

**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
“КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
ІМЕНІ ІГОРЯ СІКОРСЬКОГО”**

Факультет прикладної математики
Кафедра програмного забезпечення комп’ютерних систем

КУРСОВА РОБОТА

з дисципліни “Бази даних”

спеціальність 121 – Програмна інженерія

на тему: **Система аналізу цін споживчих товарів**

Студент
групи КП-92

Колеснік Антон
Сергійович

(підпис)

Викладач
к.т.н, доцент кафедри
СПіСКС

Петрашенко А.В.

(підпис)

Захищено з оцінкою _____

Київ – 2020

Анотація

Метою розробки даного курсового проекту є набуття практичних навичок розробки сучасного програмного забезпечення, що взаємодіє з реляційними базами даних та навичок оформлення відповідного текстового, програмного та ілюстративного матеріалу у формі проектної документації.

У результаті виконання курсового проекту мною було опановано розроблення програмного забезпечення для реляційних баз даних, володіння основами використання СУБД, засобами аналізу великих обсягів даних, генерації та отримання даних з мережі Інтернет, індексації таблиць бази даних та роботи з ними.

Темою даного курсового проекту є система аналізу цін споживчих товарів.

Галуззю застосування даної розробки є біржа ігрових цінностей.

Результатами даного проекту став програмний додаток для роботи з, наближеною до реальної, базою даних біржі, оновлення та підтримання даної БД актуальною інформацією та прискорення швидкодії отримання даних з неї.

Зміст

1. Вступ.
2. Аналіз інструментарію для виконання курсової роботи.
3. Структура бази даних.
4. Опис програмного забезпечення.
5. Аналіз функціонування засобів реплікації.
6. Аналіз функціонування засобів резервування/відновлення.
7. Аналіз результатів підвищення швидкодії запитів.
8. Висновки.
9. Література.
10. Додатки

Вступ

Для продавців, які викладають свої послуги на біржі ігрових цінностей дуже важливо завжди бути конкурентно спроможним з іншими, і для цього йому потрібно аналізувати ціни конкурентів, їх відгуки і оцінки. З кожним днем ціни і конкуренція змінюється.

Для вирішення проблеми витрати часу на моніторинг та аналіз всієї біржі і продавця самому, який можна витратити на поліпшення якості надання послуг, був створений додаток, який вирішує проблеми, яка краде велику кількість часу.

Галуззю застосування даної розробки є біржа ігрових цінностей.

Метою розробки даного курсового проекту є набуття навичок розробки сучасного програмного забезпечення, що взаємодіє з реляційними базами даних, володіння основами використання СУБД, засобами аналізу великих обсягів даних, генерації та отримання даних з мережі Інтернет, індексації таблиць бази даних, роботи з ними, реплікації даних засобами СУБД.

У результаті виконання курсового проекту отримано програмне забезпечення для реляційних баз даних.

Дані для аналізу були взяті з скрапінгу онлайн сервісу FunPay, що забезпечує актуальність та наближеність даних до реальної ситуації.

Аналіз інструментарію для виконання курсової роботи

Було використано наступні бібліотеки та фреймворки:

- `cheerio` — надає підтримку великих багатовимірних масивів і матриць та операцій над ними. Використовується для обробки даних для візуалізації даних. [2]
- `express` — бібліотека Python 2D, яка представляє числові дані у різноманітних форматах та інтерактивних середовищах на різних платформах. Також ця бібліотека - математичне розширення NumPy. Засоби даної бібліотеки дозволили легко побудувати потрібні діаграми та графіки для візуального представлення. [3]
- `google charts` — бібліотека для обробки та аналізу даних. Працює разом з NumPy та Matplotlib для легкої та швидкої обробки даних для візуалізації даних. [4]
- `pg` — бібліотека для роботи з СУБД PostgreSQL. Потрібна для роботи ORM sqlalchemy. [5]

Було обрано PostgreSQL, як найбільш відповідну СУБД для даного проекту. Ця СУБД є масштабованою і здатна обробляти терабайти даних, має підтримку формату JSON, має безліч визначених функцій і можливостей індексації та є реляційною, що дуже важливо для складної системи зв'язків сутностей таблиць БД інтернет-магазину.

Також, слід зазначити, що PostgreSQL є безкоштовною СУБД, з відкритим кодом.

Опис програмного забезпечення

Скріншот структури файлів у додатку.

