



МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
“КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
ІМЕНІ ІГОРЯ СІКОРСЬКОГО”

Факультет прикладної математики
Кафедра програмного забезпечення комп’ютерних систем

Лабораторна робота № 3
з дисципліни “Бази даних”
тема “Засоби оптимізації роботи СУБД PostgreSQL”

Виконала
студентка II курсу
групи КП-03

Сіренко Вікторія Юріївна
(прізвище, ім'я, по батькові)

варіант № 16

Перевірів
“ ____ ” “ ____ ” 20__ р.
викладач

Радченко Константин
Олександрович
(прізвище, ім'я, по батькові)

Київ 2021

Мета роботи

Метою роботи є здобуття практичних навичок використання засобів оптимізації СУБД PostgreSQL.

Постановка завдання

1. Перетворити модуль “Модель” з шаблону MVC лабораторної роботи No2 у вигляд об’єктно-реляційної проекції (ORM).
2. Створити та проаналізувати різні типи індексів у PostgreSQL.
3. Розробити тригер бази даних PostgreSQL.

Варіант 16:

Види індекси: GIN, Hash

Умови тригера: after delete, insert

16	GIN, Hash	after delete, insert
----	-----------	----------------------

Аналіз вимог і проектування

Вимоги до пункту завдання №1

Для перетворення функцій, що реалізують запити до об’єктної бази даних, необхідно встановити бібліотеку sqlalchemy, налаштувати програму на роботу з ORM, розробити класи-сутності для об’єктів-сутностей, представлених відповідними таблицями БД та пов’язаних зв’язками 1:M, M:M та 1:1 виконати опис схеми бази даних. Особливу увагу приділити контролю зовнішніх зв’язків між таблицями засобами ORM.

Замінити виклики запитів мовою SQL на відповідні запити засобами SQLAlchemy по роботі з об’єктами. Обов’язковим є реалізація вставки, вилучення та редагування екземплярів класів-сутностей. Розробка запитів на генерацію даних та пошук екземплярів класів-сутностей вітається, але не є

обов'язковою.

Інтерфейси функцій (вхідні та вихідні аргументи функцій модуля “Модель”) мають залишитись без змін.

Вимоги до пункту завдання №2

Відповідно до варіанту індексування продемонструвати на прикладах запитів SQL SELECT підвищення швидкодії їх виконання з використанням індексів, а також пояснити чому для деяких випадків індексування використовувати недоцільно. При цьому для наочного представлення слід використати функцію генерування рандомізованих даних з лабораторної роботи No2, створивши необхідну кількість тестових даних. Навести 4-5 прикладів запитів SELECT (із виведенням результуючих даних), що містять фільтрацію, агрегатні функції, групування та сортування (у необхідних комбінаціях).

Вимоги до пункту завдання №3

Створити тригер бази даних PostgreSQL відповідно до варіанта. Тригерна функція має включати обробку запису, що модифікується (вставляється або вилучається), умовні оператори, курсорні цикли та обробку виключних ситуацій. Виконати відлагодження тригера при різних вхідних даних, навівши 2-3 приклади його використання.

Вимоги до інструментарію

1. Бібліотека для реалізації ORM - SQLAlchemy для Python або інша з подібною функціональністю.
2. Середовище для відлагодження SQL-запитів до бази даних – pgAdmin 4.
3. СУБД - PostgreSQL 11-12.

Вимоги до оформлення лабораторної роботи у електронному вигляді

Опис та вміст репозиторію лабораторної роботи у репозиторії GitHub

включає: оновлені файли додатку, назву лабораторної роботи та варіант студента.

Звіт лабораторної роботи має містити:

