

<https://doi.org/10.31891/2307-5732-2025-357-1>

УДК 004.4:004.65:004.9

**АНТОНОВ ЮРІЙ**

Донецький національний університет імені Василя Стуса

<https://orcid.org/0000-0001-9285-2988>

e-mail: [y.s.antonov.edu@gmail.com](mailto:y.s.antonov.edu@gmail.com)

## РЕАЛІЗАЦІЯ ПІДСИСТЕМИ КЕРУВАННЯ РОЗКЛАДОМ УНІВЕРСИТЕТУ НА ОСНОВІ ЗМІШАНОЇ АРХІТЕКТУРИ

*Розроблено систему керування розкладом університету на основі клієнт/серверної архітектури та архітектури трирівневих баз даних. Запропонована система складається з таких компонентів: Schedule Client, SSH Server, A.S.T.S. v3 Database, Schedule Database, Schedule Site and Service, Schedule WebService. Для синхронізації даних між A.S.T.S. v3 Database та Schedule Database використовується механізм реплікації master/slave. Розглянута система керування розкладом вищого навчального закладу експлуатувалась на факультеті математики та інформаційних технологій з 2015 по 2019 навчальні роки та у 2022-2025 в обмеженому режимі на кафедрі інформаційних технологій. Студенти та викладачі мали можливість отримувати доступ до актуального розкладу навчальних занять у режимі реального часу за допомогою браузера.*

**Ключові слова:** реляційна база даних, CAP теорема, керування розкладом, клієнт/серверна архітектура, архітектура трирівневих баз даних, SSH тунелі.

**ANTONOV YURIY**

Vasyl' Stus Donetsk National University

## UNIVERSITY SCHEDULE CONTROL SUBSYSTEM IMPLEMENTATION BASED ON MIXED ARCHITECTURE

*The schedule analysis on the websites was conducted for more than 60 universities. It has been established that the class schedule is provided to the educational process participants in the form of Excel, Word, PDF files, static and dynamic web pages and a personal cabinet with mandatory authorization. The university schedule management system (Schedule module for Antonov Students Test System) was developed based on the client / server architecture and the architecture of three-level databases. The proposed system consists of such components as SSH Server, A.S.T.S. v3 Database, Schedule Database, Schedule Site and Service, Schedule WebService and Schedule Client. The CAP theorem was used to justify the type of Database Management System. The total number of relational tables in A.S.T.S. v3 Database is over 150. This DB consists of the following components: Core, HRDepartment, Testing, WebGUI, Reports, GradeBook, StudWorks, TimeTableClasses, SocialForm, TransferRestore, EDBO plugins, ScientificDirector, ScientificProfile, Admissions Committee. The Schedule Database consists of two components: Core and TimeTableClasses, and a total has 48 relational tables. The master/slave replication mechanism was used to synchronize data between A.S.T.S. v3 Database and Schedule Database. Access to the A.S.T.S. v3 Database server was implemented using SSH tunnels based on personal keys (RSA or newer). JSON format and Web API can be used for interaction with external systems or services. The higher education institution's schedule management system considered in the work was operated at the Faculty of Mathematics and Information Technologies from 2015 to 2019 academic years and in 2022-2025 in limited mode at the IT Department. Students and teachers have the opportunity to access the current schedule of classes in real time using a browser.*

**Keywords:** relational database, CAP theorem, schedule management, client/server architecture, three-tier database architecture, SSH tunnels.

Стаття надійшла до редакції / Received 21.07.2025

Прийнята до друку / Accepted 15.08.2025

### Вступ

Проблема автоматизації різних видів діяльності, що супроводжують навчальний процес залишається актуальною на протязі багатьох років. Не є винятком і процес керування розкладом у вищому навчальному закладі. Аналіз понад 60 сайтів закладів вищої освіти в Україні показав, що частина навчальних закладів продовжує викладати розклад на офіційних ресурсах у вигляді: файлів Excel (17 сайтів); файлів Word (5 сайтів); файлів PDF (22 сайти); статичних веб сторінок (2 сайти); динамічних веб сторінок (14 сайтів); особистого кабінету з обов'язковою авторизацією (4 сайти). Окрім цього, попередній аналіз показав, що 39 закладів вищої освіти зберігають та обробляють інформацію пов'язану з розкладом виключно за принципом верхній/нижній тиждень, і лише 14 за принципом календаря. Результати даного аналізу було зібрано та систематизовано автором у вигляді таблиці на gitlab репозиторії [1].

### Постановка проблеми у загальному вигляді

#### та її зв'язок із важливими науковими чи практичними завданнями

Відсутність автоматизації під час створення розкладу у вищому навчальному закладі, може призводити до різноманітних проблем, такі як:

- в аудиторії заплановано декілька різних занять в один й той самий час;
- викладач має проводити заняття в різних аудиторіях одночасно;
- в аудиторії буде знаходитись більше студентів, ніж це можливо технічно;
- учасники навчального процесу дізнаються про зміни у розкладі із запізненням;
- на офіційних ресурсах навчального закладу знаходиться не актуальний розклад;
- відсутні можливості для визначення зайнятості тієї чи іншої аудиторії для широкого кола

користувачів;

- відсутність можливості інтеграції/синхронізації розкладу з особистим або корпоративним календарем (Google Calendar, MS Calendar).

Окремо необхідно відзначити, що у навчальній діяльності будь-якого закладу вищої освіти присутні види діяльності, які відбуваються на не регулярній основі, але все одно потребують обліку та керування аудиторним фондом, наприклад:

- консультації на протязі семестру;
- розклад заліків, іспитів, та консультацій перед іспитами;
- захист кваліфікаційних робіт;
- захист дисертацій (PhD або докторів наук);
- робота гуртків;
- засідання вчених рад, КТК, тощо.

Тому, будь-яка сучасна система для роботи з розкладом має враховувати такі нерегулярні події разом з лекційними, лабораторними, практичними заняттями.

#### Аналіз досліджень та публікацій

Усі роботи, що присвячено проблемі автоматизації керування розкладом можна поділити на 2 категорії, а саме:

- системи, що дозволяють автоматизувати процес керування розкладом, при цьому головне місце у створенні розкладу відводиться людині [2];
- системи, що дозволяють автоматично генерувати один або декілька варіантів розкладу [3, 4].