Опис:

- **addons folder**
 - print, enter, input відповідають за вивід та ввід у консоль.
 - comman, link додаток до роутера.
 - parse відповідає за парсинг даних з бд у відповідну сутність.
- **database**

Відповідає за працю з postgresql та зв'язок з серверами.
- **logic**

В цій папці логика вся логіка праці з MVC моделлю.
- **server**

В цій папці механізм створення сервера для підключення фронтенду.
- **tables**

В цій папці класи всіх сутностей бд.

Аналіз функціонування засобів реплікації

При будь яких змінах, які відбуваються з даними, на головному сервері (master) ці ж дані автоматично змінюється на підрядному сервері (slave). Дублювання на підрядний сервер відбувається при оновленні, видаленні або генерації нових даних на головному сервері.

Але використати такі ж самі дії на підрядному сервері не дозволено. Такі дані можливо лише зчитувати та використовувати для пошуку. Підрядний сервер починає приймати запити лише коли головний сервер зробився недієздатним.

Опис результатів аналізу предметної галузі

Дані для аналізування були взяті з бази даних за допомогою postgresql запитів та ілюстрована в різних видах за допомогою google charts. Також є один графік, який оновлюється в реальному часі і показуємо приріст продавців та покупців, який росте з кожним роком.

1. Аналіз популярності послуг серед фракцій.
2. Графік, який оновлюється в реальному часі, показує приріст продавців та продавців з кожним роком.
3. Діаграма, яка демонструє різницю кількості зареєстрованих користувачів (продавців і покупців) кожен рік.
4. Аналіз різницю середніх цін на різну складність послуг та їх кількість.
5. Аналіз цін на різну складність послуг для різних фракцій.

Приклади наведені в Додатку.

Аналіз результатів підвищення швидкодії виконання запитів

Для прискорення роботи з даними послуг, продавців і відгуків були використані індекси видів hash і btree. Ефективність прискорення запитів відображена в Додатках.

CREATE INDEX index_products_cost ON products USING btree (cost)
CREATE INDEX index_products_seller ON products USING btree(seller_id)
CREATE INDEX index_sellers_id ON sellers USING btree(id)
CREATE INDEX index_sellers_name ON sellers USING btree(name)

Висновок

В процесі виконання даного курсового проекту було отримано практичні навички роботи з великими масивами даних за допомогою мови програмування NodeJS та СУБД PostgreSQL, їх реплікацію, та забезпечення отримання цієї інформації різними шляхами.

Було створено програмне забезпечення, яке забезпечило виконання таких пунктів:

1. Засоби попередньої обробки, генерація даних скрапінгом.
2. Реплікація даних
3. Візуалізація та аналіз даних за допомогою бібліотек NodeJS.
4. Засоби підвищення швидкодії, зокрема, індекси БД
5. Консольний інтерфейс користувача.

В ході виконання даного курсового проекту було досягнуто поставленої мети: було набуто практичних навичок розробки сучасного програмного забезпечення, що взаємодіє з SQL базами даних, а також були здобуті навички оформлення відповідного текстового, програмного та ілюстративного матеріалу у формі проектної документації. У результаті виконання курсового проекту я навчився писати програмне забезпечення для SQL баз даних, володіти основами використання СУБД, а також інструментальними засобами аналізу, візуалізації та роботи з великими обсягами даних.