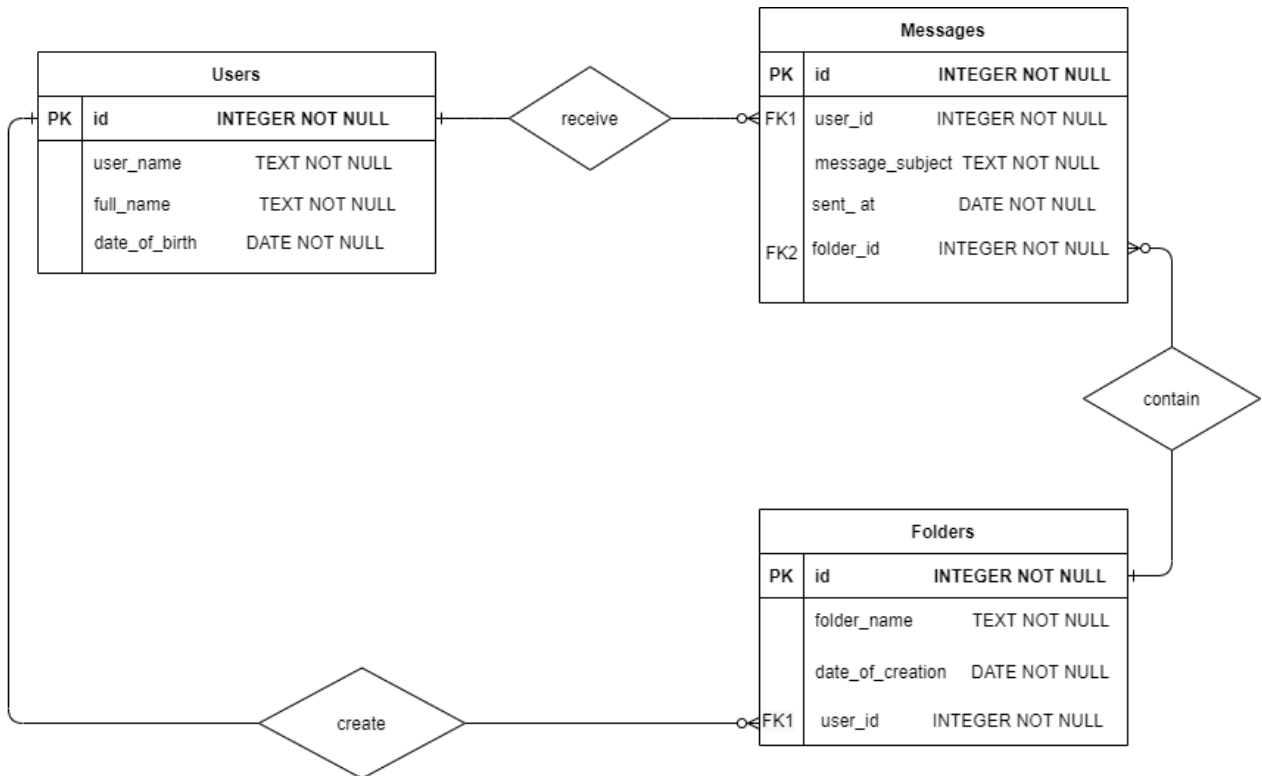
- титульний аркуш затвердженого зразка;
- завдання на лабораторну роботу з обов'язковим наведенням варіанту студента;
- копії екрану (скріншоти), що підтверджують вимоги 1-3 завдання, а також:
- для завдання №1: схему бази даних у вигляді таблиць і зв'язків між ними, а також класи ORM, що відповідають таблицям бази даних.

Навести приклади запитів у вигляді ORM.

- для завдання №2: команди створення індексів, тексти, результати і час виконання запитів SQL, пояснити чому індекси прискорюють (або не прискорюють) швидкість виконання запитів.
- для завдання №3: команди, що ініціюють виконання тригера, текст тригера та скріншоти зі змінами у таблицях бази даних;

Виконання

Завдання 1:



Функції, які були перетворені у вигляд об'єктно-реляційної проєкції (ORM):

Model.cs

```
1. public class FolderRepository
2. {
3.     private DbContext db;
4.     private NpgsqlConnection connection;
5.     public FolderRepository(DbContext dbContext, NpgsqlConnection
6. connection)
7.     {
8.         this.db = dbContext;
9.         this.connection=connection;
10.    }
11.    // adds a folder to the database
12.    public void Insert(Folder folder)
13.    {
14.        User user = db.users.Find(folder.UserId);
15.        if (user != null)
16.        {
17.            db.folders.Add(folder);
18.            user.Folders.Add(folder);
19.            db.SaveChanges();
20.        }
21.    }
22.
23.
24.    //returns a folder on its id
25.    public Folder GetByFolderId(int id)
```

```

26.     {
27.
28.         Folder folder = db.folders.Find(id);
29.         return folder;
30.
31.     }
32.
33.
34.
35.
36.     // updates the folder
37.     public bool Update(Folder folder, int folderId)
38.     {
39.         var newFolder = db.folders.Find(folderId);
40.         newFolder.Messages = folder.Messages;
41.         newFolder.folder_name = folder.folder_name;
42.         newFolder.Userid = folder.Userid;
43.         int res = db.SaveChanges();
44.         return res == 1;
45.
46.     }
47.
48.     //removes the folder by id
49.     public bool Delete(int folderId)
50.     {
51.
52.         Folder folder = GetByFolderId(folderId);
53.         if (folder != null)
54.         {
55.             db.folders.Remove(folder);
56.             db.SaveChanges();
57.             return true;
58.         }
59.         return false;
60.
61.     }
62.
63.     public void DeleteAllByUserId(int userId)
64.     {
65.         var folders = db.folders.Where(p => p.Userid == userId);
66.         foreach (Folder folder in folders)
67.         {
68.             db.folders.Remove(folder);
69.         }
70.         db.SaveChanges();
71.
72.     }
73.
74. }
75. public class MessageRepository
76. {
77.     private DbContext db;
78.     private NpgsqlConnection connection;
79.     public MessageRepository(DbContext dbContext, NpgsqlConnection
connection)
80.     {
81.         this.db = dbContext;
82.         this.connection=connection;
83.     }
84.
85.     // adds a message to the database
86.     public void Insert(Message message)
87.     {

