

**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ**

Харківський національний університет імені В. Н. Каразіна

Факультет математики і інформатики

Кафедра теоретичної та прикладної інформатики

**Кваліфікаційна робота**

**магістр**

на тему

**“Аналіз та розробка методів синхронізації слабо структурованої інформації між відділами організації”**

Виконав: студент 2 курсу, другого  
(магістерського) рівня вищої освіти, групи  
МФ-61  
спеціальність 122 «Комп’ютерні науки»  
освітньо-професійна програма  
«Інформатика»

Соврасов Ю.Е.

Керівник: Морозова А. Г.

Рецензент \_\_\_\_\_

Харків – 2025 року

## ЗМІСТ

### 1. ВСТУП

- 1.1. Формулювання мети роботи, задач та обґрунтування актуальності теми
- 1.2. Стислий огляд відомих результатів
- 1.3. Відомості про одержані результати та їх новизна

### 2. ОСНОВНА ЧАСТИНА

- 2.1. Постановка задачі
- 2.2. Розвинутий огляд сучасного стану справ в області
- 2.3. Матеріали та методи дослідження
- 2.4. Описання, обґрунтування та аналіз алгоритмів і результатів дослідження
  - 2.4.1. Інформаційні системи
  - 2.4.2. Схема взаємодії
  - 2.4.3. Інформацію, яку зберігає кожна система про студентів
  - 2.4.4. Опис кожного виду синхронізації
    - 2.4.4.1. Синхронізація ЄДЕБО → ОКО
    - 2.4.4.2. Синхронізація ОКО → Електронний деканат
    - 2.4.4.3. Синхронізація Бухгалтерія → ОКО (Обробка DBF-файлів)
    - 2.4.4.4. Синхронізація ОКО → Бухгалтерія
  - 2.4.5. Візуалізація результатів синхронізації

### 3. ВИСНОВКИ

### 4. СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

## 1. ВСТУП

### 1.1. Формулювання мети роботи, задач та обґрунтування актуальності теми

У сучасних організаціях зростає потреба у синхронізації інформації між різними відділами, особливо якщо вони працюють з різними інформаційними системами та базами даних. У таких умовах часто виникають труднощі, пов'язані з узгодженням даних, їх актуальністю та достовірністю. Це особливо актуально для освітніх установ, де важливо своєчасно оновлювати інформацію про студентів, їхній статус, фінансові зобов'язання тощо.

Метою даної роботи є аналіз існуючих підходів до синхронізації слабо структурованої інформації між відділами організації та розробка ефективних методів для їх застосування. Основна увага приділяється синхронізації даних між системою ЄДЕБО (Єдина державна електронна база з питань освіти), системи Е-деканат та базою контрактного навчання. Також важливою задачею є синхронізація між бухгалтерією та відділом контрактного навчання.

#### Основні задачі роботи:

- Дослідити існуючі методи синхронізації даних та їх ефективність у контексті розподілених інформаційних систем;
- Визначити основні проблеми, що виникають при обміні даними між відділами, які використовують різні джерела інформації;
- Розробити механізм автоматизованої синхронізації даних між ЄДЕБО та базою контрактного навчання;
- Розробити механізм автоматизованої синхронізації даних між базою контрактного навчання та системою Е-деканат;

- Розробити механізм автоматизованої синхронізації даних між базою контрактного навчання та бухгалтерією;
- Запропонувати методи підвищення точності та актуальності синхронізованої інформації;
- Оцінити ефективність запропонованих методів на практичному прикладі.

Актуальність теми обумовлена необхідністю усунення дублювання інформації, забезпечення її коректності та зменшення трудовитрат, пов'язаних з ручною перевіркою та оновленням даних. Відсутність єдиного підходу до обміну інформацією між відділами може призвести до помилок, що негативно впливають на управлінські рішення.

