

**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 контролю успішності учнів навчального закладу:  
збір, фільтрація та аналіз оцінок учнів різних вікових  
категорій за різними напрямками підготовки”

**Студентка**  
**групи КП-93**

**Адамчук Анна**

\_\_\_\_\_  
(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 датасету, індексації таблиць бази даних та роботи з ними.

Програма має консольний інтерфейс для взаємодії з базою даних. Програма має значний функціонал: перегляд інформації про учнів, перегляд списку класу та всіх вчителів учнів, пошук і вивід оцінок за певний період.

Результатом роботи є можливість отримання середніх оцінок учня по предметам, середніх оцінок класа по предметам, а також середніх оцінок серед класів загалом. На основі цих даних можна спрогнозувати вдосконалення навчального процесу.

## Зміст

Вступ	4
Аналіз інструментарію для виконання курсової роботи	5
Структура бази даних	6
Опис програмного забезпечення	8
1. Загальна структура програмного забезпечення	8
2. Опис модулів програмного забезпечення	8
3. Опис основних алгоритмів роботи	8
Аналіз функціонування засобів реплікації	9
Аналіз функціонування засобів резервування/відновлення	9
Аналіз результатів підвищення швидкодії запитів	9
Опис результатів аналізу предметної галузі	11
Висновки	12
Література	13
Додатки	14

## Вступ

Метою розробки є здобуття навичок використання СУБД та створення програмного забезпечення, яке буде взаємодіяти з реляційною базою даних, яка містить велику кількість даних, а також подальший аналіз цих даних і оформлення результатів у вигляді діаграм, графіків та таблиць.

База даних містить дані про учнів, предмети, а також учителів школи. Один вчитель в різних класах може викладати різні предмети. Також у базі містяться оцінки учнів та дні, коли вони були виставлені.

На основі вимірювань, складаються графіки середніх показників за семестр, що дає змогу побачити рівень знань певного учня чи класу. Аналізуючи ці дані, можна зробити висновки про вдосконалення учнем навчального процесу.

Дана робота є актуальною, адже дозволяє контролювати навчання кожного учня школи, отримувати середні дані, а також оцінки за певний проміжок часу і може використовуватись при складанні навчальної програми. Для наочного представлення підсумки місяця зображуються як графіки.

## **Аналіз інструментарію для виконання курсової роботи**

Для виконання курсової роботи було обрано СУБД PostgreSQL, тому що ця СУБД підтримує бази даних необмеженого розміру, має потужні і надійні механізми транзакцій і реплікації, може бути встановлена на будь-якій операційній системі, підтримує багато функціональних можливостей (транзакції, вкладені запити, зовнішні ключі, підтримка успадкування і т.д), дані опрацьовуються безпосередньо на сервері. Також PostgreSQL є безкоштовною програмою для роботи з реляційними базами даних. Курсова робота написана мовою програмування python. Для роботи із базою даних була використана бібліотека SQLAlchemy та модуль psycopg2. Для графічного відображення аналізу даних була використана бібліотека matplotlib.

SQLAlchemy – інструментарій SQL та об'єктно-реляційне відображення для мови програмування Python випущене під ліцензією MIT. SQLAlchemy вважає базу даних механізмом реляційної алгебри, а не просто сукупністю таблиць. Рядки можна вибирати не тільки з таблиць, але також з об'єднань та інших операторів вибору; будь-яка з цих одиниць може бути складена у більшу структуру. Бібліотека бере на себе завдання автоматизації надлишкових завдань, тоді як розробник залишається під контролем того, як організована база даних і як побудований SQL.

Matplotlib – бібліотека на мові програмування Python для візуалізації даних двовимірною 2D графікою (3D графіка також підтримується). Отримувані зображення можуть бути використані як ілюстрації в публікаціях.

Також використовувалися бібліотека Python datetime для роботи з такими типами даних як дата та час.