Так, у роботі [2] авторами пропонується програмне рішення на основі платформи 1С Підприємство. Система у вигляді декількох підсистем, які взаємодіють з однією БД, створеною за допомогою Microsoft SQL Server, запропоновано у роботі [5]. Алгоритми, що можуть бути використані для автоматичної генерації розкладу, запропоновані у роботах [3, 4]. У роботі [6] автори враховують час, який потрібен студентам для переміщення з однієї аудиторії в іншу.

Серед закордонних розробок особливе місце займає відкрита система UniTime (University Timetabling) [7], що розробляється спільнотою розробників за ліцензією Apache License Version 2.0 та має можливість створювати розклад автоматично за допомогою Constraint Solver Library (GNU LGPL). Наукові результати пов'язані з системою UniTime (University Timetabling) висвітлено у роботах [8, 9, 10].

#### Формулювання цілей статті

Метою роботи є проектування системи керування розкладом на основі декількох архітектурних підходів, яка дозволить задовольнити потреби в актуальному розкладі викладачів, студентів та різних підрозділів навчального закладу (ЗВО), буде надійною та безпечною.

#### Виклад основного матеріалу

Проектування будь-якої інформаційної системи починається з визначення інформації, що буде зберігатися та оброблятися інформаційною системою, а також з правил обробки цієї інформації.

Відповідно до задачі, система має зберігати наступну інформацію:

**Аудиторний фонд** – номер аудиторії; корпус; поверх; кількість робочих місць; *площа*; адреса корпусу (місто, вулиця/проспект/бульвар, будинок), доступність аудиторії тому чи іншому підрозділу (доступна всім або назва підрозділу, дні тижня, пари, коли доступна).

**Викладач/Співробітник** – ПІБ; *дата народження; місце народження; вік*; стать; кафедра та факультет, на якому працює; посада; науковий ступінь; вчене звання; *стаж роботи*; чи є військовозобов'язаним; інформація про освіту; контактна інформація (телефон, пошта, месенджери, ...); телефон та пошта за замовчуванням; *паспорт (серія, номер, ким та коли виданий); адреса фактична/реєстрації (країна, місто, вулиця/проспект/бульвар, будинок, № квартира, область, район); сімейний стан.*

**Студент** – ПІБ; *дата народження; місце народження; вік*; стать; факультет; курс; група; спеціальність (повна та скорочена назви, шифр, галузь знань); форма навчання (очна, заочна, вечірня); ОКР (бакалавр, магістр, ...); *навчається за бюджетом або за контрактом*; кафедра; освітня програма за якою навчаються; чи є військовозобов'язаним; *інформація про попередню освіту*; контактна інформація (телефон, пошта, месенджери, ...); телефон та пошта за замовчуванням; *паспорт (серія, номер, ким та коли виданий); адреса фактична/реєстрації (країна, місто, вулиця/проспект/бульвар, будинок, № квартира, область, район); сімейний стан.*

**Освітня програма** – назва освітньої програми; спеціальність (повна та скорочена назви, шифр, галузь знань); ОКР (бакалавр, магістр, ...); термін навчання; дата початку та дата кінця (за наявності); кафедра що відповідає за дану освітню програму; *ПІБ гаранта та членів робочої групи*; перелік освітніх компонентів (назва дисципліни, кількість кредитів, № семестру, форма підсумкового контролю);

**Навчальний план** – освітня програма; навчальний рік; зміст навчального плану (назва дисципліни, семестр; кількість кредитів, вид діяльності (лекції/лабораторні/практичні/...) та кількість годин, кафедра що буде викладати); розподіл навчальних років за тижнями (семестр, сесія, канікули, практика, ...).

**Навчальне навантаження** – навчальний рік; семестр; факультет; кафедра; курс; група; кількість студентів; дисципліна; вид діяльності (тип заняття); кількість годин.

**Навчальне навантаження викладача** – ПІБ викладача; навчальний рік; семестр; факультет; кафедра; курс; кількість студентів; дисципліна; вид діяльності (тип заняття); кількість годин (план); кількість годин (факт).

**Розклад** – навчальний рік; факультет; курс; спеціальність; вид заняття (лекційне, лабораторне, практичне, консультація(семестр), консультація(іспит), залік, іспит, ...); № пари; ПІБ викладача та його посада; назва дисципліни; корпус та № аудиторії; назви груп/підгруп, що мають бути присутніми на парі.

**Зміст навчальної дисципліни** – назва; кількість кредитів; кількість змістових модулів; оцінювання (вид діяльності, кількість балів, № тижня).

**Військовооблікові дані** – ПІБ, статус (придатний, непридатний, знятий з обліку, ...); серія та номер військово обліково документу, тип документу (військовий квиток, тимчасове посвідчення, ...), ким виданий, коли виданий, термін дії (за наявності), назва та місце організації, де перебуває на обліку.

Необхідно зазначити, що курсивом виділена інформація, яка не стосується безпосередньо питання створення та керування розкладом, однак вона буде необхідна у разі створення повноцінної інформаційної системи.

Майбутня система керування розкладом університету має відповідати наступним вимогам:

- 1) в аудиторії неможливо запланувати декілька різних занять в один і той самий час;
- 2) викладач не може проводити заняття в різних аудиторіях одночасно;
- 3) в аудиторії не може знаходитись більше студентів, ніж це можливо технічно;
- 4) учасники навчального процесу оперативно повідомляються про зміни у розкладі;
- 5) на офіційних ресурсах навчального закладу знаходиться виключно актуальний розклад;
- 6) наявна можливість визначення зайнятості аудиторного фонду;
- 7) система забезпечує доступ до розкладу викладача/студента у вигляді динамічної веб сторінки;
- 8) розклад враховує термінологію “верхній/нижній тиждень” але зберігає його як календар, з урахуванням дати заняття, пари, та інших атрибутів;
- 9) під час додавання пар з навчальної дисципліни всі вони додаються лише у разі, якщо не порушується жодне з правил (у разі, якщо порушується хоча б одне якесь правило, то не додається увесь діапазон пар);
- 10) система забезпечує розмежування прав для різних користувачів, у тому числі на рівні бази даних;
- 11) на всіх етапах розробки враховується питання безпеки;
- 12) за потреби має бути можливість віддаленої роботи працівника, що не знаходиться на території навчального закладу;
- 13) синхронізація розкладу з особистим або корпоративним календарем (Google Calendar, MS Calendar ) за допомогою стороннього ПЗ;
- 14) отримання розкладу через Telegram/Viber за допомогою стороннього ПЗ.

