

**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: Ринок автомобілів України

**Студент**

**групи КП-93**

**Торговських О. О.**

(ПІБ)

\_\_\_\_\_  
(підпис)

**Викладач**

**к.т.н, доцент кафедри СПіСКС**

**Петрашенко А.В.**

\_\_\_\_\_  
(підпис)

Захищено з оцінкою \_\_\_\_\_

Київ – 2020

# Анотація

У процесі виконання даної курсової роботи було набуто практичні навички написання програмного забезпечення, з використанням аналізу/фільтрації/пагінації даних та генерації даних у реляційну БД(за допомогою сторонніх засобів збору інформації - Інтернет). Результатом розробки є навички роботи з обраною СУБД та інструментами для управління даною СУБД. Дана програма здатна аналізувати ринок автомобілів, а саме показує користувачу які марки авто є найпопулярнішими, відповідно на них і є найбільший попит, також програма показує які типи автомобілів є найпопулярнішими, також програма може надати користувачу автомобілі в обраному ним ціновому діапазоні та має великі можливості для розширення функціоналу, завдяки архітектурному шаблону MVC що значно полегшує роботу та робить код більш структурованим. Користувачу надаються можливості додавати, видаляти та змінювати дані у СУБД, тобто взаємодіяти з базою за допомогою CRUD операцій. Нижче будуть описані подробиці виконання та проектування курсової роботи. Початок роботи був запланований на 25 жовтня 2020, а завершення – 5 грудня. План виконаний. Галузь застосування даного програмного забезпечення - сайт ринку авто в Україні. Весь код та документація даної курсової роботи збережено на GitHub репозиторії: <https://github.com/alex123411/Data-Base>

## **Зміст**

<b>Анотація</b>	<b>1</b>
<b>Вступ</b>	<b>3</b>
<b>Аналіз інструментарію для виконання курсової роботи</b>	<b>3</b>
<b>Структура бази даних</b>	<b>3</b>
<b>Опис програмного забезпечення</b>	<b>4</b>
<b>Аналіз функціонування засобів реплікації</b>	<b>4</b>
<b>Аналіз функціонування засобів резервування/відновлення</b>	<b>5</b>
<b>Опис результатів аналізу предметної галузі</b>	<b>5</b>
<b>Висновки</b>	<b>6</b>
<b>Додатки</b>	<b>Error! Bookmark not defined.</b>

# Вступ

На сьогоднішній день ринок авто як і довгі роки до цього не стоїть на місці. Кожного дня придумують нові технології і тим самим покращують різні сфери послуг, однією цих сфер являються сайти авторинків на які людина хоче зайти та підшукати потрібний їй автомобіль. Для того щоб максимально швидко та якісно виконати пошук на допомогу приходять різні програмні забезпечення, що допомагають користувачу знайти підходящий по різним характеристикам автомобіль, перед цим уважно проаналізувавши ринок, адже авто це достатньо серйозна покупка до якої потрібно ретельно підготуватись. Дане програмне забезпечення базується на датасетах, які можна отримати у мережі Інтернет зі справжніх сайтів авторинків.

## Аналіз інструментарію для виконання курсової роботи

У ролі середовища для розробки було використано Visual Studio Code. Дане середовище має широкий функціонал, приємне у використанні, запускає програму одним натисканням кнопки, має зручний графічний інтерфейс і є дуже популярною серед програмістів на сьогодні. Весь код був написаний мовою Python 3.8, яка не є останньою версією цієї мови, але вона підтримує усі необхідні інструменти для роботи із базою даних та аналізу даних. У ролі СУБД виступає PostgreSQL та його графічний інтерфейс у вигляді pgAdmin 4. Дана СУБД є досить потужною і популярною для великих обсягів даних. pgAdmin 4 надає зручний користувацький інтерфейс для роботи із даною СУБД. PostgreSQL – об'єктно-реляційна система управління базою даних. Її перевагами є її популярність, що сприяє тому, що в мережі Інтернет присутня величезна кількість інформації, в якій можна знайти всі відповіді на питання. Також, дана СУБД має потужні механізми для реплікації та транзакції даних. Ця СУБД є не тільки реляційною, а ще й об'єктно-орієнтовною, із засобами наслідування тощо. Також використовувалися сторонні бібліотеки та модулі, завантаженні за допомогою утиліти pip. (Про бібліотеки буде розказано більше у відповідних розділах нижче).