## Структура бази даних

База даних проекту включає в себе 6 таблиць:

- student – таблиця з даними про учня
  1. student\_id – первинний ключ
  2. name – ім'я учня
  3. surname – прізвище учня
  4. date\_of\_birthday – дата народження учня
  5. class\_id\_fk – FK, вторинний ключ, який вказує на поле class\_id таблиці class і означає належність даного учня до цього класа.
- teacher – таблиця з даними працівника
  1. teacher\_id - первинний ключ
  2. name – ім'я вчителя
  3. surname – прізвище вчителя
- class – таблиця з даними про клас
  1. class\_id – первинний ключ
  2. name – назва класа
  3. teacher\_id\_fk – FK, вторинний ключ, який вказує на поле teacher\_id таблиці teacher і означає, що цей вчитель є класним керівником класа
- subject – таблиця з даними про предмет
  1. subject\_id - первинний ключ
  2. name – назва предмету

- mark – таблиця з даними про оцінку:
  1. id – первинний ключ
  2. data – день, коли була отримана оцінка
  3. student\_id\_fk – FK, вторинний ключ, який вказує на поле student\_id таблиці student і означає, що цей учень отримав оцінку
  4. subject\_id\_fk – FK, вторинний ключ, який вказує на поле subject\_id таблиці subject і означає, що оцінка по цьому предмету
  
- teacher\_subject\_class – допоміжна таблиця зв'язків, яка визначає який вчитель веде який предмет у класі.
  1. link\_id – первинний ключ
  2. teacher\_id\_fk – FK, вторинний ключ, який вказує на поле teacher\_id таблиці teacher
  3. subject\_id\_fk – FK, вторинний ключ, який вказує на поле subject\_id таблиці subject
  4. class\_id\_fk – FK, вторинний ключ, який вказує на поле class\_id таблиці class

Усі таблиці є нормалізованими до 3 нормальної форми. Див. Додаток 1

## **Опис програмного забезпечення**

### **Загальна структура програмного забезпечення**

Програмні засоби містять наступні компоненти:

1. Підсистема попередньої обробки даних, що складається з:
  - 1.1. Засобів генерації даних. Є рандомізація даних, також було створено CRUD операції, що дозволяють додавати, видаляти та оновлювати дані у таблиці.
  - 1.2. Засобів фільтрації та валідації даних.
2. База даних системи призначена для зберігання і аналізу інформації розробленої моніторингової системи.
3. Засоби аналізу даних.

### **Опис модулів програмного забезпечення**

Програмне забезпечення розроблене відповідно до структури MVC. У проєкті є консольне меню, яке дозволяє обрати користувачу необхідну операцію. Програмне забезпечення містить один контролер, одне подання та модель, де безпосередньо відбувається взаємодія з базою даних, для отримання доступу до потрібної інформації. Також через модель відбувається додавання, видалення та оновлення даних в таблицях student та mark.

### **Опис основних алгоритмів роботи**

При псевдовипадковій генерації дня народження, дані генерувались з діапазону 2005 до 20011, тобто всі діти шкільного віку; при генерації дати для оцінки дата генерувалась з діапазону від 01.09.2020 до 31.12.2020 включно. На генерацію оцінок були теж накладені обмеження: так, оцінка не може бути менша за 2 та більша за 5.



```

1 insert into mark (student_id_fk, subject_id_fk, mark, date)
2 select row_student(num),
3 row_subject(num),
4 (random() * (5 - 2) + 1)::integer,
5 ('2020/09/01'::date + trunc(random() * ('2020/12/31'::date - '2020/09/01'::date)) * '1 day'::interval)
6 from generate_series(1, num)

1 insert into student (name, surname, date_of_birthday, class_id_fk)
2 select substring(md5(random()::text), 1, 7),
3 substring(md5(random()::text), 1, 7),
4 ('2005/01/01'::date + trunc(random() * ('2011/12/31'::date - '2005/01/01'::date)) * '1 day'::interval),
5 row_class(num)
6 from generate_series(1, num)