На наступному етапі необхідно визначитись з архітектурою програмного комплексу, з типом системи управління базами даних (СУБД), а саме реляційна, чи NoSQL, та низкою інших задач.

Виходячи з описаних вище вимог та спираючись на аналіз предметної галузі, можна прийти до висновку, що

- одна сутність системи буде обслуговувати один навчальний заклад (можливо з філіями);
- немає потреби робити систему доступною в інших регіонах, країнах, континентах;
- враховуючи специфіку даних, з якими працює система, на основі теореми CAP робимо висновок, що наша система найбільше відповідає класу CA (Consistency Availability), що означає дотримання принципів узгодженості та доступності [11].

Враховуючи це, для основної бази даних (БД) найбільше підходить саме реляційна СУБД, а NoSQL СУБД краще підійдуть в якості допоміжного варіанту, наприклад для зберігання вхідної/вихідної документації у вигляді файлів [11, 12].

Так на рис. 1 зображено діаграму розгортання Schedule module for A.S.T.S [13].

Під час проектування системи (рис. 1) було вирішено, що

- система буде розташована на декількох серверах;
- усі сервери будуть знаходитись в окремій ізольованій мережі;
- система буде містити 2 сервери баз даних, а саме A.S.T.S. v3 Database та Schedule Database, між яким налаштовано режим реплікації Master/Slave.
- Серед складових системи необхідно виділити:
- *SSH Server*, за допомогою якого утворюється SSH Tunnel між авторизованим користувачем та сервером БД, за допомогою унікального (особистого) SSH ключа [14].
- *A.S.T.S. v3 Database* – повнофункціональна БД, з якою працюють співробітники, що змінюють інформацію пов'язану з розкладом або іншими процесами. Кожен, користувач використовує власний логін та пароль.
- *Schedule Database* – БД, що містить лише таблиці, які необхідні для оприлюднення розкладу на публічному сайті, та синхронізує усі зміни у даних завдяки реплікації.

- *Schedule Site and Service* – містить веб сайт, за допомогою якого користувачі та гості отримують доступ до актуального розкладу та сервіс (Web API), за допомогою якого сторонні програми можуть використовувати розклад (відображення у чат ботах месенджерів, синхронізація з календарем, тощо). Для з'єднання з БД використовуються користувач, який має право перегляду не конфіденційної (особистої інформації).
- *Schedule WebService* – веб сервіс, що дозволяє генерувати звіти у форматі xlsx.
- *Schedule Client* – кросплатформений десктопний додаток, що дозволяє працювати з базою авторизованим користувачам.

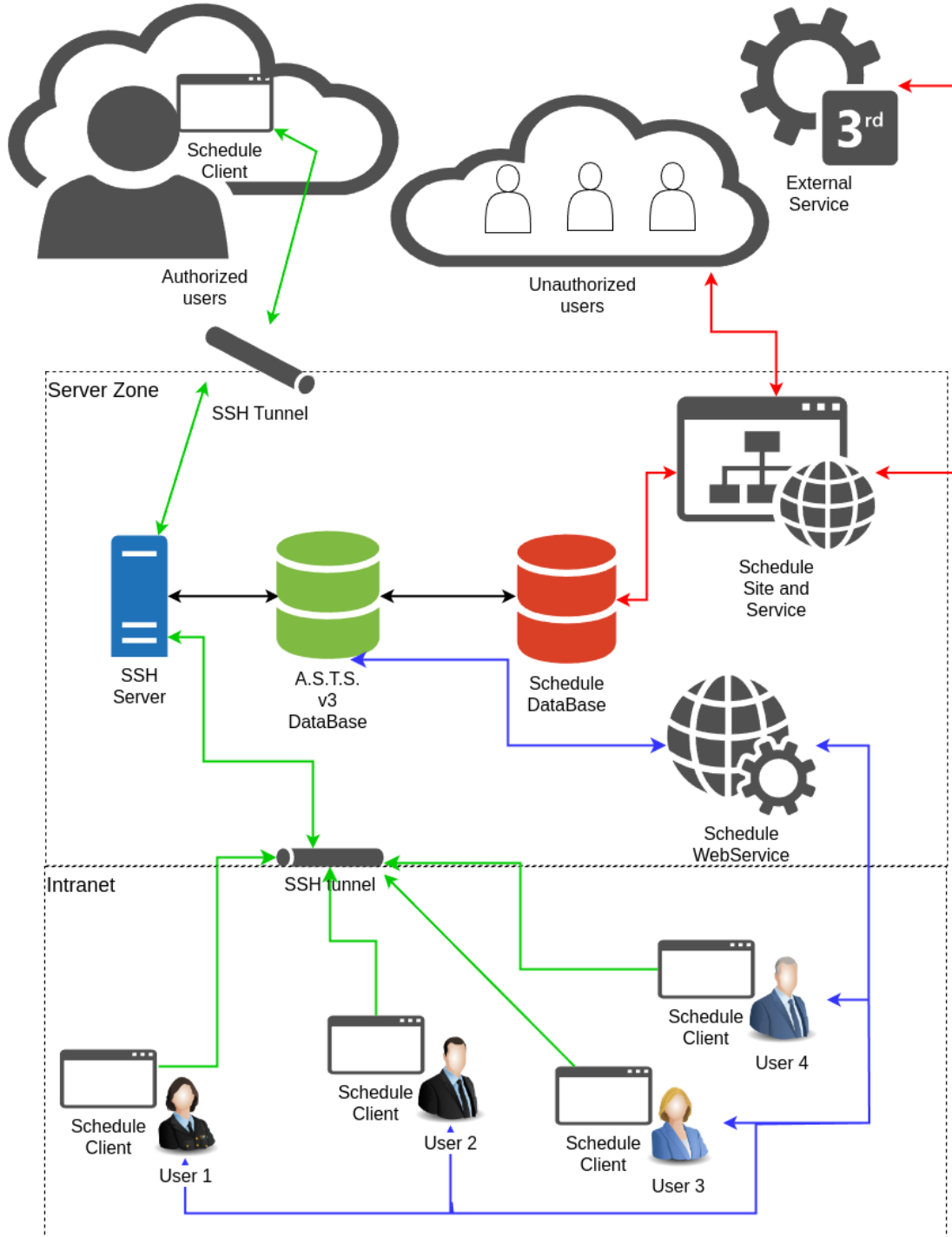


Рис. 1 Діаграма розгортання Schedule module for A.S.T.S

Загальна кількість реляційних таблиць у A.S.T.S. v3 Database становить понад 150 таблиць, та складається з наступних компонентів: Core, HRDepartment, Testing, WebGUI, Reports, GradeBook, StudWorks, TimeTableClasses, SocialForm, TransferRestore, EDBO plugins, ScientificDirector, ScientificProfile, Admissions Committee. База даних *Schedule Database* складається з двох компонентів Core та TimeTableClasses, та містить 48 таблиць. Під час створення БД усі таблиці зводились до 3-ї нормальної форми, а також до них застосовувались аномалії проектування з метою перевірки якості їх проектування. У зв'язку з необхідністю зберігання повної та скороченої назв сутностей, свідомо

робилась денормалізація (відповідність таблиць лише 2-й нормальній формі), для таблиць education\_form, education\_level, education\_speciality, university, university\_department.

