

Засоби оптимізації роботи СУБД PostgreSQL

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

Завдання роботи полягає у наступному:

1. Перетворити модуль “Модель” з шаблону MVC лабораторної роботи №2 у вигляд об’єктно-реляційної проекції (ORM).
2. Створити та проаналізувати різні типи індексів у PostgreSQL.
3. Розробити тригер бази даних PostgreSQL.
4. Навести приклади та проаналізувати рівні ізоляції транзакцій у PostgreSQL.

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:М, М:М та 1:1 виконати опис схеми бази даних.

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 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 №2, створивши необхідну кількість тестових даних. Навести 4-5 прикладів запитів SELECT, що

Д и с ц и п л і н и « Б а з и д а н и х т а з а с о б и у п р а в л і н н я » т а
« Б а з и д а н и х . Ч а с т и н а 1 ». О с і н ь 2019 р о к у
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 пункту завдання №4

Проаналізувати на прикладах використання рівнів ізоляції транзакцій READ COMMITTED, REPEATABLE READ та SERIALIZABLE, продемонструвавши феномени, які виникають, і спосіб їх уникнення завдяки встановленню відповідного рівня ізоляції транзакцій. Для виконання завдання необхідно відкрити дві транзакції у різних вікнах pgAdmin4 і виконати послідовність запитів INSERT, UPDATE або DELETE у обох транзакціях, що доводять наявність або відсутність певних феноменів.

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 10 або 11.

Вимоги до оформлення лабораторної роботи у електронному вигляді

Опис та вміст репозиторію лабораторної роботи у **репозиторії GitHub** включає: назву лабораторної роботи, варіант студента, а також:

- для завдання №1: оновлений програмний код у файлах репозиторію;
- для завдання №2: команди створення індексів, тексти і час виконання запитів SQL;

Д и с ц и п л і н и « Б а з и д а н и х т а з а с о б и у п р а в л і н н я » т а
« Б а з и д а н и х . Ч а с т и н а 1 ». О с і н ь 2019 р о к у

- для завдання №3: команди, що ініціюють виконання тригера, текст тригера та скріншоти зі змінами у таблицях бази даних;
- для завдання №4: скріншоти з ходом виконання запитів та їх результатів у обох транзакціях по кожному рівню ізоляції.

Вибір варіанту

Робота виконується індивідуально. Варіант обирається шляхом взяття останніх двох цифр номеру залікової книжки студента.

Варіанти

<i>№ варіанта</i>	<i>Види індексів</i>	<i>Умови для тригера</i>
<i>1</i>	<i>Btree, Hash</i>	<i>before insert, delete</i>
<i>2</i>	<i>Hash, BRIN</i>	<i>after insert, update</i>
<i>3</i>	<i>GIN, Hash</i>	<i>before delete, update</i>
<i>4</i>	<i>GIN, BRIN</i>	<i>after delete, insert</i>
<i>5</i>	<i>BTree, GIN</i>	<i>before update, delete</i>
<i>6</i>	<i>BTree, BRIN</i>	<i>after update, insert</i>
<i>7</i>	<i>GIN, BRIN</i>	<i>before insert, delete</i>
<i>8</i>	<i>BTree, GIN</i>	<i>after insert, update</i>
<i>9</i>	<i>BTree, BRIN</i>	<i>before delete, update</i>
<i>10</i>	<i>Hash, BRIN</i>	<i>after delete, insert</i>
<i>11</i>	<i>GIN, Hash</i>	<i>before update, delete</i>
<i>12</i>	<i>BTree, GIN</i>	<i>after update, insert</i>
<i>13</i>	<i>BTree, BRIN</i>	<i>before insert, delete</i>
<i>14</i>	<i>Btree, Hash</i>	<i>after insert, update</i>
<i>15</i>	<i>Hash, BRIN</i>	<i>before delete, update</i>

16	<i>GIN, Hash</i>	<i>after delete, insert</i>
17	<i>GIN, BRIN</i>	<i>before update, delete</i>
18	<i>BTree, GIN</i>	<i>after update, insert</i>
19	<i>BTree, BRIN</i>	<i>before insert, delete</i>
20	<i>GIN, BRIN</i>	<i>after insert, update</i>
21	<i>Btree, Hash</i>	<i>before delete, update</i>
22	<i>Hash, BRIN</i>	<i>after delete, insert</i>
23	<i>GIN, Hash</i>	<i>before update, delete</i>
24	<i>GIN, BRIN</i>	<i>after update, insert</i>
25	<i>BTree, GIN</i>	<i>after delete, insert</i>
26	<i>BTree, BRIN</i>	<i>before update, delete</i>
27	<i>GIN, BRIN</i>	<i>after update, insert</i>

Контрольні запитання

1. Сформулювати призначення та задачі об'єктно-реляційної проєкції (ORM).
2. Назвати основні види індексів у PostgreSQL.
3. Поясніть призначення тригерів та функцій у базах даних.