

**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 студентів навчального закладу”

Студент
групи КП-91(92,93)

Семьонова Поліна
Дмитрівна

(підпис)

Викладач
к.т.н, доцент кафедри
СПіСКС

Петрашенко А.В.

(підпис)

Захищено з оцінкою _____

Київ – 2020

Анотація

Під час виконання курсової роботи був створений додаток для аналізу роботи університету та успішності студентів. Для зручного зберігання великої кількості інформації про університет була створена база даних. Також в курсовій роботі було використано декілька шляхів генерування даних, щоб продемонструвати роботу додатка.

Зміст

Вступ	3
Аналіз інструментарію для виконання курсової роботи	4
Структура бази даних	5
Опис програмного забезпечення	6
Загальна структура програмного забезпечення	6
Опис модулів програмного забезпечення	6
Опис основних алгоритмів роботи	7
Аналіз функціонування засобів реплікації	8
Аналіз функціонування засобів резервування/відновлення	9
Аналіз результатів підвищення швидкодії запитів	10
Опис результатів аналізу предметної галузі	11
Література	13
Додатки	14

Вступ

Під час аналізу якості навчального процесу університетами, з метою покращення ефективності та доступності подання матеріалів, виникає необхідність в створенні зручної програми, яка б фільтрувала дані за певними критеріями, аналізувала успішність студентів, ціни на спеціальності, кількість викладачів з певними позиціями, кількість предметів, які викладають викладачі на факультеті та для наочності могла б візуально продемонструвати аналіз, щоб в подальшому експерти в цій сфері могли б дати оцінку роботи факультетів та викладачів та за необхідності прийняти певні міри для покращення навчального процесу.

Аналіз інструментарію для виконання курсової роботи

Для роботи була обрана СУБД PostgreSQL. Вона має значні переваги над іншими SQL базами даних з відкритим вихідним кодом, тому що PostgreSQL не просто реляційна, а об'єктно-реляційна СУБД.

Фундаментальна характеристика об'єктно-реляційної бази даних - це підтримка об'єктів і їх поведінки, включаючи типи даних, функції, операції, домени і індекси. Це робить PostgreSQL неймовірно гнучким і надійним. Серед іншого, він вміє створювати, зберігати та видавати складні структури даних. Існує великий список типів даних, які підтримує PostgreSQL.

Для розробки програмного додатку була обрана мова Python, оскільки це мова програмування загального призначення, націлена в першу чергу на підвищення швидкості написання коду. На Python можна написати практично що завгодно без відчутних проблем. Код багато в чому лаконічний і зрозумілий. За рахунок простоти коду, подальший супровід програм, написаних на Python, стає легше і приємніше в порівнянні з іншими мовами програмування. Також, оскільки нами була обрана СУБД PostgreSQL, нам необхідна була мова, яка б забезпечувала зручну роботу з нею. Для роботи з PostgreSQL Python пропонує зручні бібліотеки такі як psycopg2 та SQLAlchemy, що значно пришвидшує розробку.

Для написання програмного додатку мовою Python було обрано середовище розробки Pycharm, оскільки вона забезпечує зручну організацію проектів, рефакторинг та автокомпліт. Крім цього Pycharm контролює за тим, щоб написаний код був читабельним та зрозумілим. Він слідкує за дотриманням всіх правил написання коду мовою програмування Python. Також він автоматично налаштовує віртуальне середовище та забезпечує можливість зручного встановлення пакетів в проект.

Структура бази даних

(Див. додаток 1)

Перелік сутностей:

1. Сутність “Students” відповідає за зберігання особистої інформації, яка стосується студентів, які навчаються в університеті.
2. Сутність “Teachers” відповідає за зберігання особистої інформації, яка стосується викладачів, які навчаються в університеті.
3. Сутність “Subjects” відповідає за зберігання інформації про предмети, які викладаються в університеті.
4. Сутність “Scores” відповідає за зберігання інформації про оцінки з певних предметів, які отримали студенти.
5. Сутність “Faculties” відповідає за зберігання інформації, яка стосується факультетів, які знаходяться в університеті.
6. Сутність “Departments” відповідає за зберігання інформації, яка стосується спеціальностей, які знаходяться в університеті.
7. Сутність “Fac_der” відповідає за зберігання інформації, яка стосується навчальних програм, які викладаються в університеті.

