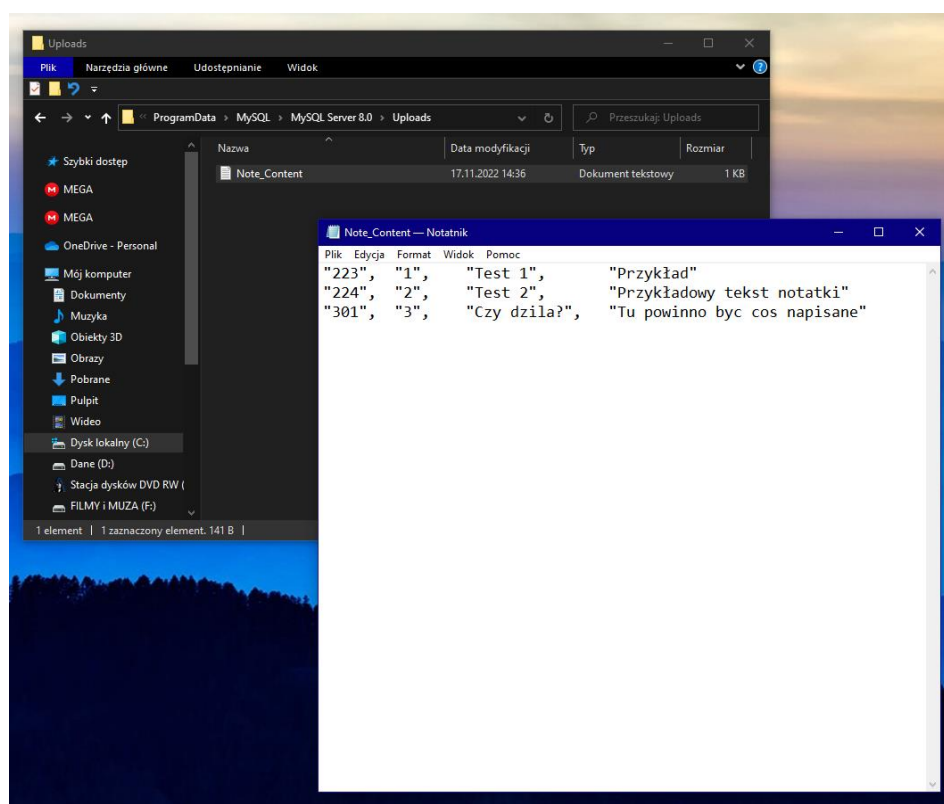
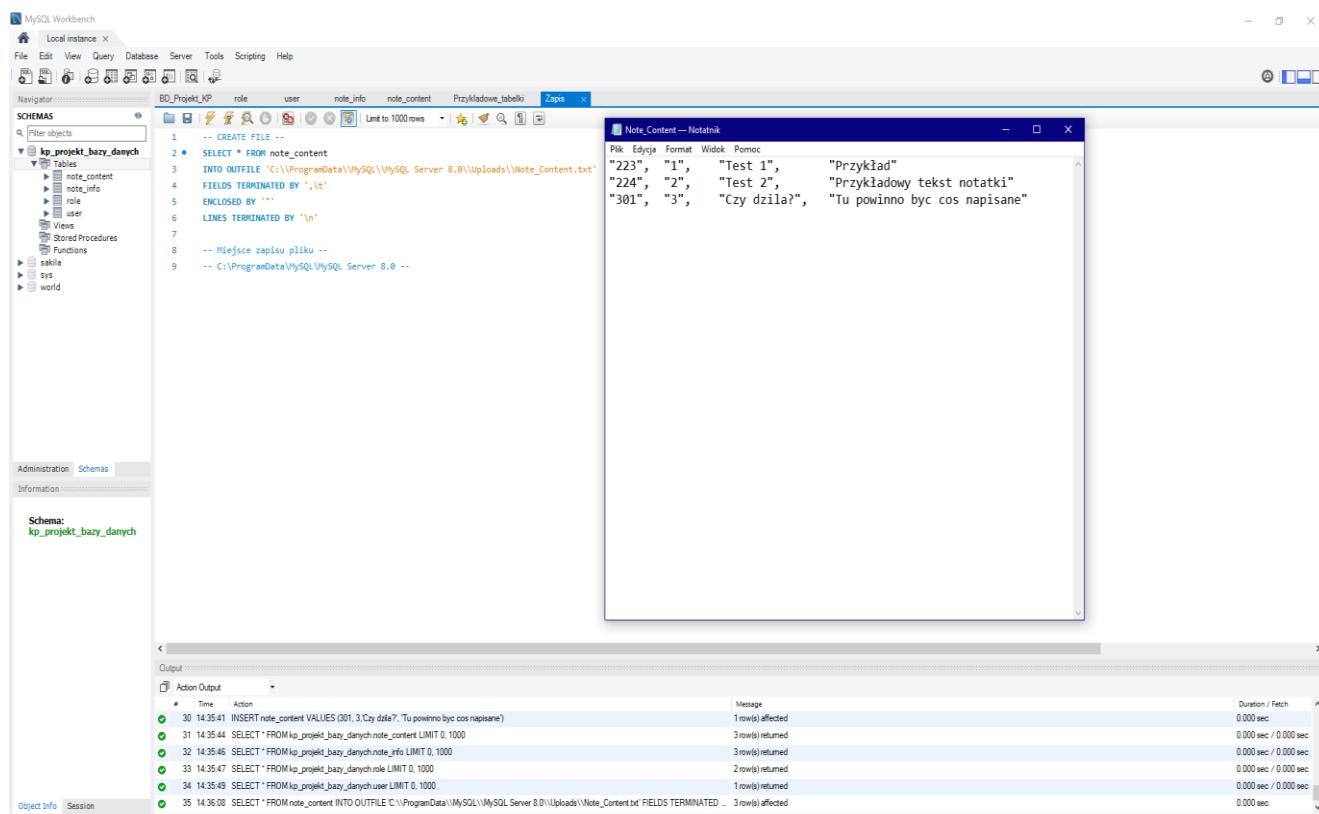
Powyższy skrypt pokazuje wybrane kolumny ze wszystkich tabel. Wyświetla imię, nazwisko, rolę, ID notatki, jej temat oraz zawartość.