Література

1. Using Google Charts

<https://developers.google.com/chart/interactive/docs>

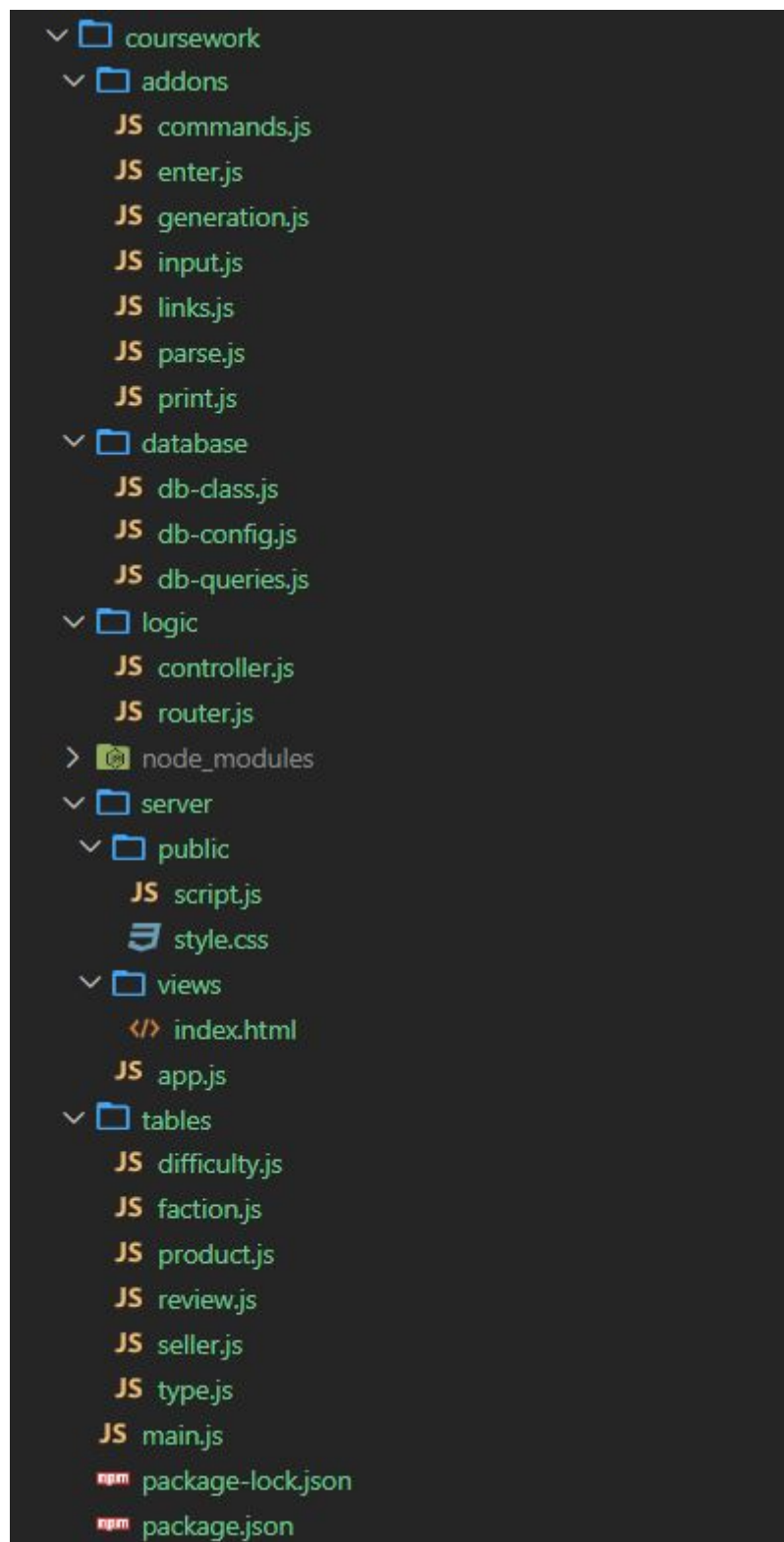
2. For developers, by developers

<https://stackoverflow.com/>

3. cheerio

<https://www.npmjs.com/package/cheerio>

Додатки



Скріншот структури файлів

Таблиці бази даних:

products

GeneralColumnsAdvancedConstraintsParametersSecuritySQL

	Name	Data type	Length/Precision	Scale	Not NULL?	Primary key?
	id	integer			<div>Yes</div>	<div>Yes</div>
	type_id	integer			<div>No</div>	<div>No</div>
	seller_id	integer			<div>No</div>	<div>No</div>
	name	text			<div>No</div>	<div>No</div>
	description	text			<div>No</div>	<div>No</div>
	cost	real			<div>No</div>	<div>No</div>
	faction_id	integer			<div>No</div>	<div>No</div>
	difficulty_id	integer			<div>No</div>	<div>No</div>
	active	boolean			<div>No</div>	<div>No</div>

Products

sellers

GeneralColumnsAdvancedConstraintsParametersSecuritySQL

Inherited from table(s)

Select to inherit from...

Columns

	Name	Data type	Length/Precision	Scale	Not NULL?	Primary key?
	id	integer			<div>Yes</div>	<div>Yes</div>
	name	text			<div>No</div>	<div>No</div>
	registration_date	timestamp without time zone			<div>No</div>	<div>No</div>

Sellers













reviews

GeneralColumnsAdvancedConstraintsParametersSecuritySQL

Inherited from table(s)

Select to inherit from...

