компанія функціонує у понад 80 країнах світу), в різних інтерпретаціях (скорочення, абревіатури, закінчення тощо) та за різними рівнями спеціалізації. Далі відповідальна особа приводить усі назви до єдиного стандарту, який узгоджено з бізнесом та відповідає цілям промоційної компанії. Вже з формалізованих даних за допомогою процедур та алгоритмів ETL формуються вітрини даних та складаються до сховища даних. Вітрини можуть бути підключені до будь-якої системи звітності що використовує фармацевтична компанія у бізнес-аналітиці: Power BI, Excel, Qlik View тощо.

Обраний підхід щодо проектування архітектури бізнес-аналітики фармацевтичної компанії базується на принципі SSOT (Single source of truth), який передбачає формування єдиного джерела отримання даних для всіх звітів компанії задля уникненні відхилень і розбіжностей. Така практика структурування інформаційних моделей та пов'язаних із ними схем даних реалізована таким чином, що кожен елемент даних зберігається або редагується лише в одному місці. Будь-які можливі зв'язки з цим елементом даних (в інших областях реляційної схеми або навіть у віддалених об'єднаних базах даних) є лише за посиланням. Оскільки всі інші розташування даних просто посилаються на первинне розташування "джерела істини", оновлення елемента даних у первинному розташуванні поширюються по всій системі без риску появи дубліката.

Висновок. Для проектування архітектури бізнес-аналітики фармацевтичної компанії було обрано переважно продукти Microsoft з дотриманням принципів масштабованості системи, формалізації даних та єдиного джерела істини, що дозволило оптимізувати як структури даних оперативного зберігання для виконання операцій введення, модифікації, знищення та пошуку, так і структури даних, що використовуються для аналізу. Оптимально продумана архітектура бізнес-аналітики допомагає фармацевтичним компаніям зменшити витрати, збільшити дохід, максимізувати цінність та надійність інформації.

Список використаної літератури

1. Edoardo Gori, Big Data Deployment Model for the Architecture's Optimization: A Pharma Manufacturing Business Case. - M.Sc. Management of Technology – Information Systems ESIEE Paris, 2019. – 64 p.: https://www.researchgate.net/publication/332470397 (27.04.2021)

2. BI solution architecture in the Center of Excellence, 11/11/2020: https://docs.microsoft.com/en-us/power-bi/guidance/center-of-excellence-business-intelligence-solution-architecture

3. Внутрішня документація з проектування бізнес-аналітики компанії Acino

УДК 004.421

Захарова К.В., *студентка 3 курсу спеціальності 122 «Комп'ютерні науки»*
Ніколюк П.К., *професор кафедри комп'ютерних наук та інформаційних технологій*

ПРОГРАМНИЙ ДОДАТОК «ДЕКАНАТ»

Донецький національний університет імені Василя Стуса, м. Вінниця

Про важливість розкладу сказано достатньо. Чіткий графік допомагає студентам планувати свій час, щоб більше встигати, виховує послідовність, що позитивно впливає на працьовитість.

За словами вчених, при дотриманні розпорядку в організмі виробляються умовні рефлекси: кожна попередня діяльність стає сигналом до наступної, це дозволяє легко перемикатися з одного заняття на інше. Крім того, так дотримується баланс між робочою діяльністю і відпочинком [1].

У чому полягає проблема мануального складання розкладу?

По-перше, це час. Особливо це відчувається, коли навчальний семестр ось-ось має початися, а розклад ще не готовий. Таким чином витрачається ще більше годин – кожен тиждень/два – новий розклад. А студенти не можуть розрахувати свій час та плани наперед й мають швидко зорієнтуватися та підстроїтися під новий розклад.

По-друге, банальна людська неуважність. Люди – не машини, важко тримати в пам’яті усі змінні – кількість студентів у групі, місткість аудиторій та багато іншого. Через це лекції проходять в малих аудиторіях чи без належного обладнання, що доставляє незручності як і викладачам, так і студентам.

По-третє, недотримання вимог та прохань викладачів і студентів. Часто виникають вікна, які нікого не приваблюють.

Для пояснення алгоритму, потрібно привести числа, що використовуються:

- В 1-ому семестрі:
  - 16 тижнів;
  - 640 пар (16 тижнів х 5 днів/тиждень х 8 пар/день);
- 20 – максимальна кількість пар на тиждень (5 днів х 4 пари/день).

Реалізація алгоритму у вигляді псевдокоду із прикладом:

На лекції з предмету_1 надається 40 годин та 20 годин на лабораторні.

Ліміт на тиждень (загальне число):

$$Lim = \lceil \frac{20}{16} \rceil + \lceil \frac{10}{16} \rceil = 2 + 1 = 3$$

```
for тиждень з 16 тижнів:
    lim = Lim
    for день з 5 днів (Пн, Вт, Ср, Чт, Пт):
        for пара з 8 пар:
  Якщо lim != 0:
      Якщо пара вільна:
                Додати лекцію до розкладу,
          lim - 1, кількість_лекцій - 1, пара + 1;
          Якщо кількість_пар / 16 >= 1 та lim != 0:
```

Додати лекцію в наступний день,
lim - 1, кількість_лекцій - 1, пара = 8;
Якщо є лабораторні та lim != 0:
додати лабораторну наступною парою,
lim - 1, кількість_лабораторних - 1;
Інакше: Lim - 1
Якщо є лабораторні та lim != 0:
Додати лабораторну наступною парою,
lim - 1, кількість_лабораторних - 1:
Якщо lim == 0:
Пропускаємо 2-ий та 3-ій цикли;

Ця частина алгоритму створює рівномірно-спадаюче навантаження на тиждень та на семестр.

Даний алгоритм має застосовуватися до усіх груп. Кожна група має «свій ідентифікатор», тобто день із найвищою завантаженістю, що відрізняється від інших груп потоку. Відповідно, для першої групи, найбільша кількість пар у понеділок, а п'ятниця може взагалі бути вільною; для другої – вівторок, а понеділок – вихідний. Але для всіх однаковий розподіл навантаження протягом семестру – початок року зі збільшеною кількістю пар, із рівномірним зменшенням кількості до початку сесії.

Для написання функціоналу використовується мова програмування Python 3.7 [2]. Аргументовано тим, що дана мова має простий і зрозумілий синтаксис та надає усі необхідні бібліотеки для створення повноцінного додатку.

Для створення графічного інтерфейсу – PyQt5 [3]. Додаток Qt Creator дозволяє швидко створити будь-який дизайн та автоматично переводить його до коду.

Використана база даних – MySQL [4]. Дана база даних відрізняється від інших аналогів своєю гнучкістю, швидкістю та масштабованістю.

Отже, після створення програмного додатку «Деканат» очікується більш зручний розклад пар, рівномірна завантаженість, оптимальне використання часу як викладачів, так і студентів. В свою чергу, працівники деканату, хто витрачає велику кількість часу на складання розкладу, отримають більше часу на інші, не менш важливі, завдання.

Список використаної літератури

1. *URL: https://en.wikipedia.org/wiki/Classical_conditioning Wikipedia. Classical conditioning (дата звернення: 24.04.2021)*
2. *URL: https://www.python.org/downloads/release/python-370/ Python 3.7.0 (дата звернення: 24.04.2021)*
3. *URL: https://pypi.org/project/PyQt5/ PyQt5 (дата звернення: 24.04.2021)*
4. *URL: https://www.mysql.com/ MySQL (дата звернення: 24.04.2021)*