## Структура бази даних

База даних складається з таких таблиць:

- 1) publication\_info – що зберігає інформацію про публікацію авто на сайті користувачем
- 2) car\_main\_info – зберігає інформацію про найважливіші характеристики автомобіля

- 3) car\_addiotional\_info – зберігає інформацію про додаткові можливості автомобіля, якщо вони є.

Структуру бази даних в графічному вигляді можна побачити в додатках - перший пункт

## Опис програмного забезпечення

### Загальна структура програмного забезпечення

Програмне забезпечення побудоване за архітектурним шаблоном MVC із рознесенням відповідних модулів. Доступ до БД реалізований за технологією ORM за допомогою бібліотеки SQLAlchemy. Також для представлення у графічному вигляді аналізованих даних використовувалась бібліотека matplotlib(дана бібліотека дозволяє візуалізувати аналізовані дані, тобто будувати графіки, діаграми, гістограми тощо).

### Опис модулів програмного забезпечення

Генератор даних в базу складається із одного файлу data\_generation.py, який має декілька функцій, які здійснюють розбивають скачаний датасет у форматі csv на підтаблиці з якими в подальшому легше працювати та за потреби створює ті самі таблиці в базі. Функції парсингу та створення таблиць не є універсальними, але за потреби можна збільшити функціонал програми. Даний файл реалізований за допомогою бібліотек csv та sqlalchemy. Дані бібліотеки дозволяють виконувати парсинг та зчитування csv файлів та створення відповідних таблиць в базі і запис даних в них. Оскільки код написаний з використанням моделі MVC то проект відповідно має файли model, view, controller. У файлі model.py описані за технологією ORM SQLAlchemy таблиці бази даних та функції доступу до інформації в базі, також деякі функції аналізу даних з цієї бази. Файл view.py містить у собі код консольного меню користувача, через яке користувач може викликати різного типу функції та запити. У даному файлі містяться функції виводу інформації, та підтвердження дії, які викликаються контроллером як реакція на виконання роботи. Файл controller.py зберігає у собі функції за допомогою яких відбувається виклик різних функцій з файлу models.py та передача отриманої інформації назад користувачу в views.py. Основна логіка програми розписана у файлі model.py, натомість у controller.py забезпечений лише необхідний виклик функцій, як реакція на дії користувача. App.py головний файл з якого запускається сама програма через контроллер.

## Аналіз функціонування засобів реплікації

Реплікація була реалізована за допомогою другої інстанції власного серверу на одній Windows системі. Реплікація реалізована за принципом

master-slave, для цього у PostgreSQL існує відповідна роль. Реплікація призначена для забезпечення цілісності, узгодженості та доступності даних, що зберігаються у СУБД. У разі подій які призведуть до несправності основної бази можна отримувати дані з реплікаційної, але не записувати и змінювати.

## Аналіз функціонування засобів резервування/відновлення

Резервування бази даних реалізовано за допомогою стороннього програмного забезпечення SQLBackupAndFTP. Дана програма дозволяє створити з'єднання із різними СУБД , зокрема із PostgreSQL. Після підключення до необхідного серверу, вибравши необхідну для резервування БД можна вибрати опції та шлях зберігання резервування даних. Наявна можливість циклічного оновлення у обраний час. У даній роботі налаштоване резервування кожен день о 10 годині вечора, та видаленням старих резервних копій , які старіше одного місяця, після успішного, або неуспішного процесу резервування мені на пошту надходить повідомлення про завершення даної дії.

## Аналіз результатів підвищення швидкодії запитів

При спробі покращити швидкість запитів було створено три індекси для трьох основних типів аналізу, це індекси для полів price, mark, type. У першому випадку індекс значно покращив швидкість запиту програми для поля price а в інших він приніс незначні зміни що майже не помітні.