Так на рис.2 - 4 зображено фрагменти бази даних A.S.T.S. v3 Database. Як видно з рис. 2 та рис. 3 інформація про людину (як фізичну особу) зберігається за допомогою 3-х таблиць, а саме person, person\_extended, military\_reservist, зв'язок між якими один до одного, та у налаштуваннях тригерів зовнішніх ключів містять ON DELETE CASCADE ON UPDATE CASCADE. Такий підхід виправдовує себе оскільки:

- у ряді випадків інформація, що зберігається у таблиці person\_extended є зайвою;
- не всі люди є військовозобов'язаними;
- під час реплікації на допоміжний сервер не передається конфіденційна інформація (таблиці country, document\_issued, family\_status, military\_ranks, military\_reservist, military\_reservist\_types, military\_specialities, person\_extended, person\_document\_types, person\_documents, where\_born відсутні у Schedule Database);
- спрощується процес керування доступом за допомогою команд GRANT / REVOKE / CREATE USER / CREATE ROLE / DROP USER / DROP ROLE
- спрощується процес додавання/редагування інформації різними підрозділами – наприклад, відділ кадрів відповідає за дані, що стосується освіти людини, її сімейного стану, наказів про звільнення та переведення і обробляє виключно її, з іншого боку, верифікацією та оновленням військово-облікових документів займається відповідний підрозділ.

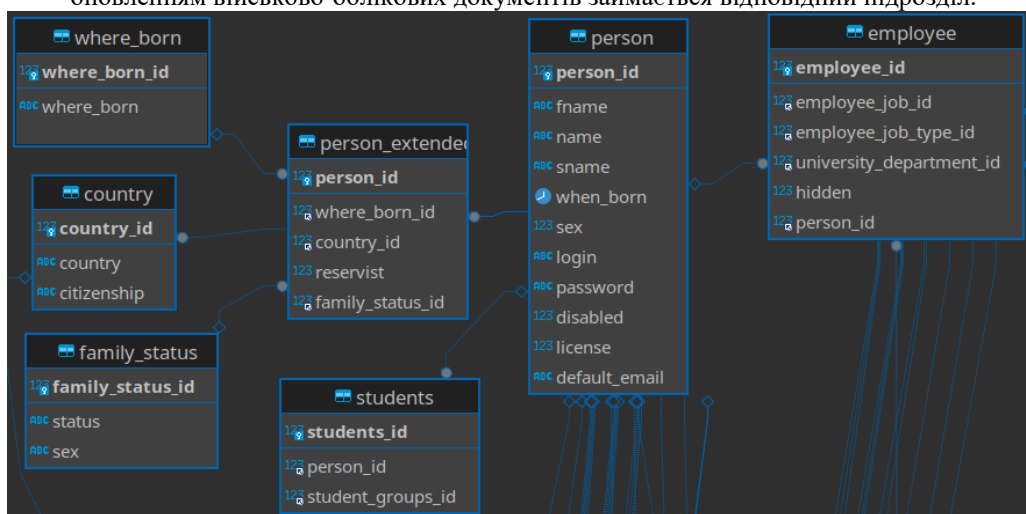


Рис. 2 Фрагмент БД, що ілюструє зв'язок між персонами та співробітниками і студентами

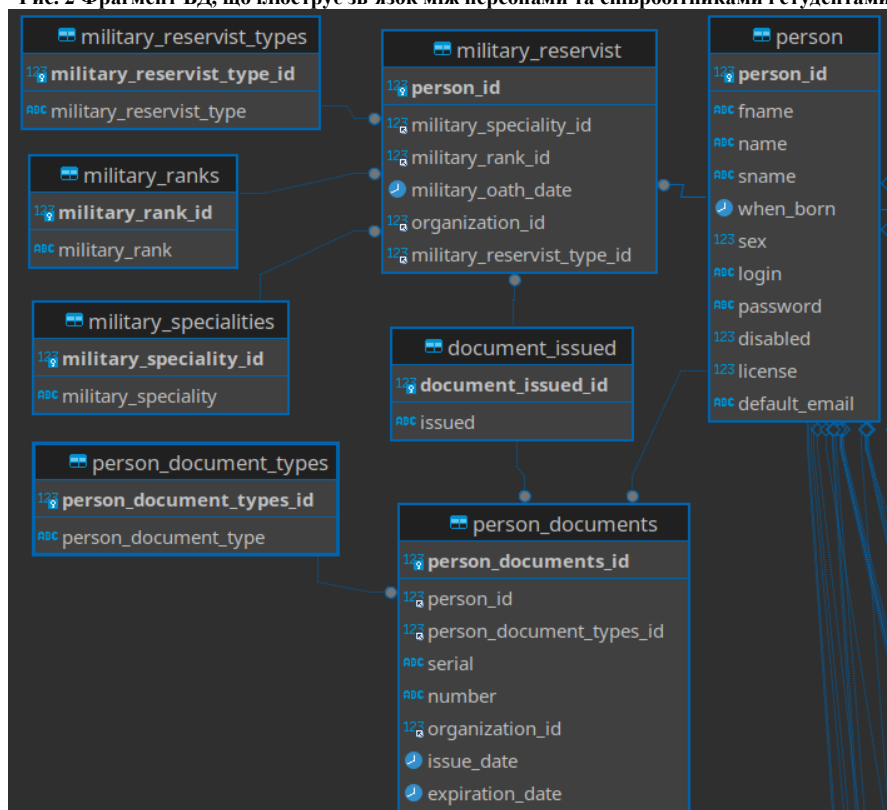


Рис. 3 Фрагмент БД, що ілюструє зв'язок між персонами та їх документами

Під час реалізації даної системи використовувались наступні ролі: гість, лаборант, гарант ОП, завідувач кафедри, методист деканату, диспетчер деканату, декан, фахівець навчального відділу, адміністратор.

