

Лабораторна робота № 3.

Розробка об'єктно-реляційної проекції Web-орієнтованої інформаційно-пошукової системи

Метою роботи є здобуття практичних навичок використання об'єктно-реляційної проекції (ORM) та дослідження серверних функціональних можливостей СУБД MySQL.

Основою розробки є програмні засоби, розроблені у лабораторній роботі №1.

Завдання роботи полягає у наступному:

1. Переробити модуль взаємодії з базою даних у об'єктно-реляційну проекцію ORM, вбудовану в Django.
2. Створити тригер, серверну процедуру (*stored procedure*) та подію (*scheduled event*) згідно з варіантом.
3. Проаналізувати різні рівні ізоляції транзакцій в MySQL.

Функціональні вимоги

1. Для усіх сутностей створити класи-моделі Django ORM; переписати функції, що використовують SQL-запити, у відповідні виклики ORM.
2. Забезпечити наявність функції увімкнення/вимкнення тригера.
3. Забезпечити наявність функції налаштування часу спрацювання події згідно з варіантом.
4. Зміст операцій у тригері, серверній процедурі та події обрати самостійно.
5. Підготувати приклади для ілюстрації роботи тригера, події та рівнів ізоляції транзакцій.

Вимоги до інтерфейсу користувача

1. За основу взяти розроблений в лабораторній роботі №2 інтерфейс користувача.
2. Забезпечити інтерфейс користувача елементами щодо увімкнення/вимкнення тригера та налаштування часових параметрів події.
3. Приклади використання рівнів ізоляції транзакцій проілюструвати в консолі MySQL.

Вимоги до інструментарію

1. Середовище для відлагодження SQL-запитів до бази даних – MySQL Workbench.

2. Мова програмування – Python 2.7
3. Середовище розробки програмного забезпечення – PyCharm Community Edition 2016
4. Web-фреймворк – Django 1.8
5. Доступ до MySQL – вбудовані засоби Django ORM.

Вибір варіанту

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

Варіанти

№ вар.	Часові параметри події	Рівні ізоляції транзакції
1.	Щотижня	REPEATABLE READ, READ COMMITTED
2.	Щодня	READ UNCOMMITTED, READ COMMITTED
3.	Щомісяця	SERIALIZABLE, REPEATABLE READ
4.	Один раз	READ UNCOMMITTED, READ COMMITTED
5.	Щотижня	SERIALIZABLE, REPEATABLE READ
6.	Щодня	REPEATABLE READ, READ COMMITTED
7.	Щомісяця	REPEATABLE READ, READ COMMITTED
8.	Один раз	READ UNCOMMITTED, READ COMMITTED
9.	Щотижня	SERIALIZABLE, REPEATABLE READ
10.	Щодня	READ UNCOMMITTED, READ COMMITTED
11.	Щомісяця	SERIALIZABLE, REPEATABLE READ
12.	Один раз	REPEATABLE READ, READ COMMITTED
13.	Щотижня	REPEATABLE READ, READ COMMITTED
14.	Щодня	READ UNCOMMITTED, READ COMMITTED
15.	Щомісяця	SERIALIZABLE, REPEATABLE READ
16.	Один раз	READ UNCOMMITTED,

		READ COMMITTED
17.	Щотижня	SERIALIZABLE, REPEATABLE READ
18.	Щодня	REPEATABLE READ, READ COMMITTED
19.	Щомісяця	REPEATABLE READ, READ COMMITTED
20.	Один раз	READ UNCOMMITTED, READ COMMITTED
21.	Щотижня	SERIALIZABLE, REPEATABLE READ
22.	Щодня	READ UNCOMMITTED, READ COMMITTED
23.	Щомісяця	SERIALIZABLE, REPEATABLE READ
24.	Один раз	REPEATABLE READ, READ COMMITTED
25.	Щотижня	SERIALIZABLE, READ COMMITTED
26.	Щодня	READ UNCOMMITTED, REPEATABLE READ
27.	Щомісяця	READ UNCOMMITTED, READ COMMITTED
28.	Один раз	SERIALIZABLE, READ COMMITTED

Вимоги до оформлення лабораторної роботи

Протокол лабораторної роботи має містити: варіант студента, список класів Django ORM, тексти для створення тригера, події та серверної процедури, 2-3 копії екранних форм (screenshots). Протокол оформити у вигляді README-файлу та розмістити разом із вихідним кодом у власному репозиторії на сайті github.com.

Контрольні запитання

1. Дати визначення об'єктній реляційній проекції (ORM).
2. Проаналізувати поняття транзакції.
3. Дати визначення рівням ізоляції транзакцій в MySQL.
4. Сформулювати призначення та види тригерів.
5. Перерахувати можливі налаштування події в MySQL.