## **1.2. Стислий огляд відомих результатів**

Сучасні освітні та адміністративні установи дедалі більше стикаються з необхідністю інтеграції розрізнених інформаційних систем і баз даних, що веде до виникнення комплексної проблеми синхронізації даних. Умикнення ЄДЕБО, локальних систем академічного обліку, електронного деканату та бухгалтерії створює низку викликів, що активно досліджувалися в науковій та практичній літературі останніх років. Нижче наведено основні підходи до синхронізації, які застосовувалися в реальному середовищі:

- **Централізовані сховища (Data Warehouse).**

Даний підхід передбачає створення єдиного репозиторію, куди збираються дані з усіх відділів. Перевага — наявність «точки правди» для аналітики й звітності, але водночас централізація вимагає значних ІТ-інвестицій і може призвести до суттєвих затримок оновлення, адже зміни в локальних системах відображаються в сховищі за розкладом.

- **API-інтеграція та сервіси обміну.**

Використання RESTful або SOAP API забезпечує можливість «жвавого» (near-real-time) обміну невеликими обсягами даних між системами. Такий підхід дозволяє миттєво передавати оновлення від однієї системи до іншої, але потребує наявності стабільного, добре задокументованого інтерфейсу з обох боків — недолік для ЄДЕБО, яке не має відкритого публічного API.

- **ETL-процеси (Extract, Transform, Load).**

Декілька разів на добу чи добу в цілому запускаються спеціальні модулі, що вибирають дані з вихідних систем, перетворюють їх відповідно до потреб цільових баз і завантажують у ці бази. ETL підходить для структурованих, добре описаних наборів даних, але його налаштування вимагає окремих інструментів (Talend, Pentaho, SSIS) і значних зусиль на підтримку трансформацій.

- **Автоматизовані файли експорту/імпорту (cronjobs, Quartz).**

Це класичний сценарій, коли дані «вигризаються» у вигляді CSV чи DBF-файлів за фіксованим розкладом (наприклад, щодня, щомісяця) і передаються до інших систем. У контексті освітніх установ, і зокрема ЄДЕБО, такий підхід є найпоширенішим через відсутність відкритого API. Проте необхідність ручного запуску скриптів або перевірки оператором ускладнює процес і може призвести до ручних помилок.

У реальних впровадженнях часто застосовують комбінований підхід: регулярні ETL-завдання доповнюються легкими скриптами на Java або Python для роботи з файловими експортами, а для максимальної автоматизації

використовують планувальники (Quartz Scheduler). Однак кожен із цих методів має власні обмеження: централізовані сховища — затримки, API-інтеграції — відсутність стабільних інтерфейсів, ETL — складність адаптації під змінні формати, файловий обмін — ризик неконсистентності через пізні або неповні експортні файли.

У рамках цієї роботи ми спираємося саме на механізм автоматизованого файлового обміну (CSV/DBF), доповненого власними модулями нормалізації та перевірки даних, інтегрованими в існуючу Java-додатку та розкладові Quartz-jobs. Такий підхід забезпечує не лише мінімальні втрати даних, але й гнучкість у налаштуванні правил звірки та можливість оперативного втручання при виявленні невідповідностей.

### **1.3. Відомості про одержані результати та їх новизна**

В результаті виконання даної кваліфікаційної роботи було досягнуто таких ключових результатів:

- **Комплексний аналіз існуючих методів синхронізації**

Детально вивчено та класифіковано основні підходи — централізовані сховища, API-інтеграцію, ETL-процеси та автоматизований файловий обмін. Виявлено їхні сильні сторони (швидкість оновлення, гнучкість, наявність «точки правди») та обмеження (затримки, складність налаштування, відсутність відкритих інтерфейсів у ЄДЕБО).

- **Розробка модуля синхронізації ЄДЕБО → ОКО**

Створено Java-модуль, який за допомогою Quartz-job двічі на рік завантажує CSV-експорт з ЄДЕБО, виконує багатоетапну нормалізацію (апострофи, різні

варіанти Ф.І.Б.), нечітке зіставлення по декількох атрибутах і оновлює локальну базу контрактного відділу.