## Опис результатів аналізу предметної галузі

Під час використання даного програмного забезпечення був завантажений та проаналізований датасет на 20000 рядків. Так як інформація в датасеті була достатньо застарілою, а саме за 2019 рік її не можна вважати актуальною, але для демонстративного прикладу роботи програми прекрасно підійде. Проаналізувавши ринок 2019 року на сайті OLX найбільшу кількість авто на продаж було виставлено марки VAZ, що являється другою маркою пострадянського простору в топі разом з ZAZ. Але більшу частину ринку займають, звичайно, імпортовані автомобілі, такі як: volkswagen, mercedes, BMW та

інші. При виборі за ціновим діапазоном існує достатньо обширна кількість авто на будь-який бюджет, від 1000\$ до 100000\$ і більше. Також можемо побачити що найпопулярнішим видом авто є седан, він опереджає свого найближчого опонента(хетчбек) більш ніж удвічі, що є достатньо важливою інформацією для постачальників авто.

## Висновки

У даній курсовій роботі було розроблено програмне забезпечення для аналізу онлайн ринку автомобілів України, найцікавішої та найкорисніші інформації як для покупців так і продавців. У процесі роботи використовувалися такі сучасні та популярні бібліотеки для роботи з даними мовою Python як matplotlib, sqlalchemy, [psycopg2 та інші]. Код роботи було логічно розподілено на модулі, згідно з архітектурним шаблоном програмування MVC(model-view-controller). Для можливого пришвидшення виконання запитів до бази було використано індекси PostgreSQL. Для коректного розподілення даних з величезного та незручного датасету в СУБД було застосовано такі бібліотеки, як csv. Результатом роботи даного програмного забезпечення є графіки найпопулярніших марок та типів авто а можливість знайти авто за своїм ціновим діапазоном.

## Література

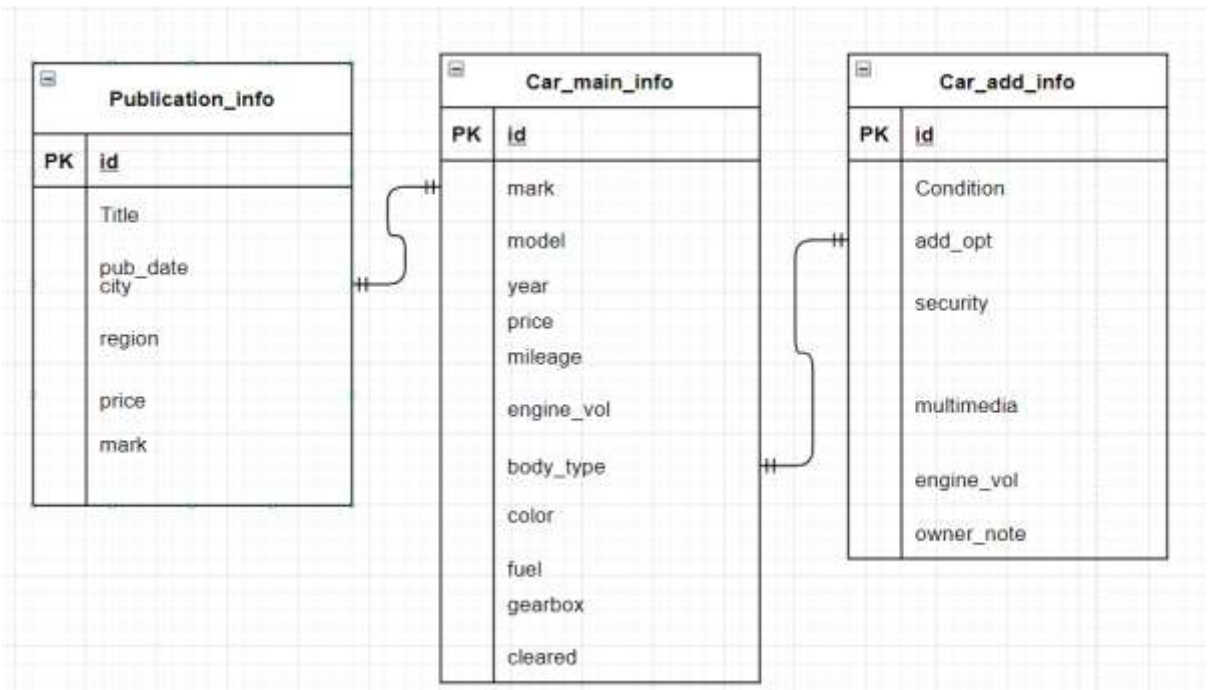
1. Data analysis using mean, median, mode  
<https://www.pbslearningmedia.org/resource/mwnet-math-spmmmr/data-analysis-using-mean-median-mode-and-range/>
2. Documentation postgresql <https://www.postgresql.org/docs/>
3. Python documentation <https://www.python.org/doc/>
4. Backup app <https://sqlbackupandftp.com/>
5. Анализ данных с использованием Python <https://habr.com/ru/post/353050/> 7.  
Репликация баз данных PostgreSQL по типу master-slave  
<https://www.8host.com/blog/replikaciya-baz-dannyx-postgresql-potipu-master-slave/>
6. “The clean Coder” Bob Martin 12. “Learn python 3 the hard way” Zed A. Shaw
7. ORM sqlalchemy tutorial <https://docs.sqlalchemy.org/en/13/orm/tutorial.html>