```

```

88.
89.     User user = db.users.Find(message.Userid);
90.     Folder folder = db.folders.Find(message.Folderid);
91.
92.     if (user != null && folder != null)
93.     {
94.         db.messages.Add(message);
95.         user.Messages.Add(message);
96.         folder.Messages.Add(message);
97.         db.SaveChanges();
98.     }
99.
100. }
101.
102.
103. //returns a message on its id
104. public Message GetByIdMessageId(int id)
105. {
106.     Message message = db.messages.Find(id);
107.     return message;
108. }
109.
110.
111.
112. // updates the message
113. public bool Update(Message message, int messageId)
114. {
115.     var newMessage = db.messages.Find(messageId);
116.     newMessage.message_subject = message.message_subject;
117.     newMessage.Userid = message.Userid;
118.     newMessage.Folderid = message.Folderid;
119.     int res = db.SaveChanges();
120.     return res == 1;
121. }
122.
123.
124. //removes the message by id
125. public bool Delete(int messageId)
126. {
127.
128.     Message message = GetByIdMessageId(messageId);
129.     if (message != null)
130.     {
131.         db.messages.Remove(message);
132.         db.SaveChanges();
133.         return true;
134.     }
135.     return false;
136. }
137.
138.
139. public void DeleteAllByUserId(int userId)
140. {
141.     var messages = db.messages.Where(p => p.Userid == userId);
142.     foreach (Message message in messages)
143.     {
144.         db.messages.Remove(message);
145.     }
146.     db.SaveChanges();
147. }
148.
149.
150. public void DeleteAllByFolderId(int folderId)

```

```

151.     {
152.         var messages = db.messages.Where(p => p.Folderid == folderId);
153.         foreach (Message message in messages)
154.         {
155.             db.messages.Remove(message);
156.         }
157.         db.SaveChanges();
158.     }
159. }
160. }
161. public class UserRepository
162. {
163.     private DatabaseContext db;
164.     private NpgsqlConnection connection;
165.     public UserRepository(DatabaseContext databaseContext, NpgsqlConnection
connection)
166.     {
167.         this.db = databaseContext;
168.         this.connection=connection;
169.     }
170.
171.     //adds a new user to the database
172.     public void Insert(User user)
173.     {
174.         db.users.Add(user);
175.         db.SaveChanges();
176.     }
177.
178.
179.     //deletes the user by his ID
180.     public bool Delete(int id)
181.     {
182.
183.         User user = GetByUserId(id);
184.         if (user != null)
185.         {
186.             db.users.Remove(user);
187.             db.SaveChanges();
188.             return true;
189.         }
190.         return false;
191.     }
192. }
193.
194.
195.     //updates the user
196.     public bool Update(User user, int userId)
197.     {
198.         var newUser = db.users.Find(userId);
199.         newUser.user_name = user.user_name;
200.         newUser.full_name = user.full_name;
201.         int res = db.SaveChanges();
202.         return res == 1;
203.     }
204.
205.
206.     public User GetByUserId(int id)
207.     {
208.         User user = db.users.Find(id);
209.         return user;
210.     }
211. }
212. }

```


Завдання 2:

Пошук користувача за підрядком user_name та підрядком full_name :

```
On what entity do you want to perform actions?('user' / 'message' / 'folder')
or 'exit'
user
What do you want to do with the user table?('add' / 'delete' / 'edit' /
'find' / 'generate')
find
Enter the username you want to find
v
Enter the full name of the user you want to find
ca
User: Id:9 | Username:WVY | Fullname:ca6e544c
User: Id:30 | Username:NNV | Fullname:98f9bca4
User: Id:271 | Username:VUL | Fullname:c19b6ca7
User: Id:463 | Username:RVV | Fullname:237cae35
User: Id:759 | Username:VHY | Fullname:13ca1128
User: Id:802 | Username:GJV | Fullname:96cca044
User: Id:1007 | Username:VQQ | Fullname:fcadea0c
User: Id:1571 | Username:SNV | Fullname:9fe32cca
User: Id:2164 | Username:VFV | Fullname:bca8d323
User: Id:2747 | Username:KEV | Fullname:b88cae47
User: Id:3317 | Username:YKV | Fullname:b1ebca8b
User: Id:3440 | Username:VIS | Fullname:5cade6b7
User: Id:3466 | Username:VUP | Fullname:22f1a2ca
User: Id:3467 | Username:XSV | Fullname:137e1caf
User: Id:3843 | Username:VSI | Fullname:efbcaf8e
User: Id:4159 | Username:VKY | Fullname:4ca519a0
User: Id:4275 | Username:IVA | Fullname:84ccac45
User: Id:4425 | Username:AVL | Fullname:ca13131b
User: Id:4712 | Username:VER | Fullname:9a4cabff
User: Id:4732 | Username:HVE | Fullname:c0ca13ca
User: Id:4871 | Username:VFG | Fullname:8b935ca8
User: Id:5144 | Username:VVX | Fullname:d2d612ca
User: Id:5313 | Username:ULV | Fullname:c69ca07d
User: Id:5613 | Username:FVF | Fullname:625ca91e
User: Id:5690 | Username:VBH | Fullname:ca5056ad
User: Id:5902 | Username:IOV | Fullname:19c46ca7
User: Id:6421 | Username:SHV | Fullname:1ab971ca
User: Id:7591 | Username:VGU | Fullname:27ca001c
User: Id:7993 | Username:VHY | Fullname:08ddca5e
User: Id:8063 | Username:UEV | Fullname:caf18a97
User: Id:8649 | Username:VMS | Fullname:176ca134
User: Id:9488 | Username:HVV | Fullname:caa157d8
User: Id:9576 | Username:TVX | Fullname:dcab9cbf
User: Id:9625 | Username:HVX | Fullname:c2c19ca8
User: Id:9711 | Username:GOV | Fullname:caf62f44
User: Id:10166 | Username:FVU | Fullname:dcad6aeb
User: Id:11247 | Username:VDN | Fullname:6c557aca
User: Id:11682 | Username:VPH | Fullname:84eda2ca
User: Id:11743 | Username:NKV | Fullname:3ccf21ca
User: Id:11802 | Username:TVR | Fullname:6d709ca5
User: Id:12283 | Username:WVE | Fullname:1ca5ebb7
User: Id:12457 | Username:WVJ | Fullname:cab3c695
```

```
User: Id:12558 | Username:LVV | Fullname:ca2f59ce
User: Id:13159 | Username:BMV | Fullname:6518cacf
User: Id:13560 | Username:TXV | Fullname:cae0fe4e
User: Id:13971 | Username:NHV | Fullname:e309ca94
User: Id:14571 | Username:TWV | Fullname:bbecab11
User: Id:14709 | Username:VGI | Fullname:0ca1eed1
User: Id:15372 | Username:XWV | Fullname:cae5871d
User: Id:15514 | Username:TLV | Fullname:c9ca56b7
User: Id:16014 | Username:GTV | Fullname:dca684dd
User: Id:16609 | Username:DVD | Fullname:dd76caa5
User: Id:16903 | Username:KVY | Fullname:ca159203
User: Id:17211 | Username:RVB | Fullname:caf819a6
User: Id:17555 | Username:CHV | Fullname:9eb79cad
User: Id:18021 | Username:JVC | Fullname:ca809f2b
User: Id:18452 | Username:VKW | Fullname:be3046ca
User: Id:18599 | Username:VPT | Fullname:1caec03a
User: Id:18674 | Username:PSV | Fullname:5f01c1ca
User: Id:18911 | Username:VQA | Fullname:e9caab2a
User: Id:19419 | Username:TVK | Fullname:ca3441d9
User: Id:19598 | Username:VER | Fullname:ccad5b5f
User: Id:19778 | Username:BTB | Fullname:57cdca9b
User: Id:19786 | Username:VQV | Fullname:20c78ca7
Time: 6766 ms
```

```
User: Id:17555 | Username:CHV | Fullname:9eb79cad
User: Id:18021 | Username:JVC | Fullname:ca809f2b
User: Id:18452 | Username:VKW | Fullname:be3046ca
User: Id:18599 | Username:VPT | Fullname:1caec03a
User: Id:18674 | Username:PSV | Fullname:5f01c1ca
User: Id:18911 | Username:VQA | Fullname:e9caab2a
User: Id:19419 | Username:TVK | Fullname:ca3441d9
User: Id:19598 | Username:VER | Fullname:ccad5b5f
User: Id:19778 | Username:BTB | Fullname:57cdca9b
User: Id:19786 | Username:VQV | Fullname:20c78ca7
Time: 6766 ms
```