7. Tworzenie pliku txt zawierającego notatki

Do tworzenia pliku .txt zawierającego całość tabeli „Note_Content” użyty został następujący skrypt:

```
-- CREATE FILE --
SELECT * FROM note_content
INTO OUTFILE 'C:\\ProgramData\\MySQL\\MySQL Server 8.0\\Uploads\\Note_Content.txt'
FIELDS TERMINATED BY ','
ENCLOSED BY ''
LINES TERMINATED BY '\n'
```


Po wykonaniu skryptu zapisu zawartość pliku txt wygląda następująco:

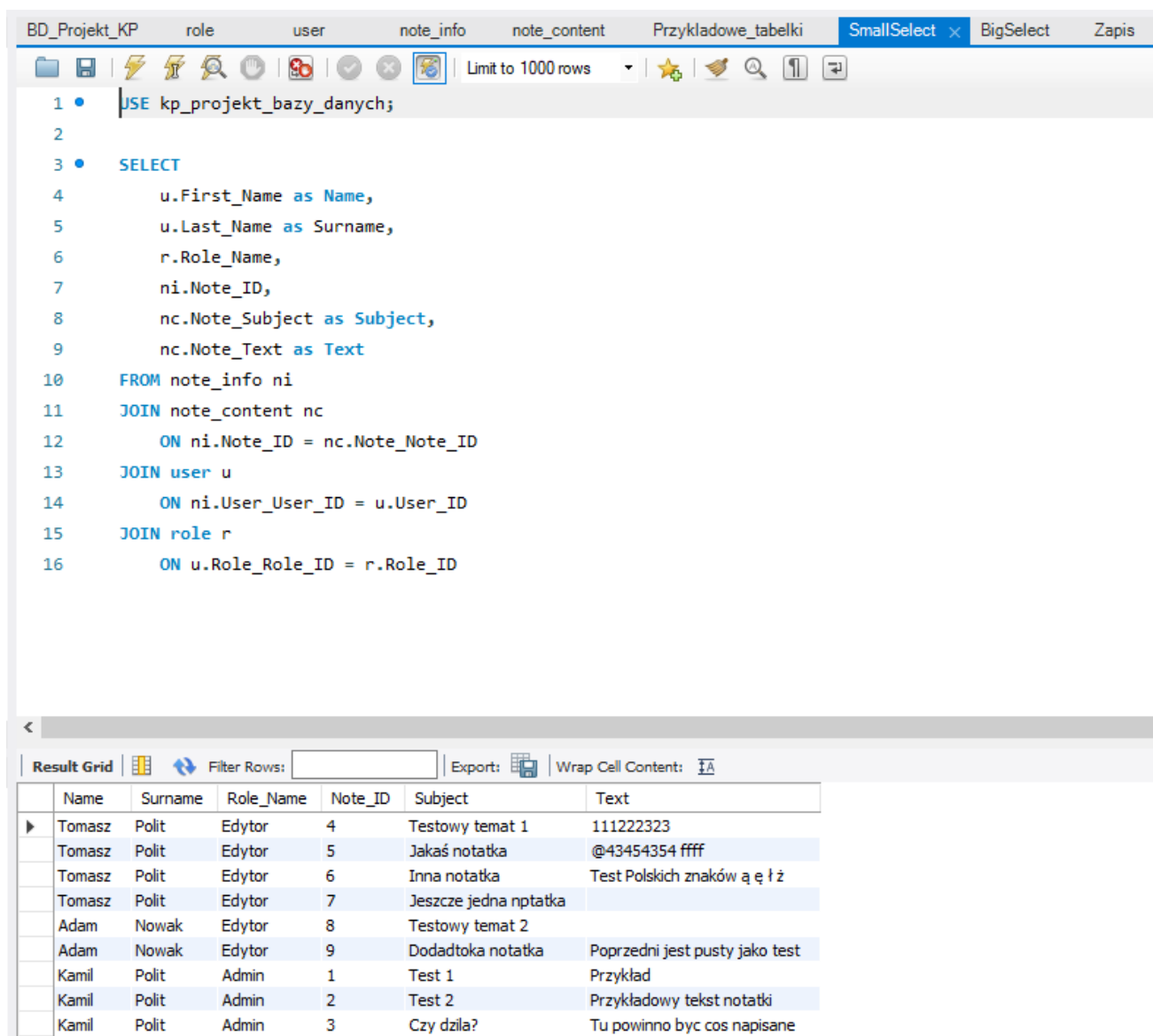


Opcja edycji zapisanych notatek jest bardzo utrudniona ze względu na fakt, że MySQL nie pozwala na usuwanie lub nadpisywanie istniejących plików. Można to jednak wykonać ręcznie usuwając lub zmieniając nazwę istniejącego pliku.

```
BD_Projekt_KP  role  user  note_info  note_content  Przykładowe_tabelki x SmallSelect  Zapis
Limit to 1000 rows
1 • INSERT role VALUES (10,'Edytor');
2 • INSERT role VALUES (30,'Admin');
3
4 • INSERT user VALUES (301, 'Kamil', 'Polit', NULL, 30);
5 • INSERT user VALUES (101, 'Tomasz', 'Polit', NULL, 10);
6 • INSERT user VALUES (102, 'Adam', 'Nowak', NULL, 10);
7 • INSERT user VALUES (103, 'Paweł', 'Kowalski', NULL, 10);
8
9 • INSERT note_info VALUES (1, 301, '2022-11-17', 0, NULL, 0);
10 • INSERT note_info VALUES (2, 301, '2022-11-17', 0, NULL, 0);