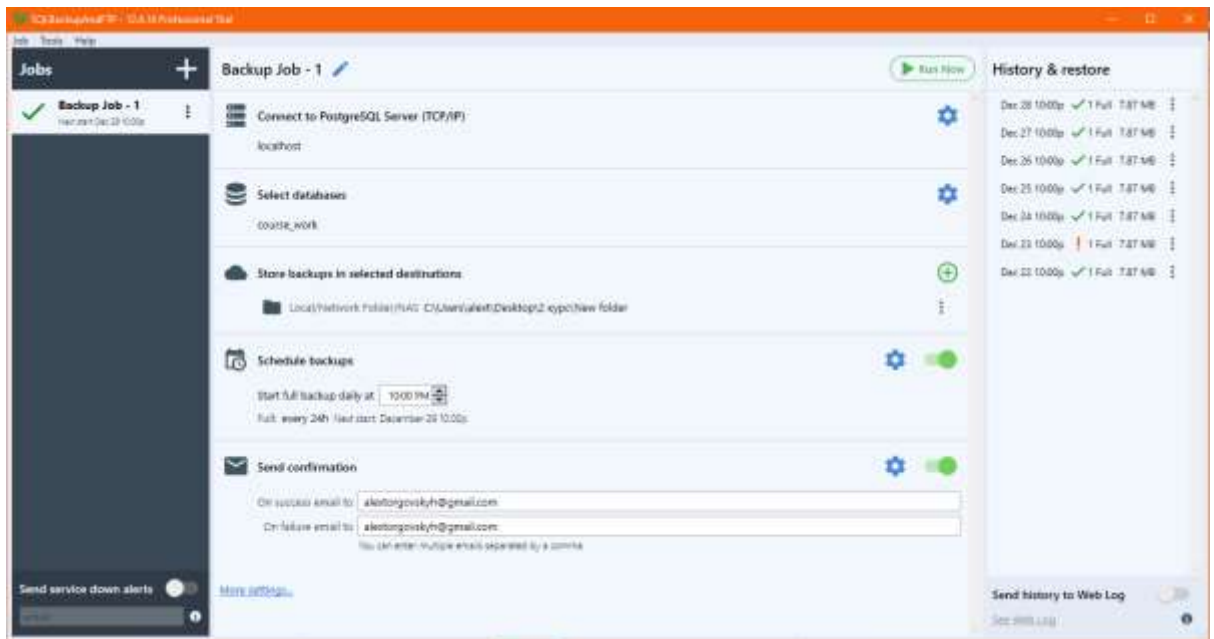
# Додатки

A

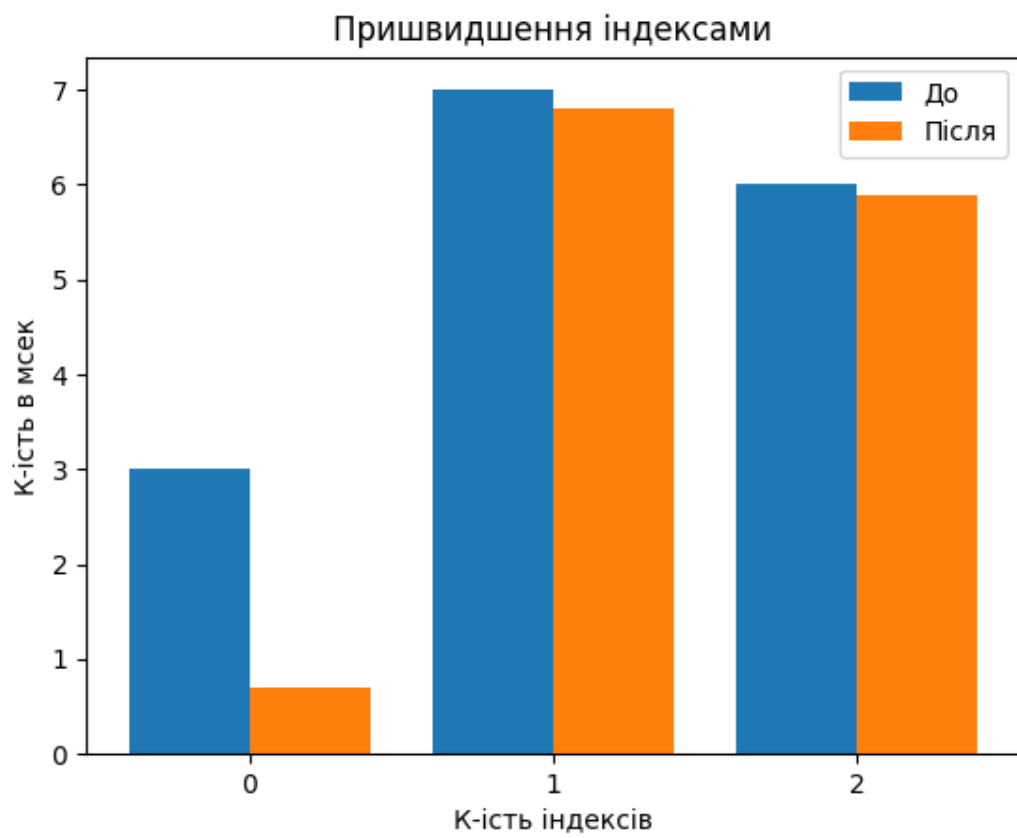