Пошук користувача за підрядком user_name та підрядком full_name з використанням індексу GIN:

```
On what entity do you want to perform actions?('user' / 'message' / 'folder')
or 'exit'
user
What do you want to do with the user table?('add' / 'delete' / 'edit' /
'find' / 'generate')
find
Enter the username you want to find
v
Enter the full name of the user you want to find
ca
```

User:	Id:9		Username:WVY		Fullname:ca6e544c
User:	Id:30		Username:NNV		Fullname:98f9bca4
User:	Id:271		Username:VUL		Fullname:c19b6ca7
User:	Id:463		Username:RVV		Fullname:237cae35
User:	Id:759		Username:VHY		Fullname:13ca1128
User:	Id:802		Username:GJV		Fullname:96cca044
User:	Id:1007		Username:VQQ		Fullname:fcadea0c
User:	Id:1571		Username:SNV		Fullname:9fe32cca
User:	Id:2164		Username:VFV		Fullname:bca8d323
User:	Id:2747		Username:KEV		Fullname:b88cae47
User:	Id:3317		Username:YKV		Fullname:b1ebca8b
User:	Id:3440		Username:VIS		Fullname:5cade6b7
User:	Id:3466		Username:VUP		Fullname:22f1a2ca
User:	Id:3467		Username:XSV		Fullname:137e1caf
User:	Id:3843		Username:VSI		Fullname:efbcaf8e
User:	Id:4159		Username:VKY		Fullname:4ca519a0
User:	Id:4275		Username:IVA		Fullname:84ccac45
User:	Id:4425		Username:AVL		Fullname:ca13131b
User:	Id:4712		Username:VER		Fullname:9a4cabff
User:	Id:4732		Username:HVE		Fullname:c0ca13ca
User:	Id:4871		Username:VFG		Fullname:8b935ca8
User:	Id:5144		Username:VVX		Fullname:d2d612ca
User:	Id:5313		Username:ULV		Fullname:c69ca07d
User:	Id:5613		Username:FVF		Fullname:625ca91e
User:	Id:5690		Username:VBH		Fullname:ca5056ad
User:	Id:5902		Username:IOV		Fullname:19c46ca7
User:	Id:6421		Username:SHV		Fullname:1ab971ca
User:	Id:7591		Username:VGU		Fullname:27ca001c
User:	Id:7993		Username:VHY		Fullname:08ddca5e
User:	Id:8063		Username:UEV		Fullname:caf18a97
User:	Id:8649		Username:VMS		Fullname:176ca134
User:	Id:9488		Username:HVV		Fullname:caa157d8
User:	Id:9576		Username:TVX		Fullname:dcab9cbf
User:	Id:9625		Username:HVX		Fullname:c2c19ca8
User:	Id:9711		Username:GOV		Fullname:caf62f44
User:	Id:10166		Username:FVU		Fullname:dcad6aeb
User:	Id:11247		Username:VDN		Fullname:6c557aca
User:	Id:11682		Username:VPH		Fullname:84eda2ca
User:	Id:11743		Username:NKV		Fullname:3ccf21ca
User:	Id:11802		Username:TVR		Fullname:6d709ca5
User:	Id:12283		Username:WVE		Fullname:1ca5ebb7
User:	Id:12457		Username:WVJ		Fullname:cab3c695
User:	Id:12558		Username:LVV		Fullname:ca2f59ce
User:	Id:13159		Username:BMV		Fullname:6518cacf
User:	Id:13560		Username:TXV		Fullname:cae0fe4e
User:	Id:13971		Username:NHV		Fullname:e309ca94
User:	Id:14571		Username:TWV		Fullname:bbecab11
User:	Id:14709		Username:VGI		Fullname:0caleed1
User:	Id:15372		Username:XWV		Fullname:cae5871d
User:	Id:15514		Username:TLV		Fullname:c9ca56b7
User:	Id:16014		Username:GTV		Fullname:dca684dd
User:	Id:16609		Username:DVD		Fullname:dd76caa5
User:	Id:16903		Username:KVY		Fullname:ca159203
User:	Id:17211		Username:RVB		Fullname:caf819a6
User:	Id:17555		Username:CHV		Fullname:9eb79cad
User:	Id:18021		Username:JVC		Fullname:ca809f2b
User:	Id:18452		Username:VKW		Fullname:be3046ca
User:	Id:18599		Username:VPT		Fullname:1caec03a

```
User: Id:18674 | Username:PSV | Fullname:5f01c1ca
User: Id:18911 | Username:VQA | Fullname:e9caab2a
User: Id:19419 | Username:TVK | Fullname:ca3441d9
User: Id:19598 | Username:VER | Fullname:ccad5b5f
User: Id:19778 | Username:BTB | Fullname:57cdca9b
User: Id:19786 | Username:VQV | Fullname:20c78ca7
Time: 2833 ms
```

```
User: Id:18452 | Username:VKW | Fullname:be3046ca
User: Id:18599 | Username:VPT | Fullname:1caec03a
User: Id:18674 | Username:PSV | Fullname:5f01c1ca
User: Id:18911 | Username:VQA | Fullname:e9caab2a
User: Id:19419 | Username:TVK | Fullname:ca3441d9
User: Id:19598 | Username:VER | Fullname:ccad5b5f
User: Id:19778 | Username:BTB | Fullname:57cdca9b
User: Id:19786 | Username:VQV | Fullname:20c78ca7
Time: 2833 ms
```

