

Факультет программной инженерии и компьютерной техники

*Информационные системы и базы данных*

Курсовая работа. Этап 3

|  | **Работу выполнили**  Студенты группы P33101  Ольга Ильинская  Сыюань Хуан  **Преподаватель**  Гаврилов Антон Валерьевич |
| --- | --- |

Санкт-Петербург

2021

1. Задание

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  
  No1).
* 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.

1. Создание таблиц

<https://github.com/cantansweratthemoment/course-work/blob/main/ScriptCreate.sql>

1. Наполнение таблиц данными

<https://github.com/cantansweratthemoment/course-work/blob/main/ScriptFill.sql>

1. Удаление таблиц

<https://github.com/cantansweratthemoment/course-work/blob/main/ScriptDelete.sql>

1. Триггеры, функции, процедуры

<https://github.com/cantansweratthemoment/course-work/tree/main/Triggers%20and%20Functions>

1. Индексы

<https://github.com/cantansweratthemoment/course-work/blob/main/Indexes.sql>

Обычные запросы последовательно идут по строкам в таблице, то есть асимптотика поиска без индексов - O(n). Однако, вставка происходит достаточно просто, так как ничего не надо перестраивать, все идёт последовательно. Асимптотика вставки - О(1).

Благодаря индексам мы можем представлять данные в виде сбалансированного дерева. Как мы знаем, поиск по сбалансированному дереву быстрее - это глубина дерева. То есть О(log N). Однако вставка - тоже O(log N), так как мы не можем просто вставить новую строку в конец, нам нужно перестраивать сбалансированное дерево.

Поэтому принцип выбора индекса следующий:

Индексы не нужно использовать на небольших таблицах.

Не нужно использовать индексы на часто обновляемых таблицах.

Не стоит использовать индексы на тех атрибутах, которые часто принимают NULL значение.