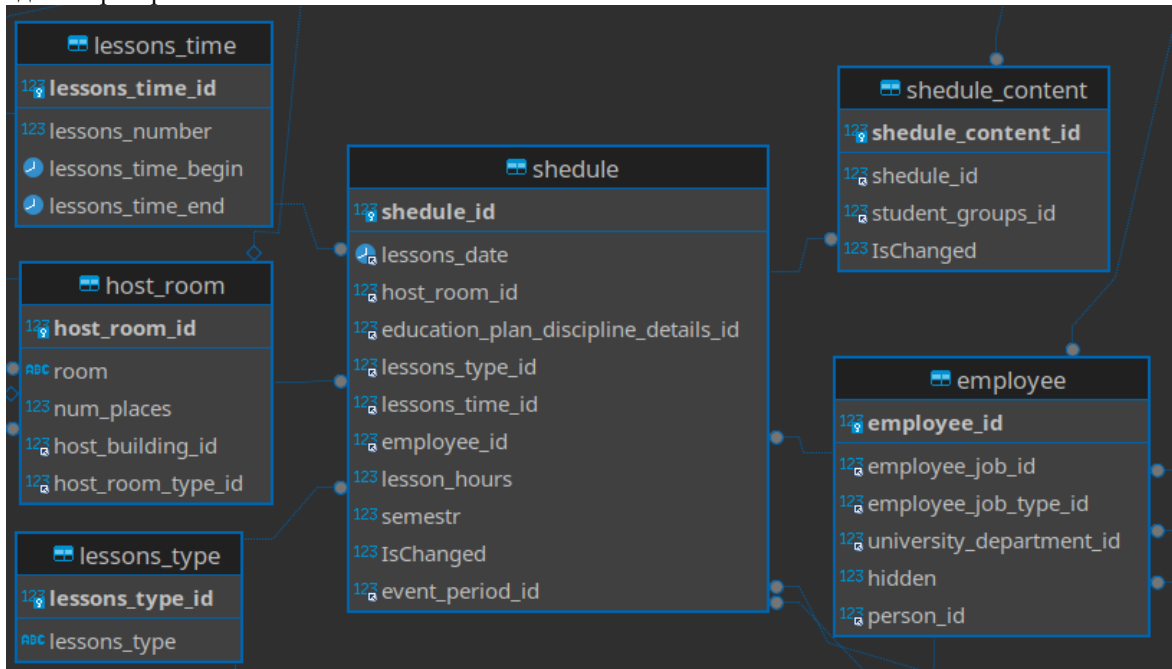


Рис. 4 Фрагмент БД, що забезпечує роботу з розкладом

Так, у лістингу 1 наведено SQL інструкції для створення ролі WebGuest, яка буде використовуватись для створення користувачів, від імені яких будуть з'єднуватись з базою веб додатки Schedule Site та Schedule Service [15, 16]. Створення ролі ScheduleDispatcher відбувається за допомогою bash скрипта, код якого наведено к лістингу 2.

Лістинг 1. Створення ролі

WebGuest

```
01. CREATE ROLE 'WebGuest';
02. GRANT SELECT ON Schedule Database.* TO WebGuest;
```

Як видно з лістингу 2 та рис. 2 - 4, роль ScheduleDispatcher не передбачає доступ до конфіденційної інформації (person\_extended, person\_contacts, person\_documents, military\_reservist та інші таблиці), водночас, ця роль дозволяє читання інформації з 38 таблиць та операції додавання, видалення та оновлення даних для таблиць host\_room\_lessons, lessons\_time, shedule, shedule\_content. Спроба користувача отримати доступ до заборонених ресурсів буде блокуватись на рівні СУБД (рис. 5).

Лістинг 2.

runScheduleDispatcherRole.sh

```
01. #!/bin/bash
02. dbName="ASTS_v3"
03. userRole='ScheduleDispatcher'
04. selectTables=("discipline" "discipline_employee" "education_form" "education_level"
"education_level_details" "education_payment" "education_plan" "education_plan_content"
"education_plan_discipline" "education_plan_discipline_content" "education_plan_discipline_details"
"education_plan_discipline_section" "education_qualification" "education_speciality"
"education_speciality_type" "education_type" "employee" "employee_job" "employee_job_type"
"event_period" "event_period_type" "group_type" "groups_flow" "host_building" "host_room"
"host_room_lessons" "host_room_type" "lessons_time" "lessons_type" "person" "science_degree"
"science_domain" "science_rank" "shedule" "shedule_content" "student_groups"
"university_department" "university_department_type")
05. IUDTables=("host_room_lessons" "lessons_time" "shedule" "shedule_content")
06. echo "#####"
07. echo '# #
08. echo '# Manage ScheduleDispatcher Role installer #
09. echo '# #
10. echo "#####"
11. mysql $1 --execute="DROP ROLE IF EXISTS '$userRole';"
12. mysql $1 --execute="CREATE ROLE IF NOT EXISTS '$userRole';"
```

```

13. echo '#                               SELECT Privileges                               #'
14. for table in "${selectTables[@]}";
15. do
16. echo "Process table: $table"
17. mysql $1 --execute="GRANT SELECT ON $dbName.$table TO $userRole;"
18. done;
19. echo '#                               INSERT,UPDATE,DELETE Privileges                               #'
20. for table in "${IUDTables[@]}";
21. do
22. echo "Process table: $table"
23. mysql $1 --execute="GRANT INSERT,UPDATE,DELETE ON $dbName.$table TO $userRole;"
24. done;
25. echo '#####'
26. echo '#                               #'
27. echo '#                               Manage role complete!                               #'
28. echo '#                               #'
29. echo '#####'

```

Використання подібних скриптів для керування ролями дозволяє полегшити процедуру адміністрування системи на різних хостах (серверах замовника). Як альтернативу даному скрипту можна використовувати чистий SQL скрипт, але при його написанні буде необхідно повторювати доволі значні частини коду.

```

MariaDB [ASTS_v4_dev]> SELECT * FROM person_documents;
ERROR 1142 (42000): SELECT command denied to user 'user1'@'localhost' fo
r table `ASTS_v4_dev`.`person_documents`
MariaDB [ASTS_v4_dev]> INSERT INTO event_period_type VALUE(1,"Семестр");
ERROR 1142 (42000): INSERT command denied to user 'user1'@'localhost' fo
r table `ASTS_v4_dev`.`event_period_type`

```

Рис. 5 Приклад блокування дій, що непередбачені роллю користувача

Після завершення проектування та налаштування бази даних, можна переходити до етапу проектування та розробки програмного забезпечення. Для реалізації заявленого функціоналу за допомогою Qt C++ 5.4 було створено A.S.T.S. Schedule Client, а за допомогою PHP 7 було створено такі компоненти: Schedule site and service та Schedule WebService.