Спробуємо знайти користувача за ідентифікатором без використання індексів:

```
On what entity do you want to perform actions?('user' / 'message' / 'folder') or 'exit'
user
What do you want to do with the user table?('add' / 'delete' / 'edit' / 'find' / 'findById' / 'generate')
findById
Enter the ID of the user you want to find
19544
User: Id:19544 | Username:GDT | Fullname:0a0f55ad
Time: 45849 ms
```

```
On what entity do you want to perform actions?('user' / 'message' / 'folder') or 'exit'
user
What do you want to do with the user table?('add' / 'delete' / 'edit' / 'find' / 'findById' / 'generate')
findById
Enter the ID of the user you want to find
19544
User: Id:19544 | Username:GDT | Fullname:0a0f55ad
Time: 45849 ms
```

Спробуємо знайти користувача за ідентифікатором використовуючи індекс

Hash:

```
On what entity do you want to perform actions?('user' / 'message' / 'folder')
or 'exit'
user
What do you want to do with the user table?('add' / 'delete' / 'edit' /
'find' / 'findById' / 'generate')
findById
Enter the ID of the user you want to find
19544
User: Id:19544 | Username:GDT | Fullname:0a0f55ad
Time: 11722 ms
```

```
On what entity do you want to perform actions?('user' / 'message' / 'folder') or 'exit'
user
What do you want to do with the user table?('add' / 'delete' / 'edit' / 'find' / 'findById' / 'generate')
findById
Enter the ID of the user you want to find
19544
User: Id:19544 | Username:GDT | Fullname:0a0f55ad
Time: 11722 ms
```

Спробуємо відфільтрувати папки за назвою та користувацьким ім'ям :

```
On what entity do you want to perform actions?('user' / 'message' / 'folder') or 'exit'
folder
What do you want to do with the folder table?('add' / 'delete' / 'edit' / 'find' / 'generate')
find
Enter the user_name of the user to whom the folder belongs
BK
Enter the name of the folder you want to find
ca
```

Без використання індексу:

```
FOLDER: Id : 18978 | Folder name: BUPMQaca | User id: 17935 BELONGS TO THE USER: Id:17935 | Username:BKQ
Time: 52830 ms
```

