

Федеральное государственное автономное
образовательное учреждение
высшего образования
«СИБИРСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

Институт космических и информационных технологий
институт

Кафедра «Информатика»
кафедра

ОТЧЕТ О ПРАКТИЧЕСКОЙ РАБОТЕ

Индексы
Тема

Преподаватель

подпись, дата

А. К. Погребников

инициалы, фамилия

Студент

КИ19-16/16 031939175

номер группы, зачетной
книжки

подпись, дата

А. Д. Непомнящий

инициалы, фамилия

Красноярск 2021

1 Цель работы

Цель работы состоит в получении навыков нагрузочного тестирования разработанной архитектуры БД, а также в оптимизации запросов и структуры данных.

2 Задачи

Выполнение работы сводится к следующим задачам.

1. Добавление не менее 500000 новых объектов таблиц сущностей.
2. Написание запроса, который может быть использован в разрабатываемой системе. Запрос должен включать в себя не менее трех таблиц, а также не менее трех условий в WHERE для фильтрации данных.
3. Добавление индекса на свойство, используемое в WHERE, и повторное выполнение запроса с записью значений производительности.
4. Добавление индекса на другие свойства, используемые в WHERE, и повторное выполнение запроса с записью значений производительности.
5. Установка для одного из индексов типа bitmap и повторное выполнение запроса с записью значений производительности и количества глобальных ссылок.
6. Добавление индексов на ссылочные поля, используемые для объединения в JOIN. Перестройка индексов и замер производительности.

3 Ход работы

3.1 Имеющаяся архитектура БД

За основу взята БД из предыдущей работы. На рисунке 1 далее приведена схема архитектуры БД.

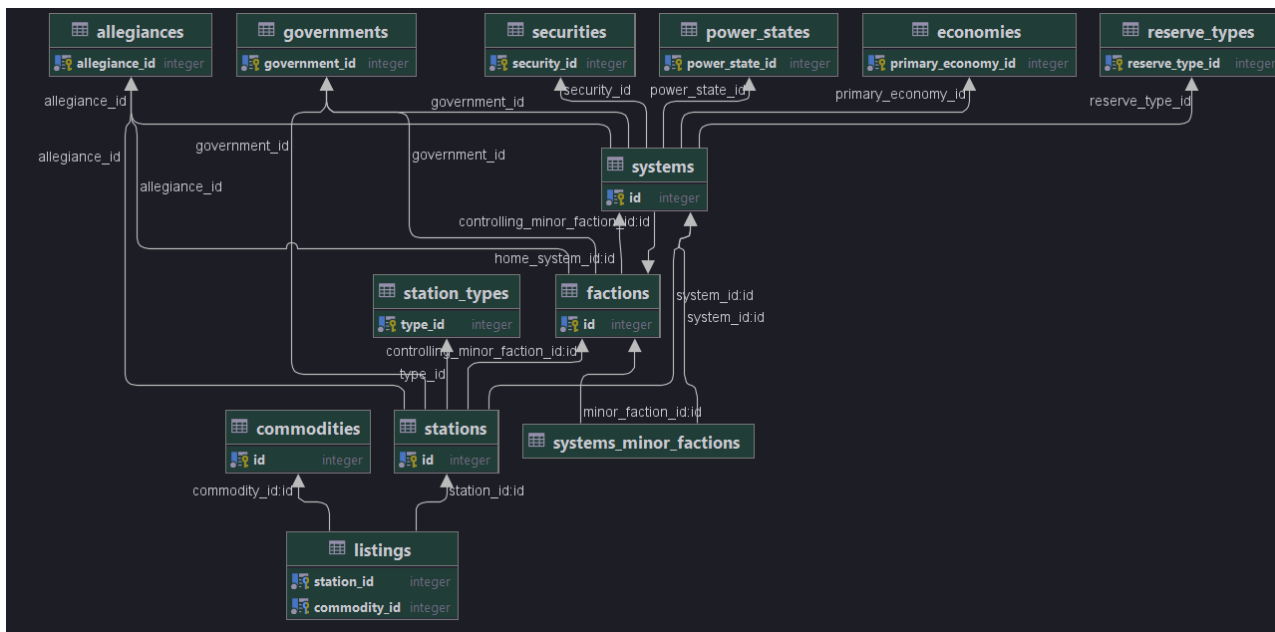


Рисунок 1 – Архитектура БД

3.2 Заполнение БД

БД была заполнена данными, полученными через API веб-сайта eddb.io. JSON-файлы с данными были преобразованы в CSV-файлы и вставлены в соответствующие таблицы.