```

При генерації в таблиці де є посилання на зовнішній ключ – значення бралось з існуючих id. Для цього були написані функції, що для випадкового вибору id з наявних у таблиці, це було потрібно для забезпечення вірних даних.

```

1 select class_id from generate_series (1,num), class
2 order by random() limit num

1 select student_id from generate_series (1,num), student
2 order by random() limit num

1 select subject_id from generate_series (1,num), subject
2 order by random() limit num

```

## Аналіз функціонування засобів реплікації

При виконанні курсової роботи реплікація бази даних не була реалізована.

## Аналіз функціонування засобів резервування/відновлення

При виконанні курсової роботи резервування/відновлення бази даних не були реалізовані.

## Аналіз результатів підвищення швидкодії запитів

Для запитів, які використовуються для пошуку, були використані індекси для підвищення їх швидкодії. Були застосовані індекси BTree, адже пошук здійснюється у заданому діапазоні з використанням знаків оператора between в умові.

## До використання індексу:

```
1 explain analyze (select mark, date from mark where student_id_fk = 30 and
2 subject_id_fk = 3 and date between '2020-09-01' and '2020-12-31' order by date)
```

Data Output Explain Messages Notifications

QUERY PLAN	
text	
1 Gather Merge (cost=6458.71..6469.40 rows=93 width=8) (actual time=17.964..19.264 rows=157 loops=1)	
2 Workers Planned: 1	
3 Workers Launched: 1	
4 -> Sort (cost=5458.70..5458.93 rows=93 width=8) (actual time=15.831..15.837 rows=78 loops=2)	
5 Sort Key: date	
6 Sort Method: quicksort Memory: 29kB	
7 Worker 0: Sort Method: quicksort Memory: 27kB	
8 -> Parallel Seq Scan on mark (cost=0.00..5455.66 rows=93 width=8) (actual time=0.200..15.737 rows=78 loops=2)	
9 Filter: ((date >= '2020-09-01'::date) AND (date <= '2020-12-31'::date) AND (student_id_fk = 30) AND (subject_id_fk = 3))	
10 Rows Removed by Filter: 149804	
11 Planning Time: 0.065 ms	
12 Execution Time: 19.297 ms	

## Після використання індексу:

```
1 create index mrk_in on mark (student_id_fk, subject_id_fk, date);|
2 explain analyze (select mark, date from mark where student_id_fk = 30 and
3 subject_id_fk = 3 and date between '2020-09-01' and '2020-12-31' order by date)
```

Data Output Explain Messages Notifications

QUERY PLAN	
text	
1 Sort (cost=495.76..496.15 rows=158 width=8) (actual time=0.197..0.208 rows=157 loops=1)	
2 Sort Key: date	
3 Sort Method: quicksort Memory: 32kB	
4 -> Bitmap Heap Scan on mark (cost=6.83..489.99 rows=158 width=8) (actual time=0.043..0.163 rows=157 loops=1)	
5 Recheck Cond: ((student_id_fk = 30) AND (subject_id_fk = 3) AND (date >= '2020-09-01'::date) AND (date <= '2020-12-31'::date))	
6 Heap Blocks: exact=148	
7 -> Bitmap Index Scan on mrk_in (cost=0.00..6.79 rows=158 width=0) (actual time=0.023..0.023 rows=157 loops=1)	
8 Index Cond: ((student_id_fk = 30) AND (subject_id_fk = 3) AND (date >= '2020-09-01'::date) AND (date <= '2020-12-31'::date))	
9 Planning Time: 0.089 ms	
10 Execution Time: 0.256 ms	

Зокрема швидкість пошуку у таблиці так збільшилась у 100 разів з використанням індексів, для цього ми порівняли швидкість виконання до створення і після створення 10 разів. Діаграми порівняння швидкодії запитів наведені у Додатку 2.