11 • INSERT note_info VALUES (3, 301, '2022-11-20', 0, NULL, 0);
12 • INSERT note_info VALUES (4, 101, '2022-11-21', 0, NULL, 0);
13 • INSERT note_info VALUES (5, 101, '2022-11-21', 0, NULL, 0);
14 • INSERT note_info VALUES (6, 101, '2022-11-24', 0, NULL, 0);
15 • INSERT note_info VALUES (7, 101, '2022-11-25', 0, NULL, 0);
16 • INSERT note_info VALUES (8, 102, '2022-11-25', 0, NULL, 0);
17 • INSERT note_info VALUES (9, 102, '2022-11-25', 0, NULL, 0);
18
19 • INSERT note_content VALUES (322, 1, 'Test 1', 'Przykład');
20 • INSERT note_content VALUES (323, 2, 'Test 2', 'Przykładowy tekst notatki');
21 • INSERT note_content VALUES (324, 3, 'Czy dzila?', 'Tu powinno byc cos napisane');
22 • INSERT note_content VALUES (202, 4, 'Testowy temat 1', '111222323');
23 • INSERT note_content VALUES (203, 5, 'Jakaś notatka', '@43454354 ffff');
24 • INSERT note_content VALUES (204, 6, 'Inna notatka', 'Test Polskich znaków ą ę ł ż');
25 • INSERT note_content VALUES (205, 7, 'Jeszcze jedna nptatka', '');
26 • INSERT note_content VALUES (190, 8, 'Testowy temat 2', '');
27 • INSERT note_content VALUES (191, 9, 'Dodadtoka notatka', 'Poprzedni jest pusty jako test');
```

Powyższy skrypt służy do wypełnienia tabel w bazie danych.