Columns

	Name	Data type	Length/Precision	Scale	Not NULL?	Primary key?
 	id	integer			<div>Yes</div>	<div>Yes</div>
 	seller_id	integer			<div>No</div>	<div>No</div>
 	comment	text			<div>No</div>	<div>No</div>
 	mark	integer			<div>No</div>	<div>No</div>
 	date	timestamp without time zone			<div>No</div>	<div>No</div>
 	product_cost	integer			<div>No</div>	<div>No</div>

Reviews





difficulties

GeneralColumnsAdvancedConstraintsParametersSecuritySQL

Inherited from table(s)

Select to inherit from...

Columns

	Name	Data type	Length/Precision	Scale	Not NULL?	Primary key?
 	id	integer			<div>Yes</div>	<div>Yes</div>
 	name	text			<div>No</div>	<div>No</div>

Difficulties

factions

General

Columns

Advanced

Constraints

Parameters

Security

SQL

Inherited from table(s)

Select to inherit from...

Columns

	Name	Data type	Length/Precision	Scale	Not NULL?	Primary key?
<div><div></div><div></div></div>	id	integer			<div>Yes</div>	<div>Yes</div>
<div><div></div><div></div></div>	name	text			<div>No</div>	<div>No</div>

Factions

types

General

Columns

Advanced

Constraints

Parameters

Security

SQL

Inherited from table(s)

Select to inherit from...