before	after	
21,094	0,208	
20,591	0,156	
17,314	0,278	
19,942	0,208	
18,423	0,112	
17,695	0,086	
19,752	0,231	
19,305	0,302	
17,398	0,25	
19,518	0,228	

## Опис результатів аналізу предметної галузі

У даному додатку є можливість отримувати такі дані:

- Інформація про учня
- Список класа
- Список учителів класа

У даному додатку є можливість аналізувати такі дані:

- Оцінки учня по предмету
- Середні оцінки учня по предметам
- Середні оцінки класа по предметам
- Середні оцінки по класам

Введення усіх даних, що підлягають до аналізу відбувається за допомогою консольного інтерфейсу. Всі дані проходять валідацію і в разі невідповідності про це повідомляється користувачу.

Усі графіки можуть (за бажання користувача) зберігатися у відповідній папці проекту. Назву графіку користувач може задати самостійно. Уся інформація виводиться у консоль.

Приклади будуть наведені у додатках 3, 4, 5, 6, 7, 8, 9.

## Висновки

Отже, у курсовій роботі ми створили програмне забезпечення, яке дозволяє генерувати правильні дані і записувати їх у реляційну базу даних PostgreSQL. Також створена можливість додавання, видалення і оновлення записів у таблиці. Створена можливість отримувати середні показники у вигляді діаграм або виводу у консоль. Тому дане програмне забезпечення може використовуватись у сфері навчального процесу. Для написання програми ми використали, модуль `psycopg2`, `matplotlib` та `SQLAlchemy`. Також програма дозволяє здійснювати пошук, який виконується швидше завдяки використанню індексів.

Не все, що планувалося, було зроблено в кінцевому варіанті. Відсутня реплікація та система бекапів – мінус розробленого додатку.

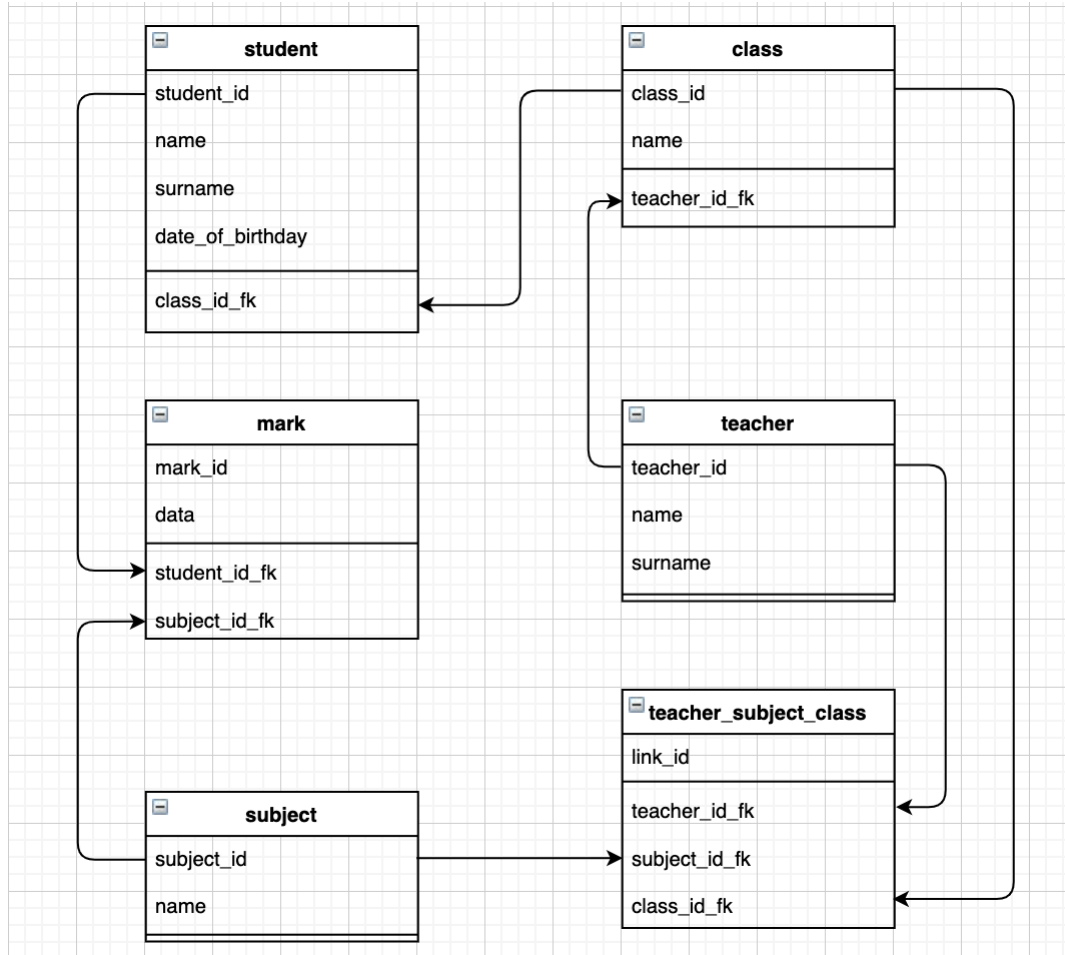
Хоч і не все було втілено, але головне це отримане знання з програмування на новій для нас мові Python та досвід роботи з реляційною системою управління базами даних PostgreSQL, який був отриманий під час виконання роботи.

