**МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РФ**

**Федеральное государственное бюджетное образовательное учреждение высшего образования**

**«Московский Авиационный Институт»**

**(Национальный Исследовательский Университет)**

**Институт: №8 «Информационные технологии   
и прикладная математика»   
Кафедра: 806 «Вычислительная математика   
и программирование»**

Лабораторная работа № 2  
по курсу «Базы дынных»

Группа: М8О-310Б-21

Студент(ка): И. В. Сектименко

Преподаватель: Н. Пономарев

Оценка:

Дата: 18.04.2025

Москва, 2025

Тема

Работа с Redis.

Цель работы

Оптимизировать работу курсового проекта за счёт внедрения Redis для работы с динамическими данными.

Задачи

Во время выполнения лабораторной работы необходимо сделать следующие задачи:

1. выделить данные, требующие быстрого доступа, временного хранения или частого обновления, включая:

* токены авторизации пользователей;
* сессионные данные;
* часто запрашиваемые или временные данные;
* данные, требующие мгновенного обновления (статусы заказов, уведомления);

1. для токенов авторизации:

* реализовать хранение в Redis с настройкой TTL (время жизни) через конфигурацию приложения;
* обеспечить автоматическое удаление просроченных токенов средствами Redis;

1. для остальных динамических данных реализовать оптимальный формат хранения в Redis, обосновав выбор структуры для конкретных сценариев использования;
2. настроить мгновенные оповещения о критичных событиях (например, новые заказы, изменения статусов) через PubSub;
3. объяснить, как и почему должна измениться скорость работы системы после внедрения Redis.

Ход работ

Для начала необходимо было найти свою курсовую работу за прошлый семестр. Найти его можно по ссылке на гитхабе: <https://github.com/Lavanda-da/db_kp>. Там закоммитчены как старая версия с прошлого семестра, так и версия с изменениями, внесенными в ходе выполнения лабораторной работы.

Проанализировав логику своей программы, я поняла, что авторизация у меня происходит «неявно», то есть каждый раз для оформления поставки или заказа пользователи должны вводить свои регистрационные данные для дальнейшей проверки наличия их в системе. Как раз тут будет наиболее разумно применить Redis, который позволит ускорить проверку наличия пользователя в системе. Если пользователь первый раз создает заказ или поставку, то его данные ищутся в базе данных Postgres и после этого добавляются в Redis. При повторном заказе проверка пользователя будет ускорена, так как его данные уже будут храниться в Redis, то есть в оперативной (быстрой) памяти, а не на жестком диске. При большой базе данных пользователей это будет особенно заметно.

Так же по заданию необходимо было настроить TTL, то есть время «жизни» данных. Я решила, что 10 минут будет достаточно, ведь мое приложение достаточно маленькое и имеет ограниченный функционал, поэтому пользователь вряд ли будет задерживаться на этом сайте. В дальнейшем при необходимости TTL может быть увеличен.

Критичные события в моем проекте – создание пользователей и оформление заказов. Я решила, что создание пользователей не так интересно, как оформление заказов и поставок, ведь именно последнее приносит деньги и является наиболее важной частью моего проекта.

В Redis публикуются события о создании заказов и поставок. Это происходит, когда данные пользователя добавляются в Redis или достаются из него, ведь в текущей версии Redis хранится только такая информация. По логике моего проекта возможна отмена заказа пользователя, что не будет обработало Redis, так как обрабатывается триггером внутри Postgres, но это не сильная потеря, ведь лучше поставщик приготовит заказ по сообщению из подписчика, а потом поймет, что он был отменен, так как в базе данных будет статус об отмене, чем поставщик не узнает о заказе и потеряет как деньги, так и клиента.

Заключение

В ходе выполнения я познакомилась с Redis и внедрила его в свой проект. Это оказалось легче, чем я думала (возможно, из-за простоты написания кода на Python и хорошей интеграции Redis в Python).

Redis позволяет оптимизировать работу с данными и ускорить их получения, так как данные Redis хранятся в оперативной памяти, а в базах данных на жестком диске. Однако из-за небольшого размера оперативной памяти важно правильно определить, что часто используется, что будет разумно хранить в ней и сколько по времени.