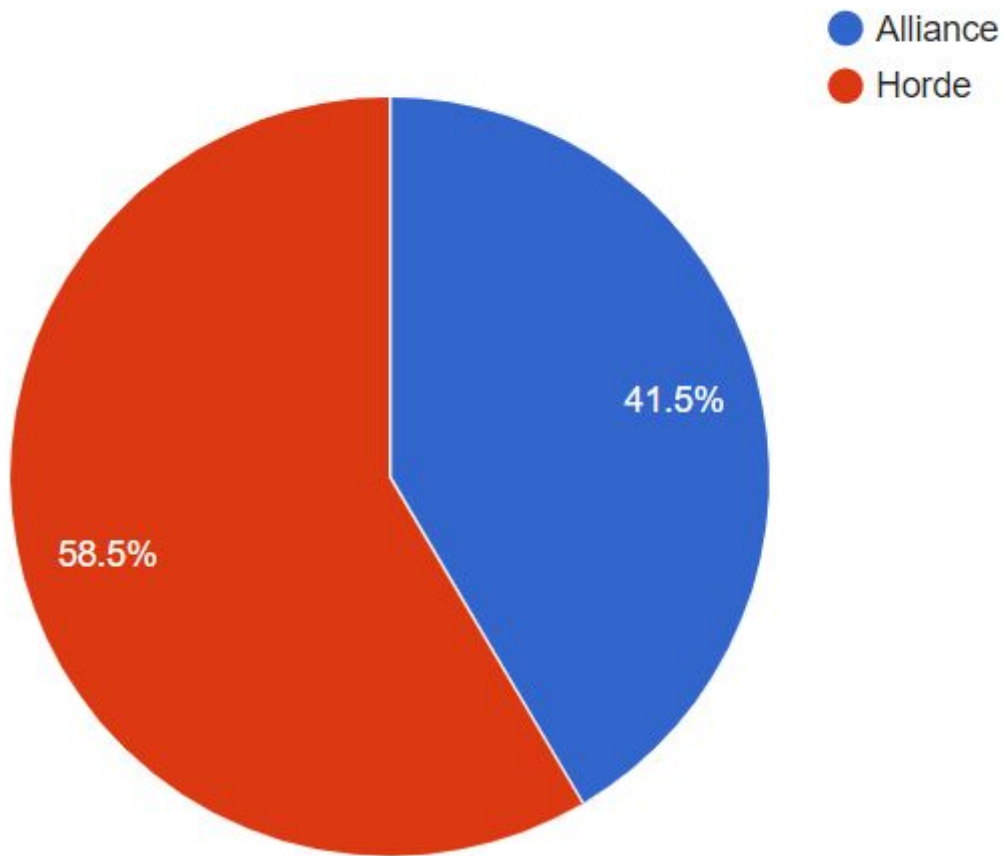
Columns

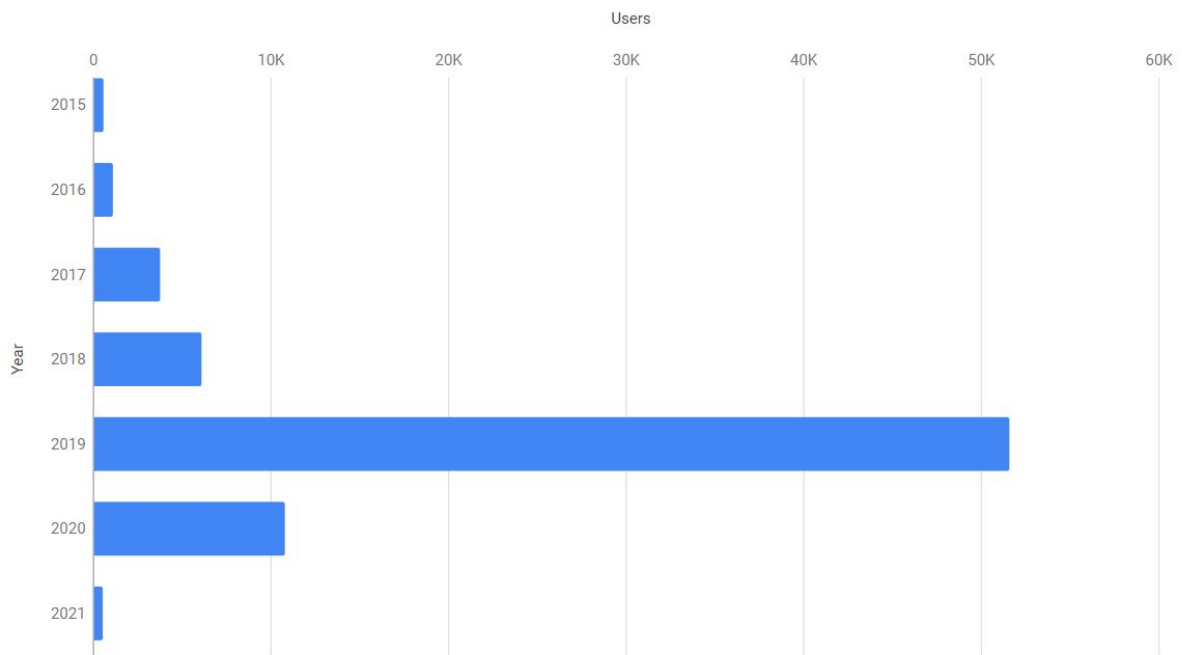
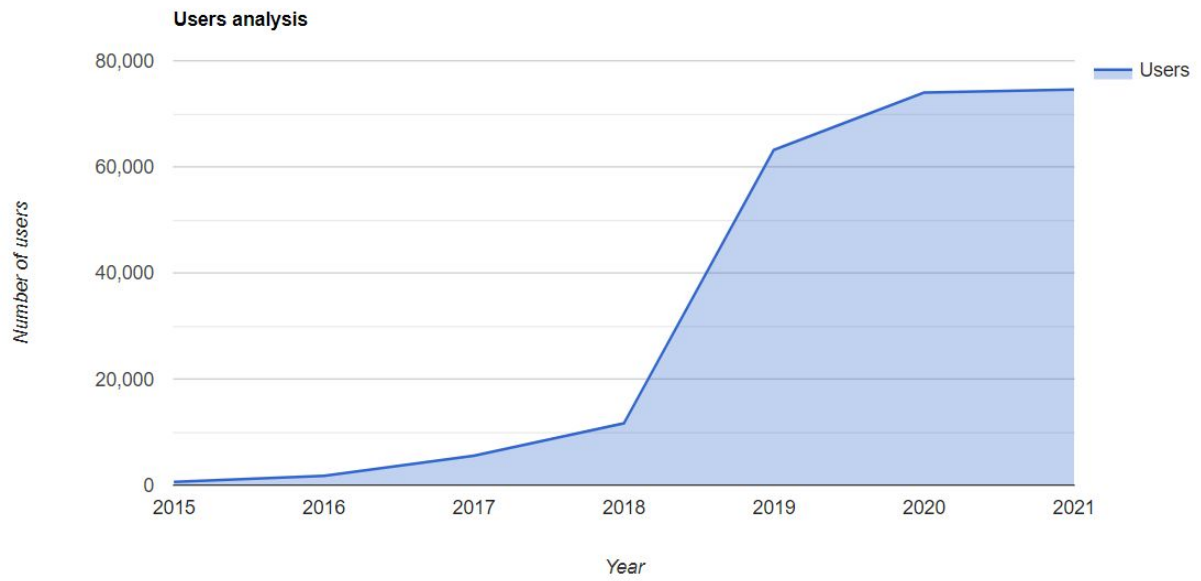
	Name	Data type	Length/Precision	Scale	Not NULL?	Primary key?
<div><div></div><div></div></div>	id	integer			<div>Yes</div>	<div>Yes</div>
<div><div></div><div></div></div>	name	text			<div>No</div>	<div>No</div>