## Література

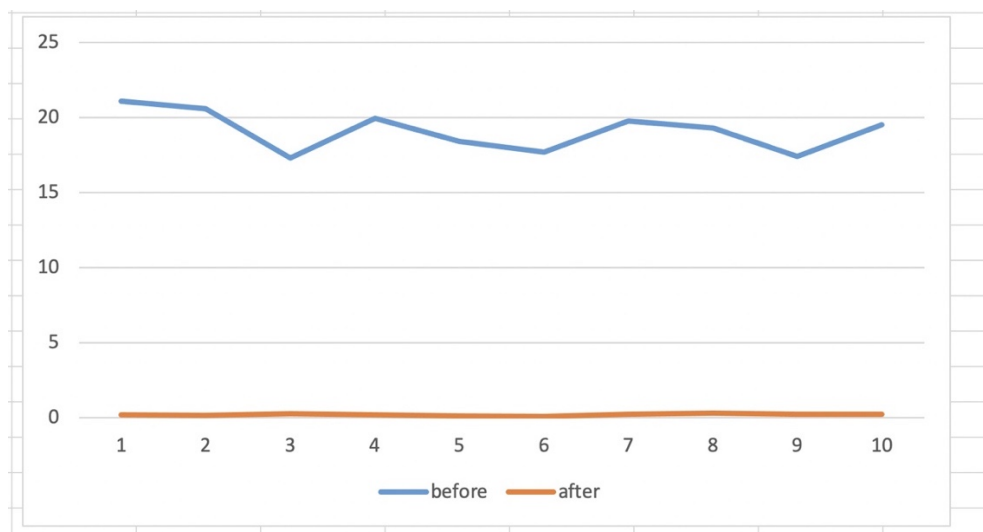
1. What is PostgreSQL? [Електронний ресурс] – Режим доступу до ресурсу: <https://www.educba.com/what-is-postgresql/>.
2. Matplotlib: Visualization with Python [Електронний ресурс] – Режим доступу до ресурсу: <https://matplotlib.org/>
3. Psycopg2 Documentation [Електронний ресурс] – Режим доступу до ресурсу: <https://pypi.org/project/psycopg2/>
4. PostgreSQL Documentation [Електронний ресурс] – Режим доступу до ресурсу: <https://www.postgresql.org/docs/>
5. SQLAlchemy ORM [Електронний ресурс] – Режим доступу до ресурсу: <https://docs.sqlalchemy.org/en/13/orm>
6. Mimesis Documentation [Електронний ресурс] – Режим доступу до ресурсу: <https://pypi.org/project/mimesis/>
7. PgAdmin 4 4.28 documentation [Електронний ресурс] – Режим доступу до ресурсу: <https://www.pgadmin.org/docs/pgadmin4/4.28/index.html>.
8. Time. Time access and conversions. [Електронний ресурс] – Режим доступу до ресурсу: <https://docs.python.org/3/library/time.html>