Опис програмного забезпечення

Загальна структура програмного забезпечення

Програмний додаток розроблений згідно шаблону MVC (тобто логіка розділена на три частини: model, view і controller). Також в курсовій роботі створено скрепер для зчитування предметів з сайту.

<https://student.apps.utah.edu/uofu/stu/ClassSchedules/main/1204/>

Крім цього в курсовій роботі збережений код для зчитування датасетів з сайту kaggle.com University Dataset і USA Name Data та створення графіків ефективності індексації.

Опис модулів програмного забезпечення

Програмний додаток, який знаходиться в директорії project складається з таких модулів:

main.py - передає отримані запити від view.py до database.py, отримує відповідь, яку обробляє та надсилає результат до view.py.

database.py - реалізує взаємодію з базою даних та передає отримані результати в main.py.

view.py - реалізує взаємодію з користувачем, тобто отримує дані, передає дані до main.py та виводить дані з нього.

models.py - зберігає інформацію про ORM моделі бази даних.

Скрепер, який знаходиться в директорії subjects складається з таких модулів:

links.py - “павук” для збору всіх посилань, які містять предмети.

subjects.py - “павук” для збору предметів з зібраних посилань.

Опис основних алгоритмів роботи

Програмний додаток реалізує такі основні алгоритми роботи:

- Різноманітні операції CRUD над усіма таблицями бази даних(get by id, get by name, delete, update, insert). (Див. додаток 2)
- Фільтрація запитів за визначеними користувачем критеріями. (Див. додаток 3)
- Генерація всіх таблиць в базі даних.(Див. додаток 4)
- Статистичний аналіз даних за допомогою pandas з наступним виведенням його у вигляді графіків за допомогою pyplot. (Див. додаток 5)

Аналіз функціонування засобів реплікації

Для забезпечення безвідмовної роботи додатку та розподілення навантаження на базу даних між декількома серверами був використаний засіб реплікації master-slave. Через відсутність додаткової машини та слабкої потужності машини, на якій був створений програмний додаток (недостатня потужність для забезпечення другого сервера на віртуальній машині) другий сервер було створено на цій же машині. При зупинці сервера master роль master бере на себе slave сервер (Див. додаток 6)

Аналіз функціонування засобів резервування/відновлення

Резервування було реалізовано за допомогою `pg_dump`. Він використовується для збереження певної бази даних в окремому файлі з розширенням `.dump`. Резервування відбувається приблизно 2 секунди, та створений файл має розмір 50,2 МБ.

Можливе резервування з іншим розширенням, але ми використали саме це розширення в курсовій роботі. (Див. додаток 7)

Також ми додали резервування до регулярних команд, тобто резервування виконується раз в день, щоб унеможливити незворотню втрату даних. (Див. додаток 8)

Аналіз результатів підвищення швидкодії запитів

Був створений індекс Btree в таблиці scores для поля score та був виконан 100 разів запит “explain analyze select score from scores where score<40” з індексом та без індекса. (Див. додаток 9) На графіку видно, що індекси допомагають значно пришвидшити роботу запитів.

Також ми спробували реалізувати хеш-індексацію на таблиці students для поля name та запустили 100 разів запит “explain analyze select name from students where name='Mary Knox'”(Див. додаток 10) Дана індексація досить сильно пришвидшила роботу, але зважаючи на те, що хеш-функцію можливо використовувати лише з оператором “=” та небезпеку цілісної роботи бази даних ми спробували для такого ж запиту використати індексування Btree. (Див. додаток 11) Результати запитів з різними індексуваннями майже не відрізняються. Тож, було вирішено залишити Btree.

