**Лабораторна робота № 3**

**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.

*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. 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.

Корисні посилання: [тут](https://www.learndatasci.com/tutorials/using-databases-python-postgres-sqlalchemy-and-alembic/) і [тут](https://auth0.com/blog/sqlalchemy-orm-tutorial-for-python-developers/).

*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 (із виведенням результуючих даних), що містять фільтрацію, агрегатні функції, групування та сортування (у необхідних комбінаціях).

Корисні посилання: [Hash](https://habrahabr.ru/company/postgrespro/blog/328280/), [B-tree](https://habrahabr.ru/company/postgrespro/blog/330544/), [GIN](https://habrahabr.ru/company/postgrespro/blog/340978/), [BRIN](https://habrahabr.ru/company/postgrespro/blog/346460/).

*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 лагодження тригера при різних вхідних даних та навести 2‑3 приклади його використання.

Корисні посилання: [тут](https://www.enterprisedb.com/postgres-tutorials/everything-you-need-know-about-postgresql-triggers), [тут](https://www.postgresqltutorial.com/postgresql-triggers/).

*Вимоги до інструментарію*

1. Бібліотека для реалізації ORM - SQLAlchemy для Python або інша з подібною функціональністю;
2. Середовище для лагодження 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:

1. титульний аркуш затвердженого зразка;
2. завдання на лабораторну роботу з обов’язковим наведенням варіанту студента;
3. 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 і зв’язків між ними, 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.

*Вибір варіанту*

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

*Варіанти*

|  |  |  |
| --- | --- | --- |
| *№ варіанта* | *Види індексів* | *Умови для тригера* |
| *1* | *Btree, Hash* | *before insert, delete* |
| *2* | *Hash, BRIN* | *after insert, update* |
| *3* | *GIN, Hash* | *before delete, update* |
| *4* | *GIN, BRIN* | *after delete, insert* |
| *5* | *BTree, GIN* | *before update, delete* |
| *6* | *BTree, BRIN* | *after update, insert* |
| *7* | *GIN, BRIN* | *before insert, delete* |
| *8* | *BTree, GIN* | *after insert, update* |
| *9* | *BTree, BRIN* | *before delete, update* |
| *10* | *Hash, BRIN* | *after delete, insert* |
| *11* | *GIN, Hash* | *before update, delete* |
| *12* | *BTree, GIN* | *after update, insert* |
| *13* | *BTree, BRIN* | *before insert, delete* |
| *14* | *Btree, Hash* | *after insert, update* |
| *15* | *Hash, BRIN* | *before delete, update* |
| *16* | *GIN, Hash* | *after delete, insert* |
| *17* | *GIN, BRIN* | *before update, delete* |
| *18* | *BTree, GIN* | *after update, insert* |
| *19* | *BTree, BRIN* | *before insert, delete* |
| *20* | *GIN, BRIN* | *after insert, update* |
| *21* | *Btree, Hash* | *before delete, update* |
| *22* | *Hash, BRIN* | *after delete, insert* |
| *23* | *GIN, Hash* | *before update, delete* |
| *24* | *GIN, BRIN* | *after update, insert* |
| *25* | *BTree, GIN* | *after delete, insert* |
| *26* | *BTree, BRIN* | *before update, delete* |
| *27* | *GIN, BRIN* | *after update, insert* |

*Контрольні запитання*

1. Сформулювати призначення та задачі об’єктно-реляційної проекції (ORM).
2. Проаналізувати основні види індексів у PostgreSQL (*BTree, BRIN, GIN, Hash*): призначення, сфера застосування, переваги та недоліки.
3. Пояснити призначення тригерів та функцій у базах даних.

*Критерії оцінювання*

1. Виконання пунктів 1-3 завдання (0-4 балів). Максимум: 3 х 4=12 балів.
2. Наявність збережених файлів у репозиторії Git (0-1 бал).
3. Якість звіту (0-2 балів).
4. Додаткові бали: до 19.11.2020 (+2), до 26.10.2020 (+1).
5. Штрафні бали: (-1) за кожен тиждень після 01.12.2020.

Всього за лабораторну роботу нараховується: 15 балів.