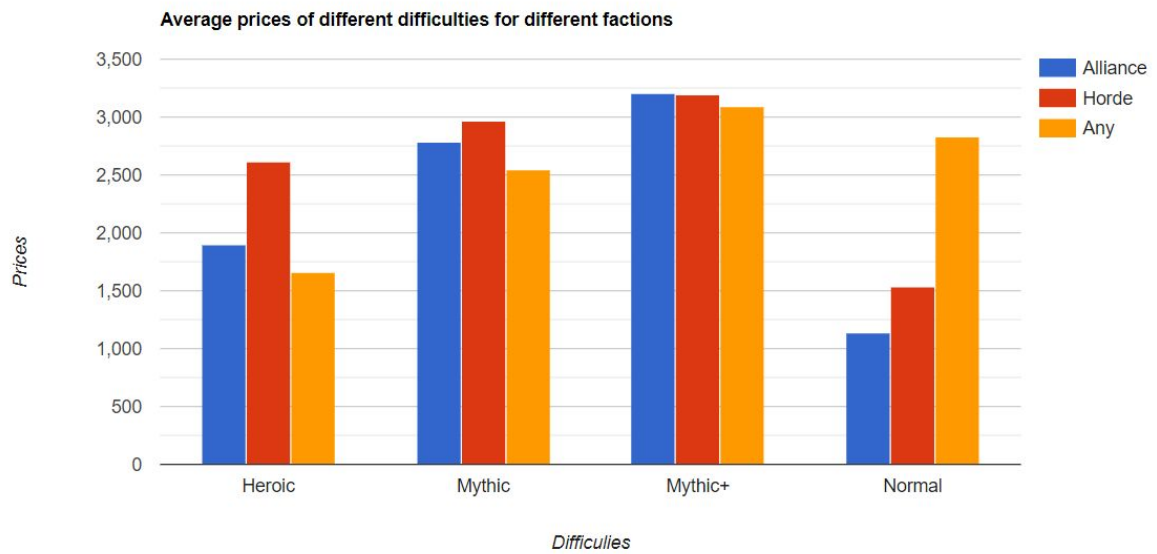
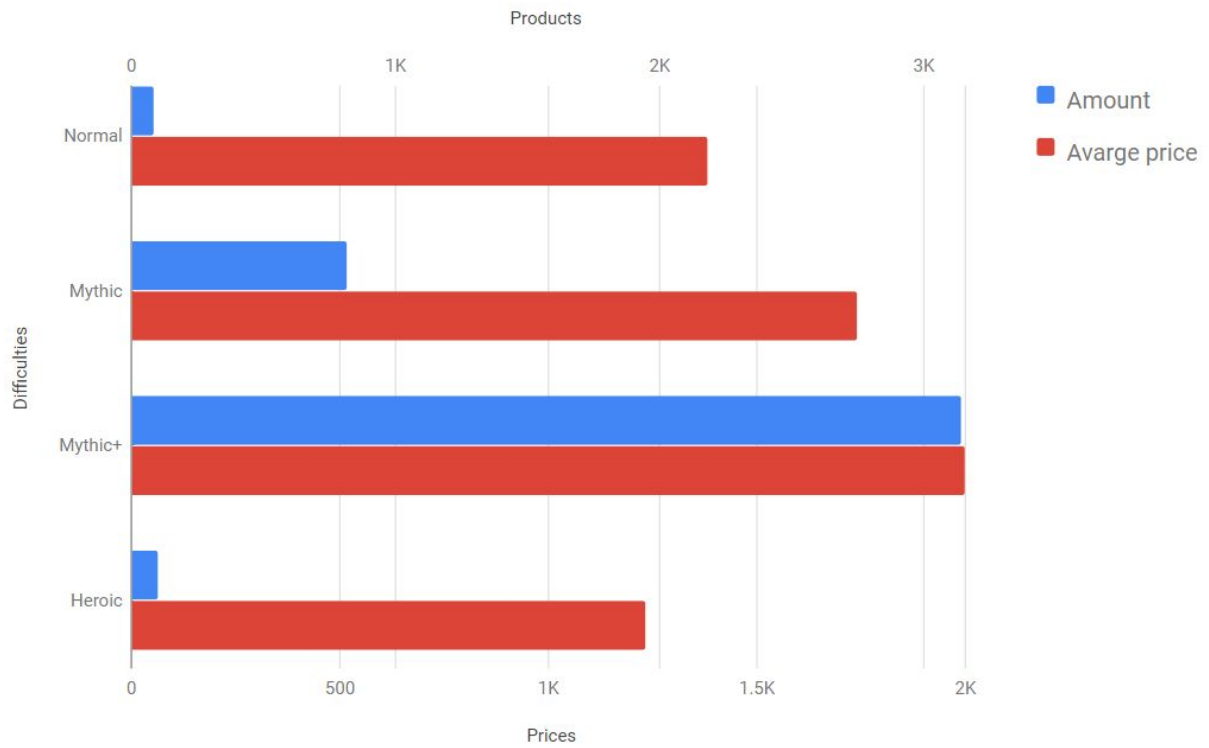
Types