Опис результатів аналізу предметної галузі

Деякі приклади виконання операцій CRUD над базою даних:

- Find by id (Див. додаток 12)
- Find by name (Див. додаток 13)
- Find all (Див. додаток 14)
- Insert (Див. додаток 15)
- Update (Див. додаток 16)
- Delete (Див. додаток 17)

Приклад фільтрації даних з веденням значень фільтрації користувачем (Див. додаток 18)

Приклад генерації (Див. додаток 19). Також генерація була проведена з датасетів, бібліотеки Faker та зібраних даних за допомогою скрепера.

Приклад статичного аналізу даних з наступним виведенням результату у вигляді графіку (Див. додаток 20)

Висновки

Отже, був розроблений програмний додаток, який реалізує основну взаємодію між створеною нами базою даних університету. В програмному додатку наявні всі можливі операції CRUD над усіма таблицями. Крім того наявна можливість фільтрації даних за допомогою попередньо введених користувачем значень фільтрації.

Також в програмі було використано декілька видів генерації, таких як внесення датасетів з kaggle.com у базу даних, створення власного датасету “subjects” за допомогою Scrapy(для якого було використано два “павуки”), використання бібліотеки Faker, та розроблена можливість генерації всіх таблиць мовою SQL.

Для підтримки безвідмовної роботи програми та розподілення навантаження була створена реплікація master-slave, яка розміщена на цьому ж пристрої як другий сервер. При відмові сервера master slave бере роль master на себе та обробляє всі запити.

Крім цього додатково регулярно системою виконується бекап нашої бази даних командою pg_dump у окремий файл, щоб унеможливити втрату даних.

Також було створено декілька індексів та було розглянуто їх швидкодія на 100 запитах мовою SQL, щоб переконатись, що створені індекси і справді пришвидшують роботу програми.

Статистичний аналіз різних даних(середня ціна на спеціальності; оцінки студента; кількість викладачів з певною позицією; кількість предметів, які викладачі викладають на певній спеціальності та на певному факультеті) був проведений за допомогою бібліотеки pandas.

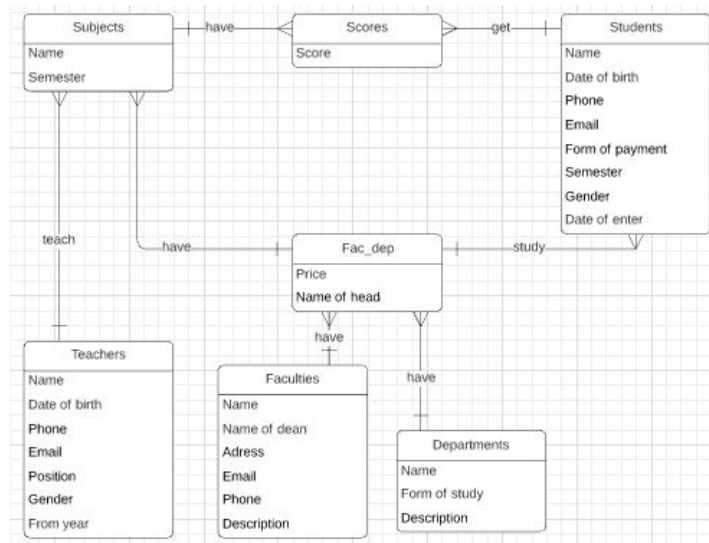
Дані про ефективність індексів та результати статистичного аналізу даних виводяться за допомогою бібліотеки pyplot у різні види графіків.

Література

1. <https://docs.scrapy.org/en/latest/>
2. <https://pandas.pydata.org/docs/>
3. <https://matplotlib.org/3.1.1/tutorials/introductory/pyplot.html>
4. <https://www.kaggle.com/ananta/student-performance-dataset>
5. <https://www.kaggle.com/datagov/usa-names>