## Додатки

Додаток 1. Схема реляційної бази даних



Додаток 2. Порівняння швидкодії запитів для таблиці mark



### Додаток 3. Отримання інформації про учня

```
Show information about student
Enter name: >? asd
Enter surname: >? asd
```

student_id	fullname	date_of_birthday	class	teacher_fullname
1	asd asd	2004-12-01	11	Susan King

### Додаток 4. Отримання списку класа

```
Show a list of a class
Enter class: >? 9B
```

fullname	date_of_birthday
ca749bc 07429a1	2005-05-15
0271545 13e6f2c	2007-12-01
39a6dd7 1ccae1d	2004-07-16
6e7b9c6 2349a1c	2013-11-10
9439a8a 2437664	2020-09-25
aa5340e 3ddc8f7	2006-08-15
8a90240 999136e	2020-11-20
2f2be88 b9c421c	2020-10-19
4872f95 bfddf5e	2005-04-07
9a076bb c079134	2020-11-24
5d8bd6a c85eb32	2007-04-17
d0a19fd dd725b2	2020-10-29
476a358 e436132	2007-07-02
c539665 eb84f87	2020-10-14
46e60ff f3605e9	2008-08-30
9d2a0a4 f4baacc	2007-03-27
c05f0cc f8a31dd	2020-11-14

### Додаток 5. Отримання списку вчителів учня

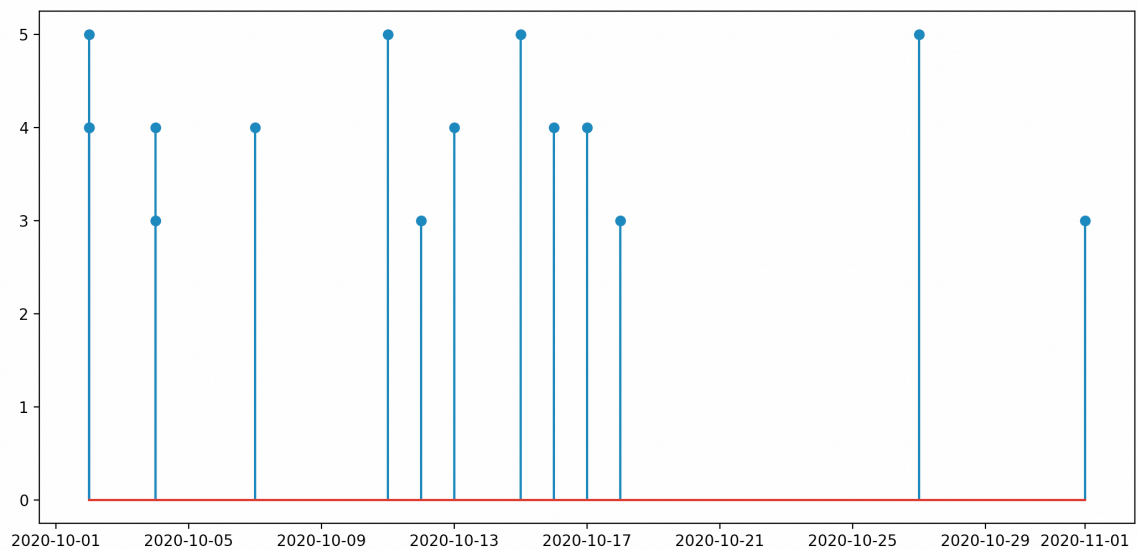
```
Show the teachers of student
Enter name: >? asd
Enter surname: >? asd
```

teacher_id	fullname	subject
11	Susan King	history
7	Isabella Parson	chemistry
6	Isla Wilson	informatics
12	Sandra Moore	biology
4	Emily Johnson	arts
1	Airic Adamson	literature
5	Jessica Davies	physics
4	Emily Johnson	language
8	Poppy Gilbert	PE
1	Airic Adamson	math