## 1) Структура БД



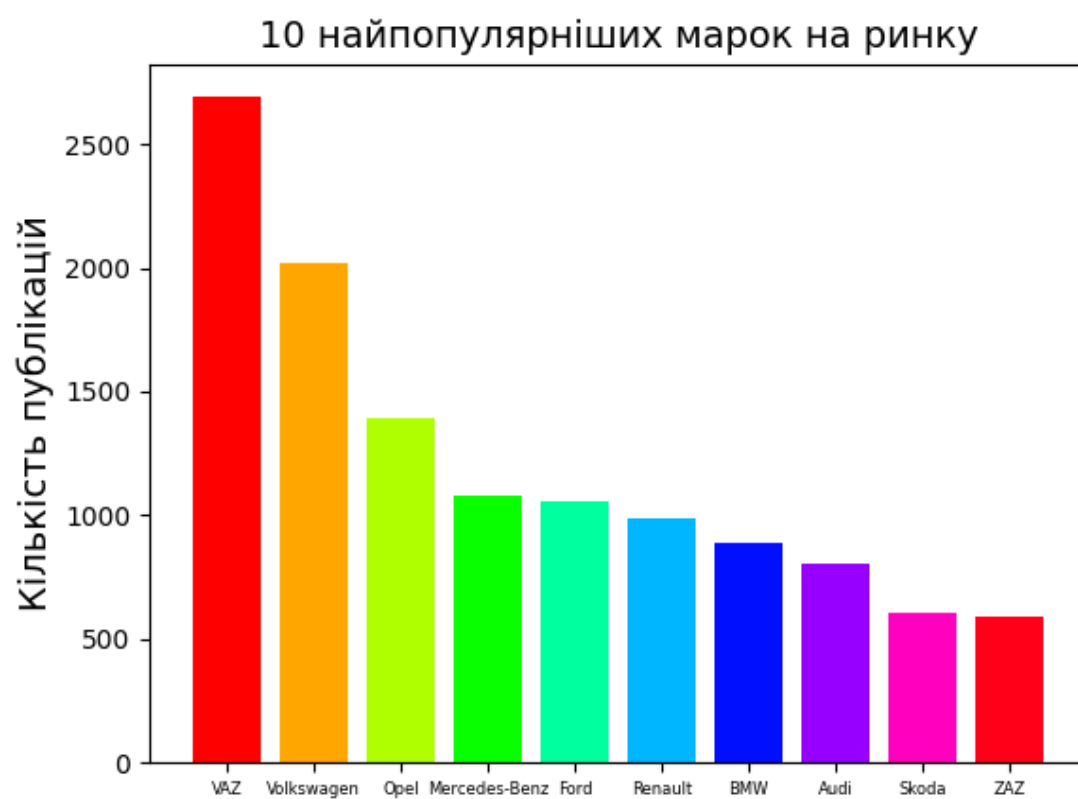
## 2) Скріншот додатку що забезпечує резервування даних мосї