Додатки

Додаток №1



ER-діаграма

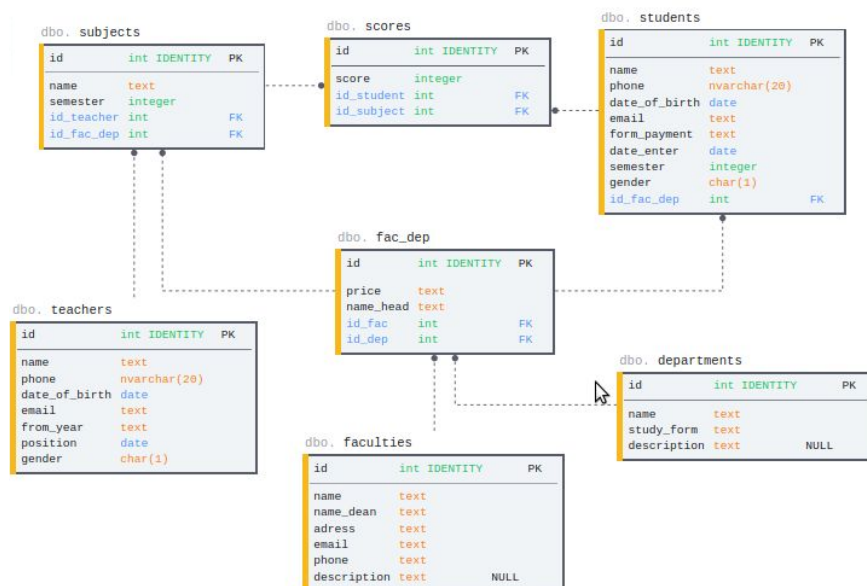


Схема бази даних

Додаток №2

1. Find a student by name	27. Add a fac_dep
2. Find a student by id	28. Add a faculty
3. Find a teacher by name	29. Add a department
4. Find a teacher by id	30. Update a student
5. Find a subject by name	31. Update a teacher
6. Find a subject by id	32. Update a subject
7. Find a faculty by name	33. Update a score
8. Find a faculty by id	34. Update a fac_dep
9. Find a department by name	35. Update a faculty
10. Find a department by id	36. Update a department
11. Find a faculty_department by id	37. Remove a student
12. Find a score by id	38. Remove a teacher
13. Find student's scores	39. Remove a subject
14. Find teacher's subjects	40. Remove a score
15. Find student's subjects	41. Remove a fac_dep
16. Find all students	42. Remove a faculty
17. Find all teachers	43. Remove a department
18. Find all subjects	
19. Find all scores	
20. Find all faculty_department	
21. Find all faculties	
22. Find all departments	
23. Add a student	
24. Add a teacher	
25. Add a subject	
26. Add a score	

Додаток №3


```

44. Find faculty_department and study_form where faculty and name like some text and price less some number
45. Find student where name like some text, semester is some number and min score less than some number
46. Find teacher, faculty, subject where faculty, teacher and subject like some text
47. Find faculty, subject, average score where faculty and subject like some text and average score is between some numbers
48. Find student's subjects for current semester where faculty and department name is some text, student name like and current semester is

```

Додаток №4

```

49. Generate faculty_department
50. Generate scores
51. Generate subjects
52. Generate students
53. Generate teachers
54. Generate faculties
55. Generate departments

```

Додаток №5

```

56. Average price of departments
57. Student scores and subjects
58. Count teachers's subjects on faculty
59. Count teachers on positions on faculty

```

Додаток №6

```

prompt] [-t timeout] [-u user] file ...
polina@pokaori:~$ sudo -u postgres psql
[sudo] password for polina:
psql (12.2 (Ubuntu 12.2-4), server 12.4 (Ubuntu 12.4-0ubuntu0.20.04.1))
Type "help" for help.

postgres=# select client_addr, state from pg_stat_replication;
 client_addr | state
-----+-----
 127.0.0.1   | streaming
(1 row)

```

Під'єднаний slave сервер

```

Enter your choice:1
Enter name:Mary Knox
<Student (id=2, name='
Find is successful

```

Коли два сервери запущені

```

Enter your choice:1
Enter name:Mary Knox
Go to slave server
<Student (id=2, name=
Find is successful

```

Коли slave бере роль master

```

def find_student_by_id(self, id):
    student = None
    try:
        student=self.s.query(Student).get(id)
        self.s.commit()
    except exc.OperationalError:
        print("Go to slave server")
        student=self.s2.query(Student).get(id)
        self.s2.commit()
    except (Exception, exc.SQLAlchemyError) as error:
        self.s.rollback()
        self.s2.rollback()
        print("Error in find_student_by_id():", error)
    if student:
        return student
    else:
        return "Can't find student by id"

