УНИВЕРСИТЕТ ИТМО

Факультет программной инженерии и компьютерной техники

Направление подготовки 09.03.04 Программная инженерия

Дисциплина «Информационный системы и базы данных»

**Курсовая работа**

Этап 3

Студент

*Митрофанов Е. Ю.*

*Любкин А. С.*

*P33101*

Преподаватель

*Николаев В. В.*

*Гаврилов А. В.*

Санкт-Петербург, 2021 г.

Реализовать даталогическую модель в реляционной СУБД PostgreSQL:

* Создать необходимые объекты базы данных
* Заполнить созданные таблицы тестовыми данными
* Сделать скрипты для:

создания/удаления объектов базы данных

заполнения/удаления созданных таблиц

* Обеспечить целостность данных при помощи средств языка DDL.
* Добавить в базу данных триггеры для обеспечения комплексных ограничений целостности
* Реализовать функции и процедуры на основе описания бизнес-процессов (из этапа №1)
* Произвести анализ использования созданной базы данных:

выявить наиболее часто используемые запросы к объектам базы данных

результаты представить в виде текстового описания

* Создать индексы и доказать, что они полезны для вашей базы данных:

доказательство должно быть приведено в виде текстового описания

Скрипты запросов

<https://github.com/EgorMIt/DB_Cursework>



Триггеры

Все скрипты в файле *triggers.sql*

«ПроверкаМатериала» *- проверяет, достаточно ли материала на складе для постройки здания*

«ДублироватьУлицу» - *добавление обратной связи при указании факта пересечении улиц*

«ДублироватьКвартал» - *добавление обратной связи при указании факта соседства кварталов*

«ПроверитьСлуюбу» - *проверка на доступность городской службы в районе постройки здания*

Функции

Все скрипты в файле *functions.sql*

СравнитьКоличество - *Сравнивает количество нужного для здания стройматериала с количеством на складе*

ДублироватьРеверсКвартал - *Добавляет обратную связь для квартала*

ДублироватьРеверсУлицу - *Добавляет обратную связь для улицы*

ПолучитьКварталИзЗдания - *Возвращает квартал, в котором находится нужное здание*

ПроверитьНаличиеСлужбыВКварталеЗдания - *Проверяет, есть ли служба в квартале здания*

ПодсчитатьСтоимостьЗдания - *Считает стоимость материалов для здания и подключения служб*

ПринятьГотовыеЗданияДляКомитета - *Выбранный комитет принимает все свои готовые здания*

ПодсчитатьДолюГотовностиУлицы - *Считает, на сколько % готова улица*

ПодсчитатьДолюГотовностиКвартала - *Считает, на сколько % готов квартал*

ПодсчитатьДолюГотовностиГорода - *Считает, на сколько % готов весь город*

Анализ

Наиболее часто используемая информация хранится в таблице *«Здание»,* т. к. именно она характеризует элемент городской инфраструктуры. В результате имеет смысл оптимизировать запросы к данной таблице. Однако при реальном мониторинге нас очень редко интересует информация об одном конкретном здании, обычно мы собираем информацию по определенному кварталу или улице, разработкой которого/которой мы в данный момент занимаемся. Поэтому мы приняли решение добавить индексирование по полю *«Улица\_id»,* а также проиндексировать таблицу *«Улица»* по полю *«Квартал\_id»,* для оптимизации доступа к кварталу по улице. Идентификатор нас редко интересует в каком-то диапазоне, обычно мы хотим получить конкретное значение, поэтому оптимально будет использовать hash индексирование.

Индексы

Здание\_idx – *hash индекс для "Улица\_id" в "Здание"*

Улица\_idx – *hash индекс для "Улица1\_id", "Улица2\_id" в "Улица\_Улица"*

Квартал\_idx - *hash индекс для "Квартал1\_id","Квартал2\_id" в "Квартал\_Квартал"*

Служба\_Команды\_idx - *hash индекс для "Служба\_id" в "Обслуживающая\_команда"*

Квартал\_Команды\_idx - *hash индекс для "Квартал\_id" в "Обслуживающая\_команда"*

Цена\_Материала\_idx - *B-tree индекс для "Цена" в "Стройматериал"*

Выводы

Во время выполнения лабораторной работы мы изучили работу функций, процедур и триггеров, которые их вызывают, для реализации ограничения целостности. Использовали индексы для ускорения обработки запросов в будущем приложении и проанализировали базу данных.