8. Prosta komenda SELECT

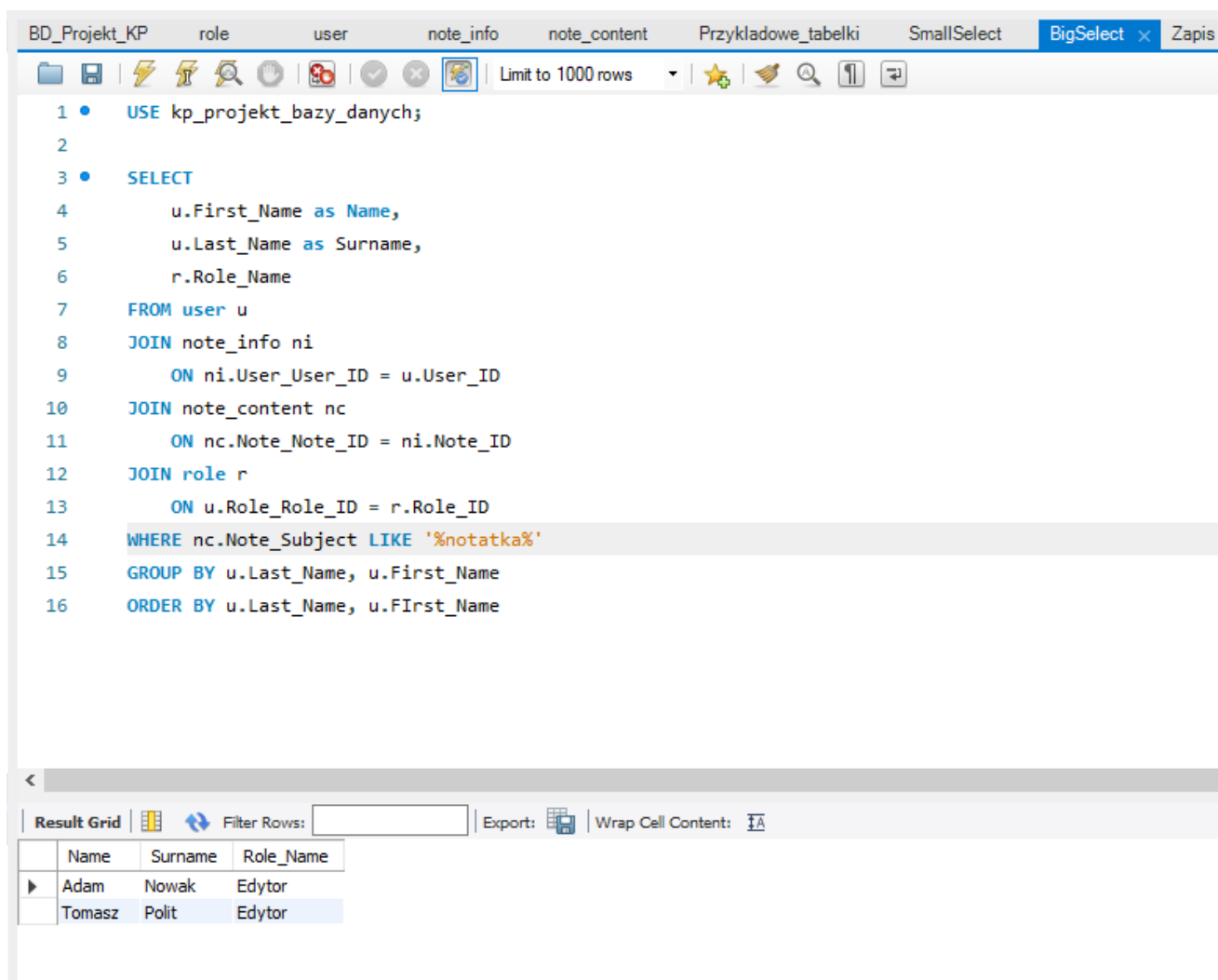


The screenshot shows a database management interface with a SQL editor and a results grid. The SQL editor contains a query that joins the 'user', 'role', 'note_info', and 'note_content' tables. The results grid displays the output of this query, showing columns for Name, Surname, Role_Name, Note_ID, Subject, and Text. The results are as follows:

	Name	Surname	Role_Name	Note_ID	Subject	Text
▶	Tomasz	Polit	Edytor	4	Testowy temat 1	111222323
	Tomasz	Polit	Edytor	5	Jakaś notatka	@43454354 ffff
	Tomasz	Polit	Edytor	6	Inna notatka	Test Polskich znaków ą ę ł ż
	Tomasz	Polit	Edytor	7	Jeszcze jedna nptatka	
	Adam	Nowak	Edytor	8	Testowy temat 2	
	Adam	Nowak	Edytor	9	Dodadtoka notatka	Poprzedni jest pusty jako test
	Kamil	Polit	Admin	1	Test 1	Przykład
	Kamil	Polit	Admin	2	Test 2	Przykładowy tekst notatki
	Kamil	Polit	Admin	3	Czy dzila?	Tu powinno byc cos napisane

Powyższy skrypt pozwala na wyświetlenie wszystkich użytkowników (ich imię, nazwisko i rolę) oraz notatek (jej ID, temat oraz zawartość).

9. Rozbudowana komenda SELECT



The screenshot shows a database management tool interface. At the top, there are tabs for different databases: BD_Projekt_KP, role, user, note_info, note_content, Przykładowe_tabelki, SmallSelect, BigSelect (selected), and Zapis. Below the tabs is a toolbar with various icons and a 'Limit to 1000 rows' dropdown. The main area displays a SQL query:

```
1 • USE kp_projekt_bazy_danych;
2
3 • SELECT
4     u.First_Name as Name,
5     u.Last_Name as Surname,
6     r.Role_Name
7 FROM user u
8 JOIN note_info ni
9     ON ni.User_User_ID = u.User_ID
10 JOIN note_content nc
11     ON nc.Note_Note_ID = ni.Note_ID
12 JOIN role r
13     ON u.Role_Role_ID = r.Role_ID
14 WHERE nc.Note_Subject LIKE '%notatka%'
15 GROUP BY u.Last_Name, u.First_Name
16 ORDER BY u.Last_Name, u.First_Name
```

Below the query, there is a 'Result Grid' section. It includes a 'Filter Rows:' input field, an 'Export:' button, and a 'Wrap Cell Content:' checkbox. The result grid shows the following data:

	Name	Surname	Role_Name
▶	Adam	Nowak	Edytor
	Tomasz	Polit	Edytor

Powyższy skrypt wyświetla wszystkie osoby, które posiadają notatkę w której temacie znajduje się fragment tekstu „notatka” z dowolnym ciągiem znaków po obu stronach. Zgrupowane oraz poukładane są one względem imienia i nazwiska.

10. Funkcja update

```
1 • USE kp_projekt_bazy_danych;
2
3 • UPDATE note_content SET Note_Subject = 'Nowy test 2'
4 WHERE Note_Note_ID = 2 AND Note_Content_PK = 323
```

Powyższe komendy służą do zmiany tematu notatki o ID 2 oraz kluczu 323.

Taką tabelę zwraca skrypt „SmallSelect” przed użyciem skryptu:

	Name	Surname	Role_Name	Note_ID	Subject	Text
▶	Tomasz	Polit	Edytor	4	Testowy temat 1	111222323
	Tomasz	Polit	Edytor	5	Jakaś notatka	@43454354 ffff
	Tomasz	Polit	Edytor	6	Inna notatka	Test Polskich znaków ą ę ł ż
	Tomasz	Polit	Edytor	7	Jeszcze jedna nptatka	
	Adam	Nowak	Edytor	8	Testowy temat 2	
	Adam	Nowak	Edytor	9	Dodadtoka notatka	Poprzedni jest pusty jako test
	Kamil	Polit	Admin	1	Test 1	Przykład
	Kamil	Polit	Admin	2	Test 2	Przykładowy tekst notatki
	Kamil	Polit	Admin	3	Czy dzila?	Tu powinno byc cos napisane

Tak wygląda tabela po użyciu podanego skryptu Update:

	Name	Surname	Role_Name	Note_ID	Subject	Text
▶	Tomasz	Polit	Edytor	4	Testowy temat 1	111222323
	Tomasz	Polit	Edytor	5	Jakaś notatka	@43454354 ffff
	Tomasz	Polit	Edytor	6	Inna notatka	Test Polskich znaków ą ę ł ż
	Tomasz	Polit	Edytor	7	Jeszcze jedna nptatka	
	Adam	Nowak	Edytor	8	Testowy temat 2	
	Adam	Nowak	Edytor	9	Dodadtoka notatka	Poprzedni jest pusty jako test
	Kamil	Polit	Admin	1	Test 1	Przykład
	Kamil	Polit	Admin	2	Nowy test 2	Przykładowy tekst notatki
	Kamil	Polit	Admin	3	Czy dzila?	Tu powinno byc cos napisane

11. **Skrypt zliczający ilość notatek** przypisanych do konkretnych użytkowników i wyświetlający ich: Imię, Nazwisko oraz ilość notatek.

```
1 • USE kp_projekt_bazy_danych;
2
3 • SELECT
4     u.First_Name as Name,
5     u.Last_Name as Surname,
6     COUNT(Note_ID) AS NumberOfNotes
7 FROM note_info ni
8 RIGHT JOIN user u
9     ON ni.User_User_ID = u.User_ID
10 GROUP BY First_Name, Last_Name
11 ORDER BY First_Name
```

	Name	Surname	NumberOfNotes
▶	Adam	Nowak	2
	Kamil	Polit	3
	Paweł	Kowalski	0
	Tomasz	Polit	4

12. Funkcja tworzenia użytkowników w bazie danych

Skrypt tworzący użytkowników bazy danych:

```

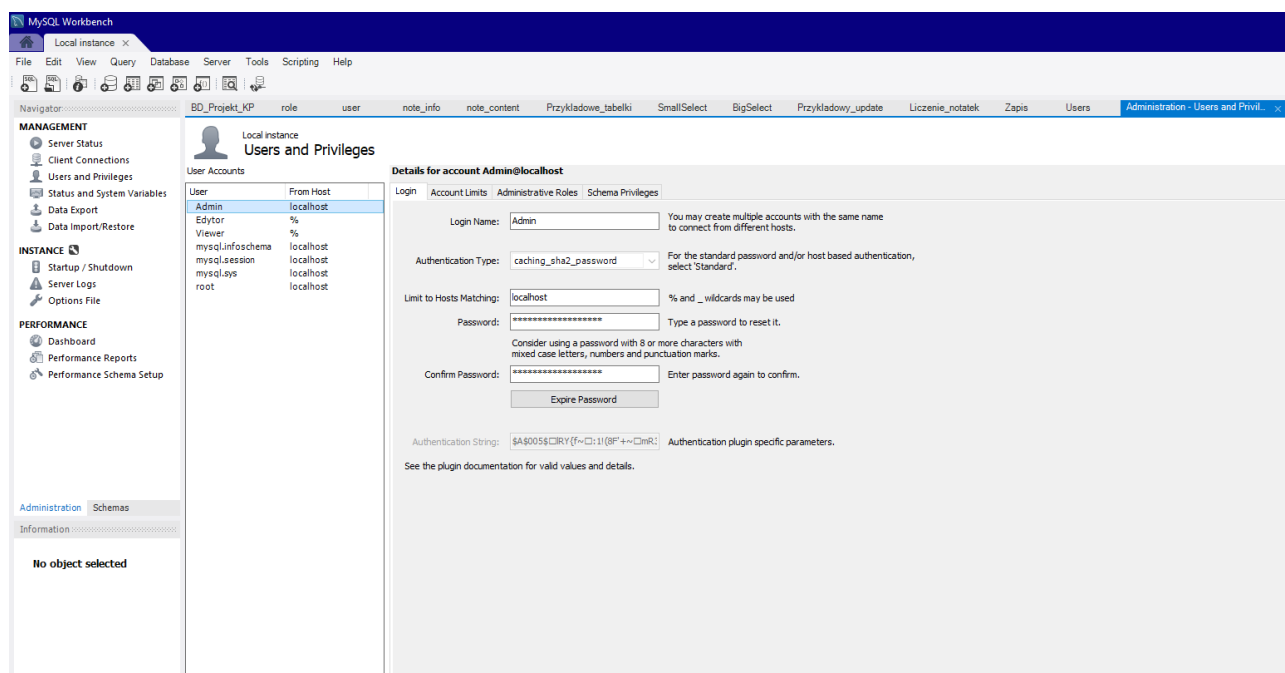
1      -- Tworzenie Admina --
2 •    CREATE USER 'Admin'@'localhost'
3          IDENTIFIED BY 'password';
4 •    GRANT ALL
5          ON *.*
6          TO 'Admin'@'localhost';
7
8      -- Tworzenie edytora --
9 •    CREATE USER 'Edytor'@'%'
10         IDENTIFIED BY 'password';
11 •    GRANT SELECT, INSERT, UPDATE, DELETE
12         ON kp_projekt_bazy_danych.*
13         TO 'Edytor'@'%';
14
15     -- Tworzenie osobę przeglądającą --
16 •    CREATE USER 'Viewer'@'%';
17 •    GRANT SELECT
18         ON kp_projekt_bazy_danych.*
19         TO 'Viewer'@'%';