```

Приклад коду

Додаток №7

```

polina@pokaori:~$ sudo -u postgres pg_dump Course_work > ~/backup/course_work.dump
[sudo] password for polina:

```

Додаток №8

```

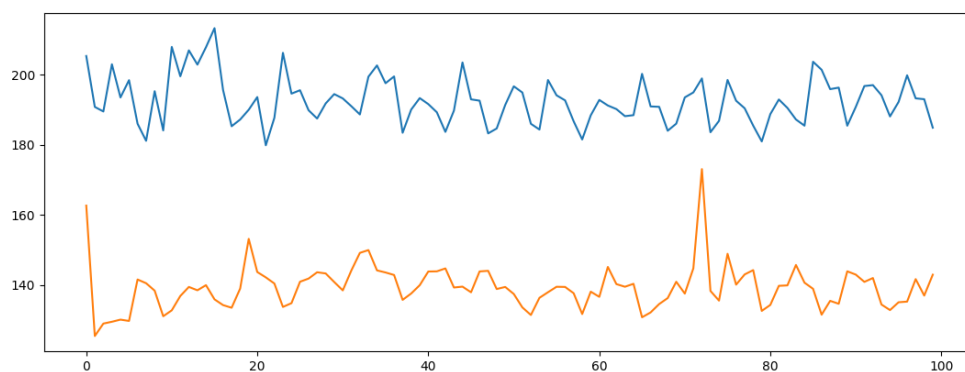
GNU nano 4.8 /etc/crontab
# that none of the other crontabs do.

SHELL=/bin/sh
PATH=/usr/local/sbin:/usr/local/bin:/sbin:/bin:/usr/sbin:/usr/bin

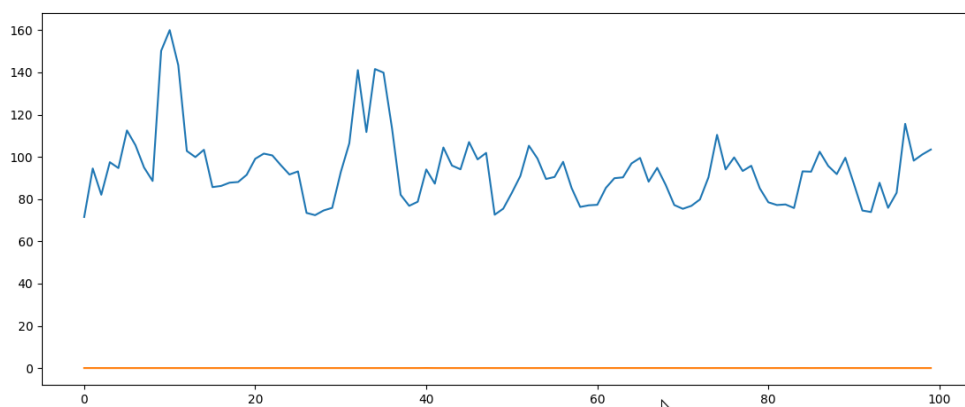
# Example of job definition:
# .----- minute (0 - 59)
# | .----- hour (0 - 23)
# | | .----- day of month (1 - 31)
# | | | .----- month (1 - 12) OR jan,feb,mar,apr ...
# | | | | .---- day of week (0 - 6) (Sunday=0 or 7) OR sun,mon,tue,wed,thu,fri,sat
# | | | | |
# * * * * * user-name command to be executed
17 * * * * root cd / && run-parts --report /etc/cron.hourly
25 6 * * * root test -x /usr/sbin/anacron || ( cd / && run-parts --report /etc/cron.daily )
47 6 * * 7 root test -x /usr/sbin/anacron || ( cd / && run-parts --report /etc/cron.weekly )
52 6 1 * * root test -x /usr/sbin/anacron || ( cd / && run-parts --report /etc/cron.monthly )
50 20 * * * polina sudo -u postgres pg_dump Course_work > ~/backup/course_work.dump