бази



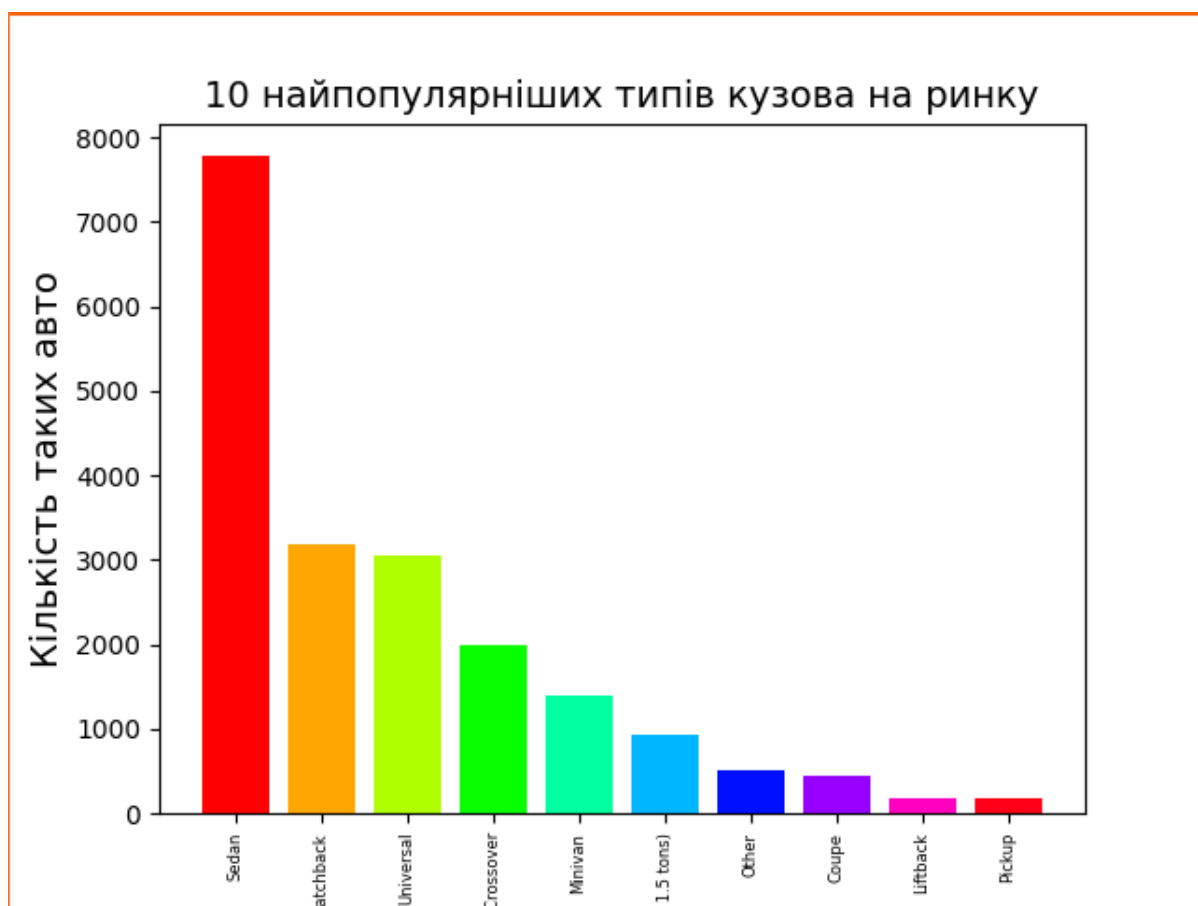
### 3)Результат покращення швидкодії за допомогою індексів