## Додаток 6. Отримання інформації щодо оцінок учня за період

```
Show the marks of student on the period
Enter name: >? asd
Enter surname: >? asd
Enter subject: >? math
Enter start date (yyyy-mm-dd): >? 2020-10-01
Enter end date (yyyy-mm-dd): >? 2020-11-01
```

mark	date
5	2020-10-02
4	2020-10-02
4	2020-10-02
4	2020-10-04
3	2020-10-04
4	2020-10-07
5	2020-10-11
3	2020-10-12
4	2020-10-13
5	2020-10-15
4	2020-10-16
4	2020-10-17
3	2020-10-18
5	2020-10-27
3	2020-11-01

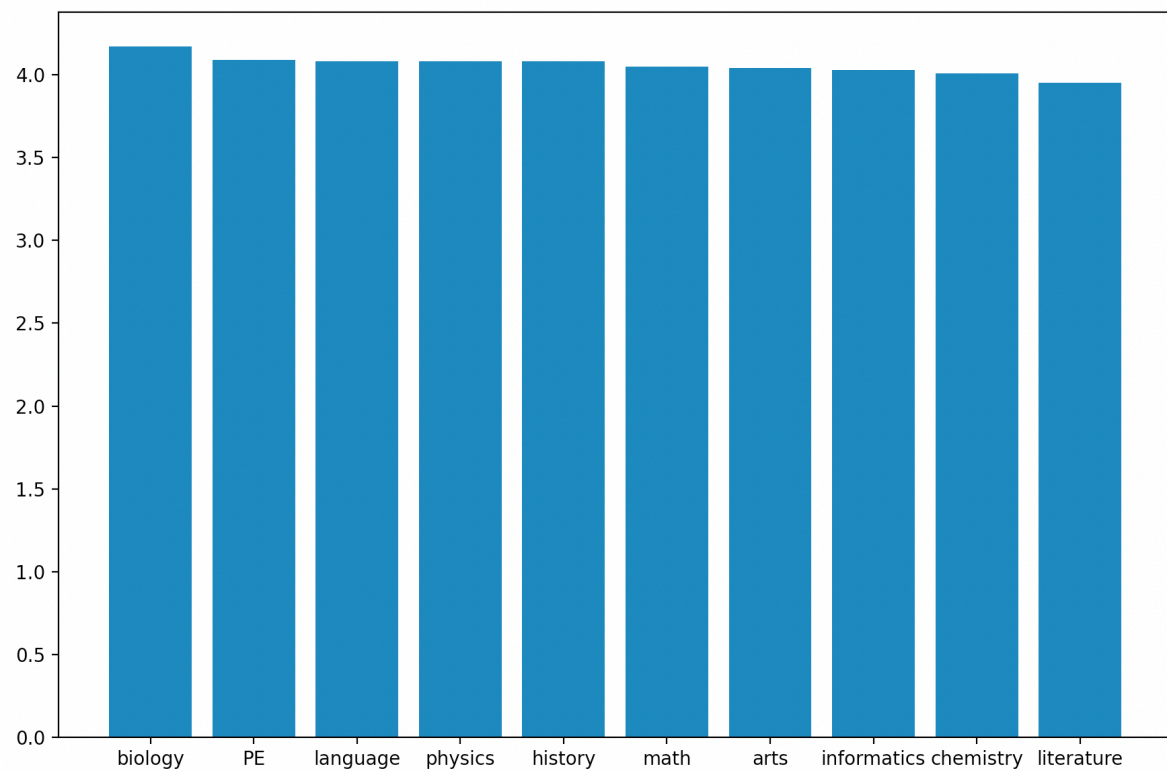




## Додаток 7. Отримання інформації щодо середніх оцінок учня по предметах

```
Show student average mark
Enter name: >? asd
Enter surname: >? asd
```