```

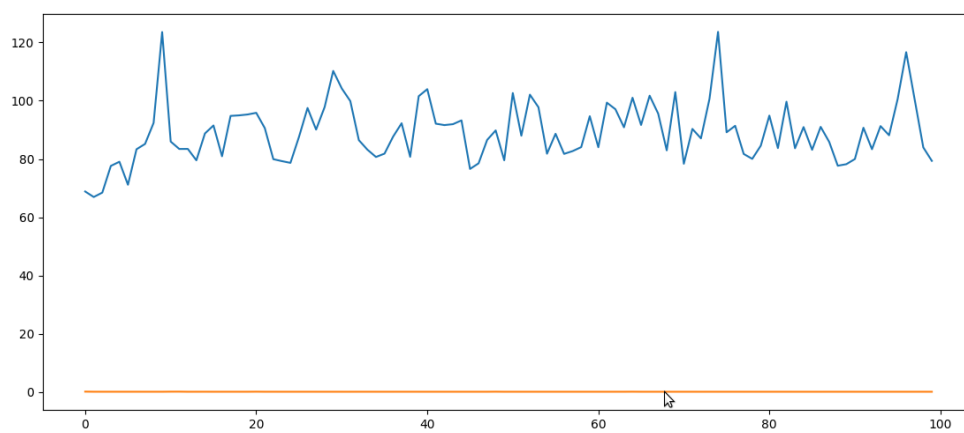
Додаток № 9



Додаток №10



Додаток №11



Додаток №12

```
Enter your choice:2
Enter id:2
<Student (id=2, name='Mary Knox', date_of_birth='1996-01-23', date_enter ='2019-12-29', phone='+114309918', email='robertfrazier@robinson.com',form_payment='Bu
Find is successful
```

Додаток №13

```
Enter your choice:1
Enter name:Mary Knox
I
<Student (id=2, name='Mary Knox', date_of_birth='1996-01-23', date_enter ='2019-12-29', phone='+114309918', email='robertfrazier@robinson.com',form_payment='Bu
Find is successful
```

Додаток №14

```
<Student (id=45885, names='Elaine Pearson', date_of_birth='2002-02-28', date_enter ='2017-01-06', phone='+951440656', email='petersstephanie@kennedy-bell.org',f
<Student (id=45886, names='Helen Garcia', date_of_birth='1997-01-08', date_enter ='2014-11-23', phone='+646692271', email='curtis13@stark.com',form_payment='Con
<Student (id=45887, names='Alice Martinez', date_of_birth='1994-05-07', date_enter ='2018-02-26', phone='+942311986', email='lrogers@gmail.com',form_payment='Bu
<Student (id=45888, names='Joanne Powell', date_of_birth='2002-01-31', date_enter ='2017-02-27', phone='+884912690', email='blakewilliam@gmail.com',form_payment
<Student (id=45889, names='Roberta Bauer', date_of_birth='2002-02-23', date_enter ='2017-07-04', phone='+204143250', email='michaelortiz@good.biz',form_payment=
<Student (id=45890, names='Melanie Day', date_of_birth='1990-07-01', date_enter ='2016-04-26', phone='+317931710', email='kylehill@mclean.com',form_payment='Bud
<Student (id=45891, names='Joan Weiss', date_of_birth='1998-01-08', date_enter ='2018-11-20', phone='+817383966', email='tracywhite@yahoo.com',form_payment='Con
<Student (id=45892, names='Irene Lawrence', date_of_birth='2000-02-27', date_enter ='2015-09-11', phone='+627040379', email='ashlee01@gmail.com',form_payment='B
<Student (id=45893, names='Norma Coleman', date_of_birth='1992-12-29', date_enter ='2012-08-08', phone='+805236832', email='stewartjuan@yahoo.com',form_payment=
<Student (id=45894, names='Holly Taylor', date_of_birth='1999-05-04', date_enter ='2018-03-23', plljone='+552343121', email='zwatson@ramsey-cruz.info',form_paymen
<Student (id=45895, names='Penny Gregory', date_of_birth='1998-11-15', date_enter ='2015-08-18', phone='+816531783', email='travisgreen@yahoo.com',form_payment=
<Student (id=45896, names='Rita Fields', date_of_birth='1990-11-03', date_enter ='2006-02-20', phone='+425420437', email='juliejones@hotmail.com',form_payment='
<Student (id=45897, names='Jackie Coleman', date_of_birth='2000-11-29', date_enter ='2016-12-23', phone='+162443822', email='bettyfrederick@gmail.com',form_paym
<Student (id=45898, names='Yvette Simpson', date_of_birth='2000-10-01', date_enter ='2015-02-19', phone='+346244450', email='kingalexander@hotmail.com',form_pay
<Student (id=45899, names='Beth Kramer', date_of_birth='1993-09-12', date_enter ='2011-11-04', phone='+395863462', email='sarah37@hotmail.com',form_payment='Bud
<Student (id=45900, names='Pam Norris', date_of_birth='1991-10-31', date_enter ='2009-09-26', phone='+455706395', email='josephlopez@hotmail.com',form_payment='
<Student (id=45901, names='Carmen Ramirez', date_of_birth='1995-07-10', date_enter ='2017-10-14', phone='+800654077', email='robertsimmons@hotmail.com',form_pay
```