с 2025-04-28 по 2025-05-04

Дата	I	II	III	IV	V	VI	VII	VIII
Понеділок 2025-04-28	комп'ютерні мережі Лекція: ауд. 205							
Вівторок 2025-04-29		комп'ютерні мережі Лабораторіє: ауд. 306	комп'ютерні мережі Лабораторіє: ауд. 306					
Середа 2025-04-30	Операційні системи та системне програмування Лекція: ауд. 205							
Четверг 2025-05-01	Операційні системи та системне програмування Лекція: ауд. 205							
П'ятниця 2025-05-02								
Субота 2025-05-03								
Неділя 2025-05-04								

© 2008-2016 Антонов Ю.С. All rights reserved! a.s.t.s.develop@gmail.com

Рис. 6 Приклад розкладу користувача

Schedule site and service – реалізує можливість отримання розкладу для викладача (рис. 6) або студента (рис. 7), зайнятість усіх викладачів кафедри, зайнятість аудиторії. За замовчуванням відображається розклад на поточний тиждень, але у користувачів є можливість за допомогою кнопок навігації переходити на будь-який потрібний тиждень назад або вперед. Також дана підсистема

дозволяє отримувати необхідну інформацію у вигляді JSON даних (факультети, кафедри, спеціальності, навчальні групи, аудиторії, розклад, тощо). Під час реалізації сервісу *ScheduleWebService* було використано PHPDoc, та побудовано відповідну об'єктну модель класів.

Понеділок 2025-04-28				
Пара	КН-24-1(А)	КН-24-1(Б)	КН-24-1(В)	КН-24-1(Г)
I			комп'ютерні мережі Лекція: ауд. 205 Антонов Ю.С.	
II			Вища математика для КН Лекція: ауд. 404 Фриз І.В.	
III	Вища математика для КН Практичне: ауд. 129 Фриз І.В.			
IV				
V				
VI				
VII				
VIII				
Вівторок 2025-04-29				
Пара	КН-24-1(А)	КН-24-1(Б)	КН-24-1(В)	КН-24-1(Г)
I			Основи програмування Лекція: ауд. 307 Бабаков Р.М.	
II	комп'ютерні мережі Лабораторне: ауд. 306 Антонов Ю.С.	комп'ютерні мережі Лабораторне: ауд. 304 Хмельський Ю.С.	Вища математика для КН Практичне: ауд. 129 Фриз І.В.	
III	комп'ютерні мережі Лабораторне: ауд. 306 Антонов Ю.С.		комп'ютерні мережі Лабораторне: ауд. 304 Хмельський Ю.С.	
IV				комп'ютерні мережі Лабораторне: ауд. 304 Хмельський Ю.С.
V				
VI				
VII				
VIII				

Рис. 7 Приклад розкладу потоку 1-го курсу з розбиттям на підгрупи

Під час реалізації A.S.T.S. Schedule Client використовувався шаблон проектування MVC, разом з використанням таких базових класів: *QSqlTableModel*, *QSqlRelationalTableModel*, *QSqlQueryModel*. Для кожної таблиці бази даних, що використовувалась для роботи з розкладом, в залежності від потреб створювалась одна з моделей: *TableModel*, *RelationalTableModel* або *SqlQueryModel*.

Взаємодія співробітників із системою відбувається наступним чином:

на рівні кафедри вноситься інформація про навчальну програму, освітні компоненти, викладачів, що будуть викладати дисципліну у вказаному навчальному періоді;

на рівні факультету або інших структурних підрозділів подібного рівня, заповнюється інформація про аудиторний фонд, а саме: назва аудиторії, її тип, кількість робочих місць тощо.

на рівні диспетчера деканату відбувається складання розкладу, враховуючи всю надану іншими підрозділами інформацію та внесення його до системи (рис. 8 та рис. 9).