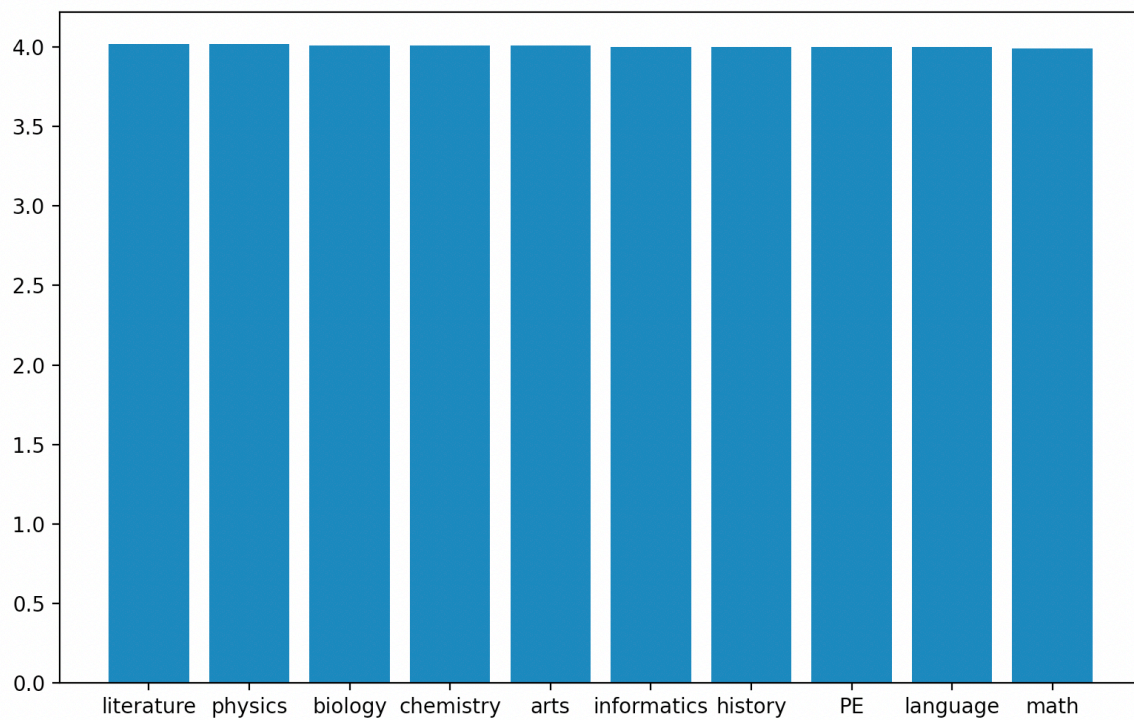
name	avg_mark
biology	4.17
PE	4.09
language	4.08
physics	4.08
history	4.08
math	4.05
arts	4.04
informatics	4.03
chemistry	4.01
literature	3.95



## Додаток 8. Отримання інформації щодо середніх оцінок класа по предметах

```
Show class average mark on subject
Enter class: >? 10
```

name	avg_mark
literature	4.02
physics	4.02
biology	4.01
chemistry	4.01
arts	4.01
informatics	4.00
history	4.00
PE	4.00
language	4.00
math	3.99



Додаток 9. Отримання інформації щодо середніх оцінок серед класів.

Show classes average marks

name	avg_mark
11	4.02
5B	4.01
8B	4.01
5A	4.01
10	4.01
6B	4.01
6A	4.01
9B	4.00
7A	4.00
7B	4.00
8A	4.00
9A	4.00