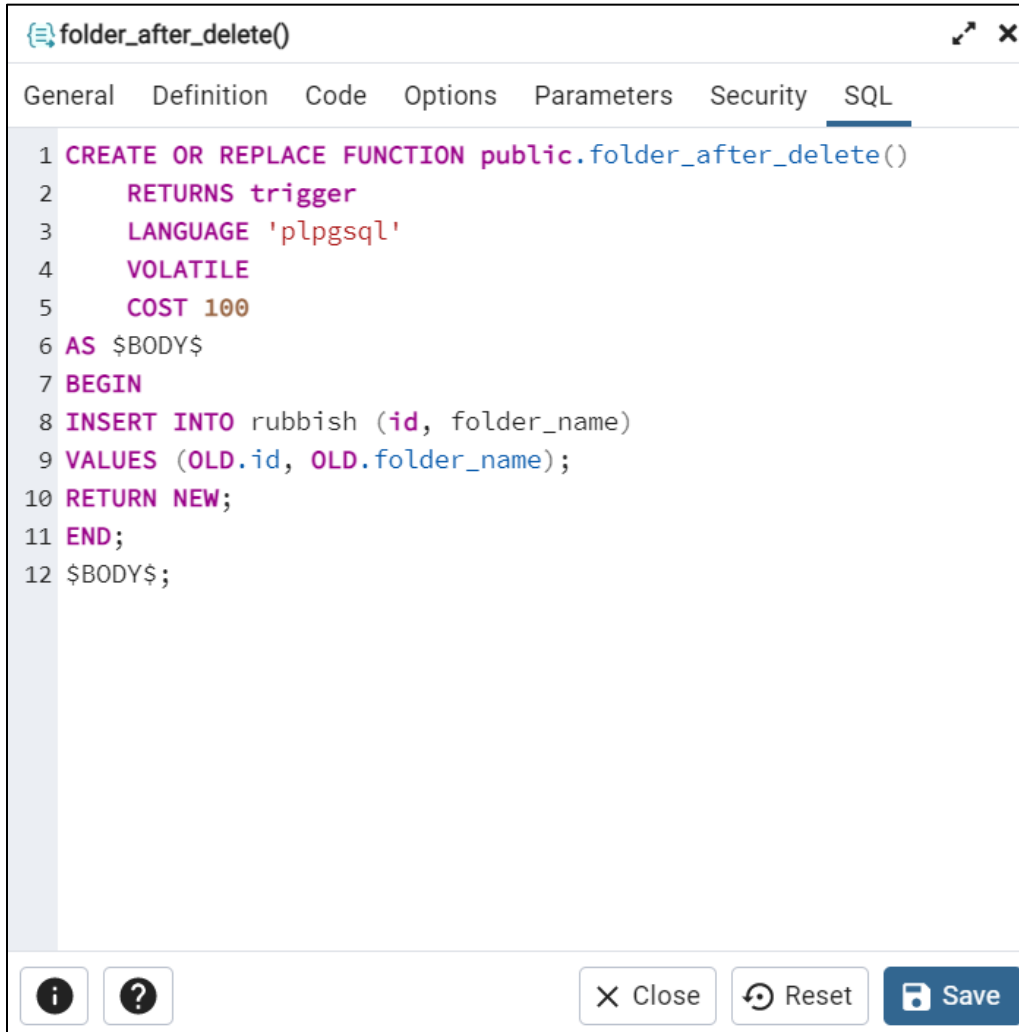
З використанням індексу виду GIN:

```
FOLDER: Id : 18978 | Folder name: BUPMQaca | User id: 17935 BELONGS TO THE USER: Id:17935 | Username:BKQ | Fullname:05c683e5
Time: 29776 ms
```

Завдання 3:

Для того щоб використати тригер after delete створимо ще одну таблицю (rubbish), в неї будемо записувати папки, які видаляються.

Тригерна функція folder_after_delete():



```
1 CREATE OR REPLACE FUNCTION public.folder_after_delete()
2     RETURNS trigger
3     LANGUAGE 'plpgsql'
4     VOLATILE
5     COST 100
6 AS $BODY$
7 BEGIN
8 INSERT INTO rubbish (id, folder_name)
9 VALUES (OLD.id, OLD.folder_name);
10 RETURN NEW;
11 END;
12 $BODY$;
```

Спробуємо видалити папку з id '6':

Таблиця folders до видалення:

	id [PK] integer	folder_name text	Userid integer
1	6	jLTZlgOZ	5277
2	7	lKmbGPnl	16549
3	8	mNgWrXCb	10542
4	9	uTufmBfQ	1877
5	10	pqbckYDU	16150
6	11	EYirAgjZ	5459
7	12	ldrVmGtZ	17279
8	13	JjSfvjqp	10652
9	14	bmkCfsZi	11637
10	15	ECvuMKNr	15310
11	16	vYCuQHce	19027
12	17	vVUulYJD	19679