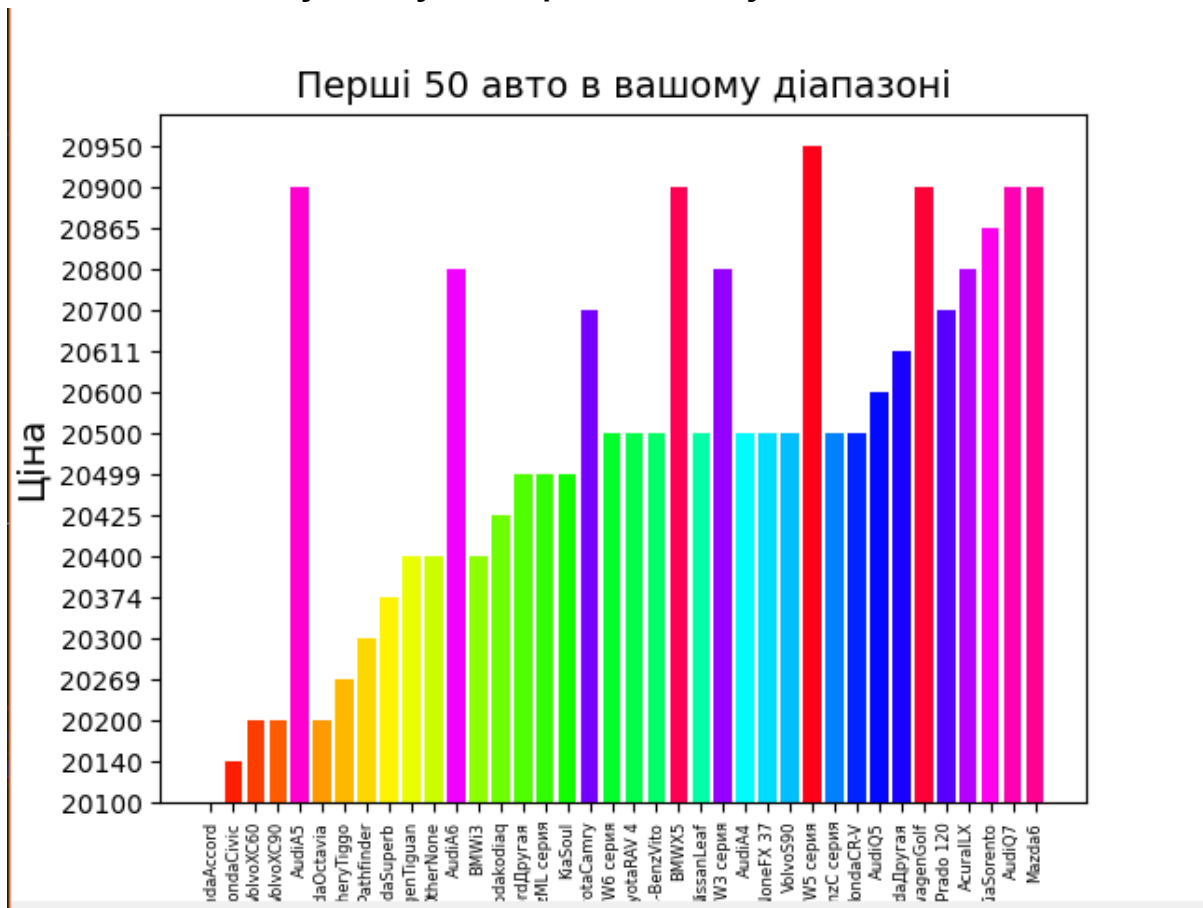
**4) Приклад аналізу даних при роботі програми**  
**А.**