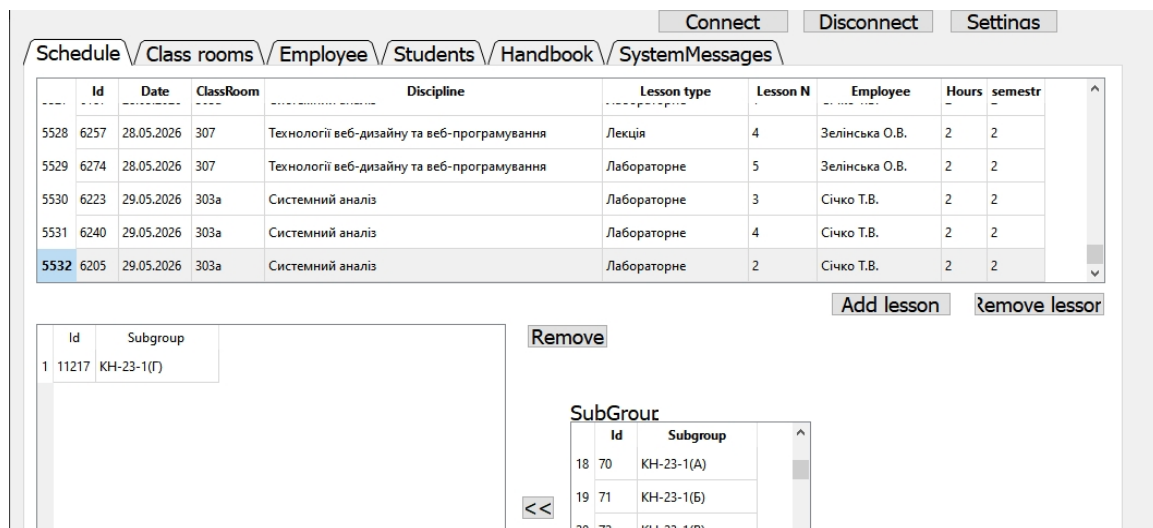


Рис. 8 Головне вікно Schedule Client після авторизації



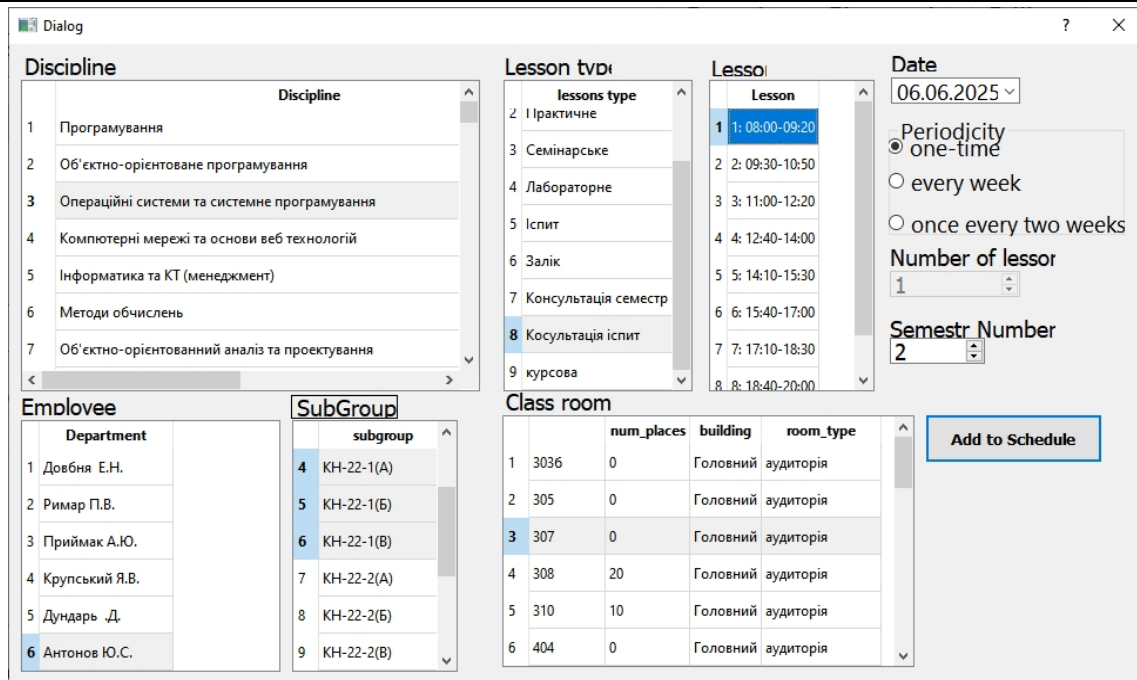


Рис. 9 Додавання занять у Schedule Client

Всі зміни, що вносяться у систему одразу автоматично стають доступними для всіх учасників навчального процесу.

### Висновки

Розглянуті у роботі підходи, щодо проектування системи керування розкладом закладу вищої освіти дозволили на протязі місяця отримати працездатну систему, яка експлуатувалась на факультеті математики та інформаційних технологій з 2015 по 2019 роки та у 2022 – 2025 в обмеженому режимі. Студенти та викладачі мали можливість отримувати доступ до актуального розкладу навчальних занять у режимі реального часу.

Зберігання розкладу у вигляді календаря, дозволило:

- уніфікувати бізнес процеси та процес розробки програмного забезпечення;
- працювати з розкладом навчальних занять;
- працювати з розкладом сесії;
- вносити у розклад семестрові консультації та інші нерегулярні події.

У подальшому планується вдосконалення існуючої підсистеми та її інтеграція з підсистемою успішності студентів.

### Література

1. schedule-system.md [Електронний ресурс] – Режим доступу: [https://gitlab.com/guyver-free-projects/scientific-articles-supplements/-/blob/master/schedule-system.md?ref\\_type=heads](https://gitlab.com/guyver-free-projects/scientific-articles-supplements/-/blob/master/schedule-system.md?ref_type=heads) (дата звернення: 13.06.2025). – Назва з екрана.
2. Рymar П. В. Розробка автоматизованої системи формування розкладу навчальних занять з використанням ІС «Підприємство». / П. В. Рymar, Б. С. Войтко // *Вісник Хмельницького національного університету. Серія «Технічні науки»*. 2021. № 5. С. 36-40. DOI: 10.31891/2307-5732-2021-301-5-36-40
3. Шинкаренко В. І. Составление расписания занятий университета на основе конструктивного моделирования. / В. И. Шинкаренко, А. А. Жеваго // *Радиоелектроника, інформатика, управління*. 2019. № 3. С. 152–162. DOI: 10.15588/1607-3274-2019-3-17.
4. Береговых Ю.В. Алгоритм составления расписания занятий / Ю.В. Береговых, Б.А. Васильев, Н.А. Володин // *«Искусственный интеллект» 2'2009* С 50–56
5. Голуб Б.Л. Програмна система формування розкладу занять у закладі вищої освіти / Б.Л. Голуб, Д.В. Ветрова, К.О. Пронішина // *Математичні машини і системи*. – 2019. – № 4 С 100–109.
6. Davison M. Modelling and solving the university course timetabling problem with hybrid teaching considerations / M. Davison, A. Kheiri, K.G. Zografos // *Journal of Scheduling*. 2024. Vol. 28. P 195–215 <https://doi.org/10.1007/s10951-024-00817-w>
7. University Timetabling [Електронний ресурс] – Режим доступу: <https://www.unitime.org> (дата звернення: 13.06.2025). — Назва з екрана.
8. Murray K. Modeling and Solution of a Complex University Course Timetabling Problem. / K. Murray, T. Müller, H. Rudová // *Practice and Theory of Automated Timetabling VI. PATAT 2006*. [Brno, Czech Republic], August 30 – September 1 of 2006. P. 189–209. [https://doi.org/10.1007/978-3-540-77345-0\\_13](https://doi.org/10.1007/978-3-540-77345-0_13)