Таблиця rubbish до видалення:

	id [PK] integer	folder_name text

```
On what entity do you want to perform actions?('user' / 'message' / 'folder')
or 'exit'
folder
What do you want to do with the folder table?('add' / 'delete' / 'edit' /
'find' / 'generate')
delete
Enter the ID of the folder you want to delete:
6
Folder has been removed from the database
```

Таблиця folders після видалення:

	id [PK] integer	folder_name text	Userid integer
1	7	lKmbGPnl	16549
2	8	mNgWrXCcb	10542
3	9	uTufmBfQ	1877
4	10	pqbckYDU	16150
5	11	EYirAgjZ	5459
6	12	ldrVmGtZ	17279
7	13	JjSfvjqp	10652
8	14	bmkCfsZi	11637
9	15	ECvuMKNr	15310
10	16	vYCuQHce	19027

Таблиця rubbish після видалення:

Data Output		Explain	Mess
	id [PK] integer	folder_name text	
1	6	jLTZlgOZ	

Висновки

ORM — технологія програмування, яка зв'язує бази даних з концепціями об'єктно-орієнтованих мов програмування, створюючи «віртуальну об'єктну базу даних».

ORM створюють модель об'єктно-орієнтованої програми з високим рівнем абстракції. Іншими словами, це створює рівень логіки без основних деталей коду. Відображення описує зв'язок між об'єктом і даними, не знаючи, як дані структуровані. Потім модель можна використовувати для з'єднання програми з кодом SQL, необхідним для керування даними. Цей «сантехнічний» тип коду не потрібно переписувати, що економить розробнику величезну кількість часу.

ORM дозволяє абстрагуватися з бази даних і значно спростити необхідний вихідний код. Окрім простоти та швидкості використання, він забезпечує захист рівня доступу до даних від атак.

PostgreSQL підтримує кілька типів індексів: B-tree, hash, GiST, SP-GiST, GIN, BRIN. Для різних типів індексів застосовуються різні алгоритми, орієнтовані певні типи запитів.

B-tree

Цей тип індексу використовується за замовчуванням і покриває дуже широке коло завдань (бази даних більшості додатків можуть успішно обходитися тільки індексами на основі B-tree).

За допомогою B-tree можна проіндексувати будь-які дані, які можуть бути відсортовані, тобто для яких застосовні операції порівняння більше/менш/рівно. Сюди можна віднести числа, рядки, дати та час, логічний тип та будь-які дані, які можна ними закодувати.

GiST та SP-GiST

GiST - скорочення від "generalized search tree". Це збалансоване дерево пошуку, так само, як і розглянуте раніше b-tree. Він дозволяє розподілити дані будь-якого типу по збалансованому дереву і використовувати це дерево для пошуку за різними умовами. Якщо при побудові B-tree ми сортуємо всі об'єкти і ділимо його на частини за принципом більш-менш, при побудові GiST індексів можна реалізувати будь-який принцип розбиття будь-якої безлічі об'єктів.

SP-GiST схожий на GiST, але він дозволяє створювати незбалансовані дерева. Такі дерева можуть бути корисні при розбитті безлічі на об'єкти, що не перетинаються.

Крім того, GiST і SP-GiST можуть бути своєрідним фреймворком, що полегшує розширення PostgreSQL і додавання до нього абсолютно нових видів дерев для індексації нових типів даних.

GIN розшифровується як Generalized Inverted Index це так званий зворотний індекс. Він працює з типами даних, значення яких не є атомарними, а складаються з елементів. У цьому індексуються не самі значення, а окремі елементи; кожен елемент посилається ті значення, у яких зустрічається. Елементи ніколи не видаляються з GIN-індексу. Вважається, що значення, що містять елементи, можуть зникати, з'являтися, змінюватися, але набір елементів, з яких вони складаються досить статичний. Таке рішення істотно

спрощує алгоритми, що забезпечують паралельну роботу з індексом кількох процесів.

Ідея **BRIN** не в тому, щоб швидко знайти потрібні рядки, а в тому, щоб уникнути перегляду непотрібних. BRIN добре працює для тих стовпців, значення в яких корелюють з їх фізичним розташуванням у таблиці.