Додаток №15

```
Enter your choice:29
Enter name:Computer Science
Enter form of payment Daytime or Extramural:Daytime
Enter description: Just for example
Add is successful
Select an option
```

Додаток №16

```
Enter your choice:36
Enter id:33
Enter name or nothing:Computer
Enter form of payment Daytime or Extramural or nothing:
Enter description or nothing:Hello
```

Додаток №17

```
Enter your choice:43
Enter id:84
Delete is successful
```

Додаток №18

```

Enter your choice:44
Enter price :100000
Enter faculty name:Math
Enter department name:Com
Faculty: Physics and Mathematics
Department: Computer Centre (CC)
Study form: Daytime

Faculty: Applied Mathematics
Department: Computer Science & Engineering
Study form: Extramural

Find is successful

```

Додаток №19

```

Enter your choice:50
Enter number for generate:100000
Generated

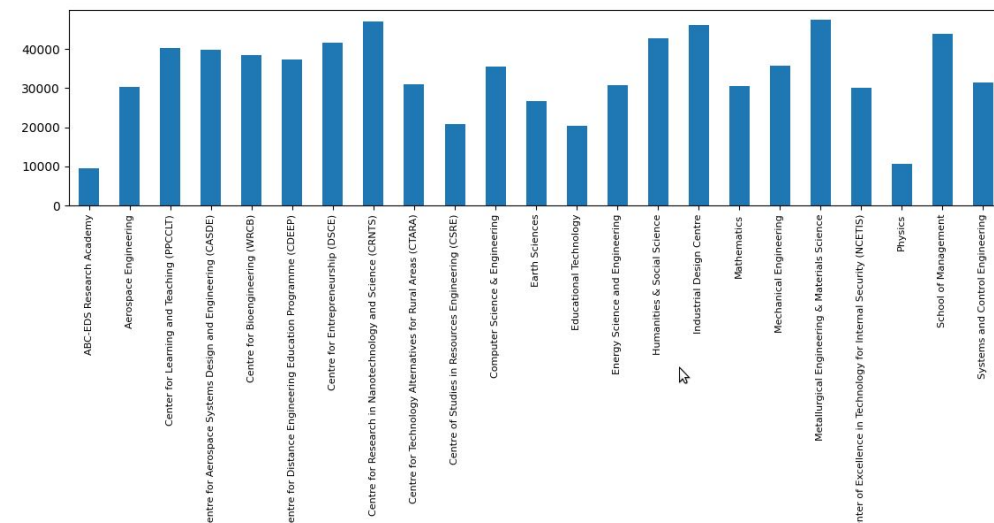
```

Додаток №20

```

Enter your choice:56
Enter form of payment Daytime or Extramural:Extramural

```



Середня ціна на спеціальності