9. Müller, T. Real-life curriculum-based timetabling with elective courses and course sections / T. Müller, H. Rudová // *Annals of Operations Research* 2016. Volume 239, P. 153–170. <https://doi.org/10.1007/s10479-014-1643-1>
10. Rudová, H. Complex university course timetabling / H. Rudová, T. Müller, K. Murray // *Journal of Scheduling* 2011 Volume 14, P. 187–207. <https://doi.org/10.1007/s10951-010-0171-3>
11. SQL vs. NoSQL data [Електронний ресурс] – Режим доступу: <https://learn.microsoft.com/en-us/dotnet/architecture/cloud-native/relational-vs-nosql-data> (дата звернення: 13.06.2025). – Назва з екрана.
12. Швець М.Ю. Порівняння SQL та NoSQL баз даних / М.Ю. Швець, Д.С. Заруба, Ю.С. Хохлов // *Вчені записки ТНУ імені В.І. Вернадського. Серія: технічні науки*. 2018. Т29 (68) Ч.2 №6 С 21-25
13. Schedule module for Antonov Students Test System: а.с. №103272 від 18.03.2021. Антонов Ю. С. Заявл. №с202100796 від 15.02.2021, Авторське право і суміжні права. Бюл. № 64, 2021 С 170-171.
14. Антонов Ю. «Залежність швидкодії програм від інструментальних засобів розробки та синтаксичних конструкцій» / Ю. Антонов // *Ukrainian Journal of Information Systems and Data Science*. 2024 №1 С. 27–39. doi: 10.31558/2786-9482.2024.1.3.
15. MariaDB. User Account Management. [Електронний ресурс] – Режим доступу: <https://mariadb.com/kb/en/account-management-sql-commands/> (дата звернення: 13.06.2025). – Назва з екрана.
16. MySQL 8.4 Reference Manual. Account Management Statements [Електронний ресурс] – Режим доступу: <https://dev.mysql.com/doc/refman/8.4/en/account-management-statements.html> (дата звернення: 13.06.2025). – Назва з екрана.

### References

1. schedule-system.md [Електронний ресурс] – Режим доступу: [https://gitlab.com/guyver-free-projects/scientific-articles-supplements/-/blob/master/schedule-system.md?ref\\_type=heads](https://gitlab.com/guyver-free-projects/scientific-articles-supplements/-/blob/master/schedule-system.md?ref_type=heads) (дата звернення: 13.06.2025). – Назва з екрана.
2. Rymar P. V. Rozrobka avtomatyzovanoi systemy formuvannia rozkladu navchalnykh zaniat z vykorystanniam IS «Pidpryemstvo». / P. V. Rymar, B. S. Voitko // *Visnyk Khmelnytskoho natsionalnoho universytetu. Seriiia «Tekhnichni nauky»*. 2021. № 5. S. 36-40. DOI: 10.31891/2307-5732-2021-301-5-36-40
3. Shinkarenko V. I. Sostavlenie raspisaniya zanyatij universiteta na osnove konstruktivnogo modelirovaniya. / V. I. Shinkarenko, A. A. Zhevago // *Radioelektronika, informatyka, upravlinnia*. 2019. № 3. S. 152–162. DOI: 10.15588/1607-3274-2019-3-17.
4. Beregovyh Yu.V. Algoritm sostavleniya raspisaniya zanyatij / Yu.V. Beregovyh, B.A. Vasilev, N.A. Volodin // «*Iskusstvennyj intellekt*» 2'2009 С 50–56
5. Holub B.L. Programna systema formuvannia rozkladu zaniat u zakladi vyshchoi osvity / B.L. Holub, D.V. Vetrova, K.O. Pronishyna // *Matematychni mashyny i systemy*. – 2019. – № 4 С 100–109.
6. Davison M. Modelling and solving the university course timetabling problem with hybrid teaching considerations / M. Davison, A. Kheiri, K.G. Zografos // *Journal of Scheduling*. 2024. Vol. 28. P 195–215 <https://doi.org/10.1007/s10951-024-00817-w>
7. University Timetabling [Електронний ресурс] – Режим доступу: <https://www.unitime.org> (дата звернення: 13.06.2025). — Назва з екрана.
8. Murray K. Modeling and Solution of a Complex University Course Timetabling Problem. / K. Murray, T. Müller, H. Rudová // *Practice and Theory of Automated Timetabling VI. PATAT 2006*. [Brno, Czech Republic], August 30 – September 1 of 2006. P. 189–209. [https://doi.org/10.1007/978-3-540-77345-0\\_13](https://doi.org/10.1007/978-3-540-77345-0_13)
9. Müller, T. Real-life curriculum-based timetabling with elective courses and course sections / T. Müller, H. Rudová // *Annals of Operations Research* 2016. Volume 239, P. 153–170. <https://doi.org/10.1007/s10479-014-1643-1>
10. Rudová, H. Complex university course timetabling / H. Rudová, T. Müller, K. Murray // *Journal of Scheduling* 2011 Volume 14, P. 187–207. <https://doi.org/10.1007/s10951-010-0171-3>
11. SQL vs. NoSQL data [Електронний ресурс] – Режим доступу: <https://learn.microsoft.com/en-us/dotnet/architecture/cloud-native/relational-vs-nosql-data> (дата звернення: 13.06.2025). – Назва з екрана.
12. Shvets M.Iu. Porivniannia SQL ta NoSQL baz danykh / M.Iu. Shvets, D.S. Zaruba, Yu.S. Khokhlov // *Vcheni zapysky TNU imeni V.I. Vernadskoho. Seriiia: tekhnichni nauky*. 2018. Т29 (68) Ч.2 №6 С 21-25
13. Schedule module for Antonov Students Test System: а.с. №103272 від 18.03.2021. Antonov Yu. S. Zaiavl. №с202100796 від 15.02.2021, Avtorske pravo i sumizhni prava. Biul. № 64, 2021 С 170-171.
14. Antonov Yu. «Zalezhnist shvydkodii prohran vid instrumentalnykh zasobiv rozrobky ta syntaksychnykh konstrukttsii» / Yu. Antonov // *Ukrainian Journal of Information Systems and Data Science*. 2024 №1 С. 27–39. doi: 10.31558/2786-9482.2024.1.3.
15. MariaDB. User Account Management. [Електронний ресурс] – Режим доступу: <https://mariadb.com/kb/en/account-management-sql-commands/> (дата звернення: 13.06.2025). – Назва з екрана.
16. MySQL 8.4 Reference Manual. Account Management Statements [Електронний ресурс] – Режим доступу: <https://dev.mysql.com/doc/refman/8.4/en/account-management-statements.html> (дата звернення: 13.06.2025). – Назва з екрана.