Візуалізація аналізу даних:

Popularity of the factions

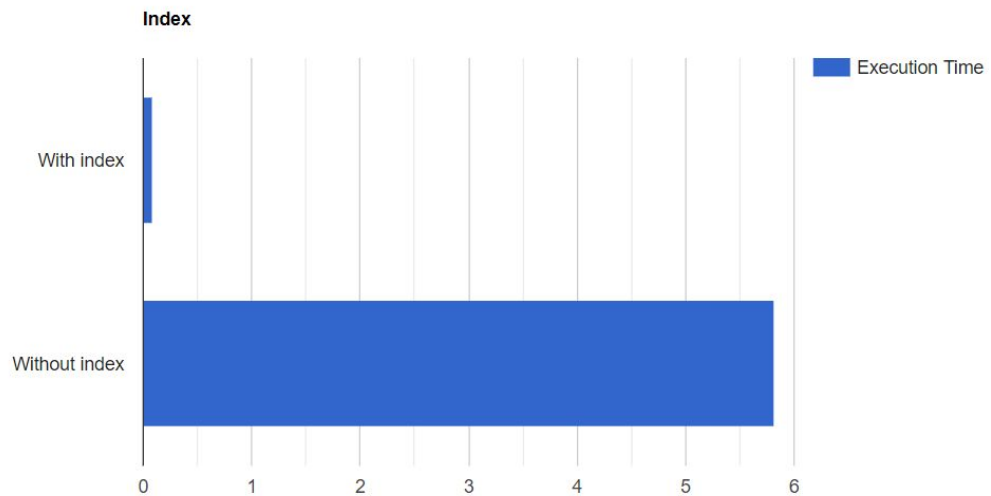




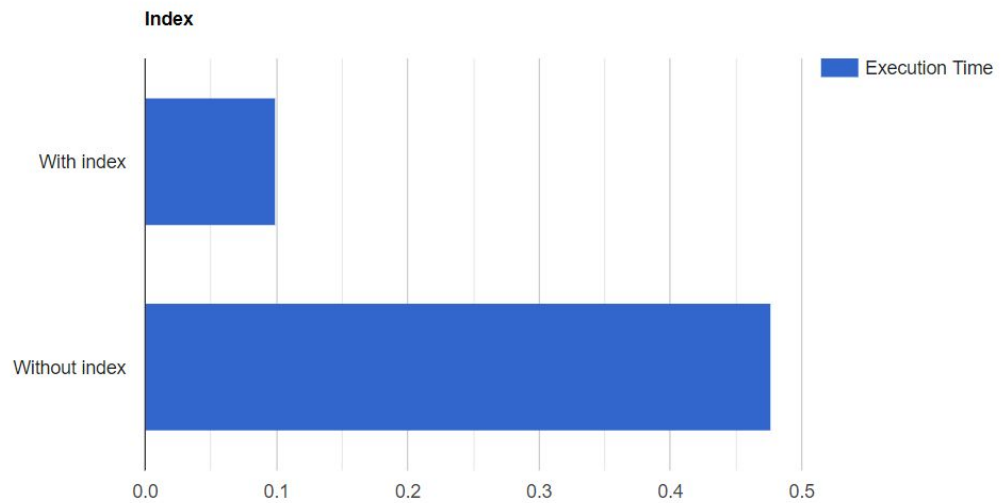


Візуалізація ефективності індексів:

```
SELECT * , reviews.comment FROM sellers  
INNER JOIN reviews ON reviews.seller_id = sellers.id  
WHERE name LIKE 'koltowa'  
OR sellers.id = 2000000
```



```
SELECT COUNT( * ) FROM products  
WHERE products.id BETWEEN 6377838 AND 6386774  
AND products.cost BETWEEN 1000 AND 3000
```



Реплікація:

▼ Servers (3)

> PostgreSQL 12

> master

> salve

master [Работает] - Oracle VM VirtualBox

Файл Машина Вид Ввод Устройства Справка

```
postgres@ubuntuserver1:~$ psql
psql (12.5 (Ubuntu 12.5-0ubuntu0.20.04.1))
Type "help" for help.

postgres=# \l

               List of databases
  Name          | Owner   | Encoding | Collate | Ctype   | Access privileges
-----+-----+-----+-----+-----+-----
 coursework    | postgres | UTF8      | C.UTF-8 | C.UTF-8 | =Tc/postgres +
               |          |          |          |          | postgres=CTc/postgres+
               |          |          |          |          | master=CTc/postgres
 postgres      | postgres | UTF8      | C.UTF-8 | C.UTF-8 |
 template0     | postgres | UTF8      | C.UTF-8 | C.UTF-8 | =c/postgres +
               |          |          |          |          | postgres=CTc/postgres
 template1     | postgres | UTF8      | C.UTF-8 | C.UTF-8 | =c/postgres +
               |          |          |          |          | postgres=CTc/postgres
 test_db       | postgres | UTF8      | C.UTF-8 | C.UTF-8 | =Tc/postgres +
               |          |          |          |          | postgres=CTc/postgres+
               |          |          |          |          | master=CTc/postgres
(5 rows)

postgres=# \du

               List of roles
 Role name | Attributes                                  | Member of
-----+-----+-----
 master   | Superuser                                  | {}
 postgres | Superuser, Create role, Create DB, Replication, Bypass RLS | {}
 replication | Replication                               | {}

postgres=# _
```

сервер master

```
salve [Работает] - Oracle VM VirtualBox
Файл Машина Вид Ввод Устройства Справка
postgres@ubuntuserver2:~$ psql
psql (12.5 (Ubuntu 12.5-0ubuntu0.20.04.1))
Type "help" for help.

postgres=# \du
                                List of roles
Role name |                               Attributes                               | Member of
-----+-----+-----+
master    | Superuser                                                            | {}
postgres  | Superuser, Create role, Create DB, Replication, Bypass RLS         | {}
replication | Replication                                                            | {}

postgres=# \l
                                List of databases
Name      | Owner   | Encoding | Collate | Ctype   | Access privileges
-----+-----+-----+-----+-----+-----+
coursework | postgres | UTF8      | C.UTF-8 | C.UTF-8 | =Tc/postgres +
           |          |           |         |         | postgres=CTc/postgres+
           |          |           |         |         | master=CTc/postgres
postgres   | postgres | UTF8      | C.UTF-8 | C.UTF-8 |
template0  | postgres | UTF8      | C.UTF-8 | C.UTF-8 | =c/postgres +
           |          |           |         |         | postgres=CTc/postgres
template1  | postgres | UTF8      | C.UTF-8 | C.UTF-8 | =c/postgres +
           |          |           |         |         | postgres=CTc/postgres
test_db    | postgres | UTF8      | C.UTF-8 | C.UTF-8 | =Tc/postgres +
           |          |           |         |         | postgres=CTc/postgres+
           |          |           |         |         | master=CTc/postgres
(5 rows)

postgres=#
postgres=#
```

сервер salve