```

Po wykonaniu skryptu zwracany jest ten komunikat:

Output			
Action Output			
#	Time	Action	
✓ 51	21:55:03	CREATE USER 'Admin'@'localhost' IDENTIFIED BY 'password'	
✓ 52	21:55:03	GRANT ALL ON *.* TO 'Admin'@'localhost'	
✓ 53	21:55:03	CREATE USER 'Edytor'@'%' IDENTIFIED BY 'password'	
✓ 54	21:55:03	GRANT SELECT, INSERT, UPDATE, DELETE ON kp_projekt_bazy_danych.* TO 'Edytor'@'%'	
✓ 55	21:55:03	CREATE USER 'Viewer'@'%'	
✓ 56	21:55:03	GRANT SELECT ON kp_projekt_bazy_danych.* TO 'Viewer'@'%'	

Potwierdzenie utworzenia nowych użytkowników:



Jakie możliwości mają poszczególne role w bazie danych:

- **Gość** – ta rola umożliwia jedynie używanie komendy SELECT. Nie jest zamknięta za żadnym hasłem i może jej używać każdy.
- **Edytor** – może używać komend: SELECT, INSERT, UPDATE, DELETE oraz jest zamknięta za hasłem. Oznacza to, że użytkownik może jedynie zmieniać zawartość tabel bazy danych i je przeglądać ale nie może w żaden sposób wpływać na konstrukcję bazy.
- **Administrator** – rola zezwala użytkownikowi na wykorzystywanie wszystkich komend wewnątrz bazy. Oznacza to że Administrator ma możliwość na zmianę struktury bazy. Jest ona również chroniona hasłem.

13. Tworzenie wariantu testowego bazy danych

Skrypt –

```
1 • CREATE SCHEMA `NEW_kp_projekt_bazy_danych`;  
2 • USE NEW_kp_projekt_bazy_danych;  
3  
4 • CREATE TABLE New_role LIKE kp_projekt_bazy_danych.role;  
5 • INSERT New_role SELECT * FROM kp_projekt_bazy_danych.role;  
6  
7 • CREATE TABLE New_user LIKE kp_projekt_bazy_danych.user;  
8 • INSERT New_user SELECT * FROM kp_projekt_bazy_danych.user;  
9  
10 • CREATE TABLE New_note_info LIKE kp_projekt_bazy_danych.note_info;  
11 • INSERT New_note_info SELECT * FROM kp_projekt_bazy_danych.note_info;  
12  
13 • CREATE TABLE New_note_content LIKE kp_projekt_bazy_danych.note_content;  
14 • INSERT New_note_content SELECT * FROM kp_projekt_bazy_danych.note_content;
```

Output –

Output			
Action Output			
	#	Time	Action
✓	61	22:14:16	CREATE TABLE New_user LIKE kp_projekt_bazy_danych.user
✓	62	22:14:16	INSERT New_user SELECT * FROM kp_projekt_bazy_danych.user
✓	63	22:14:16	CREATE TABLE New_note_info LIKE kp_projekt_bazy_danych.note_info
✓	64	22:14:16	INSERT New_note_info SELECT * FROM kp_projekt_bazy_danych.note_info
✓	65	22:14:16	CREATE TABLE New_note_content LIKE kp_projekt_bazy_danych.note_content
✓	66	22:14:16	INSERT New_note_content SELECT * FROM kp_projekt_bazy_danych.note_content

Potwierdzenie utworzenia nowej schem'y –