В результате в БД содержится информация о 140095 станциях в 20585 обитаемых системах. Информация о товарах на рынках этих станций хранится в виде 6258132 записей. Далее приведены запросы числа записей в основных таблицах БД.

count(*)	count(*)	count(*)	count(*)
140095	20585	6258132	374

```

SELECT count(*)
FROM systems;

SELECT count(*)
FROM stations;

SELECT count(*)
FROM commodities;

SELECT count(*)
FROM listings
  
```

Рисунок 2 – Результат запросов SELECT count(*) к основным таблицам

3.3 Запрос к БД

Для проверки производительности был написан запрос в соответствии с заданием.

Командиру нужно продать товар “Platinum” по наиболее выгодной цене. При этом необходимо, чтобы площадка для посадки была минимум размера М, а также спрос на товар был не менее 10000. При этом он находится в розыске и будет пойман на станциях, принадлежащих “Alliance” и “Federation”, при попытке продажи там товаров, а значит – такие станции не рассматриваются.

Код запроса приведен в листинге 1.

Листинг 1 – Запрос к БД

```
SELECT s.name, c.name, l.sell_price
FROM listings l
      JOIN stations c ON c.id = l.station_id
      JOIN systems s ON c.system_id = s.id
WHERE l.commodity_id = (SELECT id FROM commodities WHERE name =
'Synthetic Fabrics')
      AND l.demand > 10000
      AND c.max_landing_pad_size <= 'M'
      AND c.allegiance_id NOT IN (SELECT a.allegiance_id FROM allegiances a
WHERE allegiance = 'Alliance' OR allegiance ='Federation')
ORDER BY sell_price DESC;
```

3.4 Результаты выполнения запроса

Данные о производительности были получены с помощью команды EXPLAIN с параметром ANALYZE. Полученные данные изложены в таблице 1.

Таблица 1 – Данные о производительности запроса

Индексация	Производительность, мс
Без индексов (кроме ключей)	6134.945
Создан listings_demand_commodity_id_index	1822.389
Создан stations_allegiance_id_index	1836.468

Дальнейшее создание индексов не привело к результатам, так как не удалось подобрать индексы, кроме listings_demand_commodity_id_index,

которые бы применялись в запросе. Это связано с селективностью выборок, выполняемых в ходе запроса – большой селективностью отличается только выборка по параметрам demand и commodity_id (см. рисунок 2).

Operation	Params	Rows	Actual Rows	Total Cost
⌵ ⌵ Select				
⌵ Sort		233	3532	9013.33
⌵ Full Scan (Seq Scan)	table: commodities;	1	1	8.68
⌵ Nested Loops (Nested Loop)		233	3532	8994.91
⌵ Hash Join		233	3532	8916.19
⌵ Full Scan (Seq Scan)	table: stations;	47971	66206	4145.52
⌵ Full Scan (Seq Scan)	table: allegiances;	2	2	1.09
⌵ Transformation (Hash)		681	5522	4636.24
⌵ Bitmap Index Scan (Bitmap Heap Scan)	table: listings;	681	5522	4636.24
⌵ Bitmap Index Scan	index: listings_demand_commodity_id_index;	681	5522	2185.95
⌵ Index Scan	table: systems; index: systems_id_uindex;	1	1	0.34

Рисунок 3 – План выполнения запроса

4 Вывод

В ходе работы был написан запрос в соответствии с заданием и протестирована производительность при наличии разных индексов. В конечном итоге запрос был оптимизирован одним индексом.