**Б.**



**В. Після того як користувач ввів діапазон ціни в якій він хоче знайти для себе авто йому показують перші 50 що йдуть по ціні**



**5) Початковий екран консольного меню користувача(приклад роботи програми)**

```

-----
u have this commands:
1 - add new publication
2 - operations by id
3 - show the most popular marks
4 - show mostpopular types of cars
5 - show cars in your price diapason
9 - exit
select a command u wanna use

```

```
-----  
Заголовок: SsangYong corando (4wd)Dlx-3/  
Дата публікації: 10.05.2019  
Переглядів цієї публікації: 68  
Місто: Кривой Рог  
Регіон: Днепропетровская область  
Марка авто: SsangYong  
Ціна: 12600
```

```
-----  
u can:  
1 - see information about car in this publication  
2 - delete this publication  
3 - edit this publication  
4 - get back
```

```
-----  
Марка авто: SsangYong  
Модель авто: Korando  
Рік випуску авто: 2013  
Пробіг в км.: 76000  
Тип корпусу: SUV / Crossover  
Колір: Black  
Вмістимість двигуна: 2000  
Тип бензину: Diesel  
Коробка передач: Automatic  
Розтаможка: розтамований  
Ціна: 12600  
-----
```

Класи:

```
class Pub_info(Base):
    __tablename__ = '2_publication_info'
    id = Column(Integer, primary_key = True)
    title = Column(String)
    pub_date = Column(String)
    views = Column(Integer)
    city = Column(String)
    region= Column(String)
    mark= Column(String)
    price= Column(Integer)

class Car_main_info(Base):
    __tablename__ = '3_car_main_info'
    id = Column(Integer, primary_key = True)
    mark= Column(String)
    model= Column(String)
    year= Column(Integer)
    mileage= Column(Integer)
    body_type= Column(String)
    color= Column(String)
    engine_vol= Column(Integer)
    fuel= Column(String)
    gearbox= Column(String)
    cleared= Column(String)
    price= Column(Integer)

class Car_addit_info(Base):
    __tablename__ = '4_car_additional_info'
    id = Column(Integer, primary_key = True)
    condition= Column(String)
    add_opt= Column(String)
    multimedia= Column(String)
    security= Column(String)
    other= Column(String)
    owner_note= Column(String)
```



## 2) Приклад створення графіку

```
PLOT_LABEL_FONT_SIZE = 14
PLOT_MEANING_FONT_SIZE = 6
for record in records:
    records_x.append(record[0])
    records_y.append(record[1])
    print(str(record[0]) + " - " + str(record[1]) )
    x = x + 1
plt.title('10 найпопулярніших типів кузова на ринку', fontsize=PLOT_LABEL_FONT_SIZE)
plt.bar(records_x, records_y, color=getColors(len(records_x)))
plt.ylabel('Кількість таких авто', fontsize=PLOT_LABEL_FONT_SIZE)
plt.xticks(rotation = 90,fontsize=PLOT_MEANING_FONT_SIZE)
plt.show()
```

## 3) Приклад створення курсору

```
CREATE INDEX idxprice ON "2_publication_info" ((price))
```