- **Реалізація процесу синхронізації ОКО → Е-деканат**

Впроваджено щоденний експорт списку боржників (скрипт ExportDebtStudent) та автоматичний імпорт у систему електронного деканату з блокуванням «недопуску» за заборгованість. Забезпечено коректне оброблення граничних випадків (відсутність edbo\_number, мережеві таймаути) через механізм повторення неуспішних транзакцій.

- **Розробка каналу даних ОКО → Бухгалтерія**

Організовано щомісячний експорт DBF-файлу з деталями нарахувань, оплат, повернень і передплат, а також разом з Excel-звітністю («Оборотна відомість за ...») автоматизовано формування необхідних звітів факультетів.

- **Автоматизований цикл обробки файлів бухгалтерії → ОКО**

Запроваджено механізм парсингу DBF через бібліотеку Jamel DBF Reader, алгоритм видобутку номеру контракту з призначення платежу (дог.№ ...), пошук студента за контрактом або Ф.І.Б. та інтерактивне підтвердження проведення операцій перед занесенням запису в таблицю Payment\_Current.

- **Побудова звітності та верифікація**

Розроблено комплект SQL-запитів для агрегації даних по факультетах (нарахування, повернення, оплати, передплата, борг), що лягають в основу

щомісячних Excel-звітів. Додатково впроваджено механізм перевірки даних із логуванням причин невідповідностей (зміна прізвища, несумісні спецсимволи, розбіжності в курсі навчання).

Новизна роботи полягає в інтегрованому застосуванні відразу чотирьох незалежних каналів синхронізації (ЄДЕБО → ОКО, ОКО → Е-деканат, бухгалтерія → ОКО і ОКО → бухгалтерія) з урахуванням специфічних форматів освітніх даних, обмежень зовнішніх систем та особливостей кадрових процедур.

Запропонований підхід дозволяє:

- **Скоротити** ручні операції з оновлення контингенту й фінансових звітів з кількох годин до кількох хвилин автоматичної обробки.
- **Підвищити** достовірність інформації завдяки багаторівневій верифікації та нормалізації вхідних даних.
- **Уніфікувати** різноманітні джерела даних у єдину логіку обробки, що забезпечує масштабованість і швидке впровадження змін у бізнес-логіці.

Отже, розроблені методи і програмні модулі значно покращують якість та своєчасність оновлення студентських даних, зменшують ризик помилок і підвищують оперативність прийняття управлінських рішень в університеті.

## 2. ОСНОВНА ЧАСТИНА

### 2.1. Постановка задачі

У сучасних освітніх установах забезпечення узгодженості даних про студентів є критично важливим для своєчасного прийняття управлінських рішень і ефективного надання освітніх послуг. Однак інформація про студентів, їхній академічний статус та фінансові взаємовідносини водночас зберігається в кількох незалежних інформаційних системах: Єдиній державній електронній базі з питань освіти (ЄДЕБО), локальній базі контрактного навчання (ОКО), системі «Електронний деканат» та бухгалтерській підсистемі. Різна структура даних, відмінні формати файлів для обміну та різні частоти оновлення призводять до розривів у консолідованому поданні інформації, дублювання записів і затримок у відображенні змін.

### Мета дослідження

Розробити та впровадити комплекс механізмів автоматизованої синхронізації слабо структурованих даних між ключовими підсистемами університету (ЄДЕБО ↔ ОКО ↔ «Електронний деканат» ↔ бухгалтерія), що забезпечить:

- узгодженість контингенту студентів у всіх системах,
- своєчасне оновлення фінансового стану (борги, оплати, передплати),
- мінімізацію ручних операцій при імпорті/експорті даних.

### Основні завдання дослідження

1. Проаналізувати існуючі підходи до обміну та зберігання даних у підсистемах ЄДЕБО, ОКО, «Електронний деканат» і бухгалтерії, виявити їхні слабкі місця та вимоги до інтеграції.
2. Розробити єдину логіку обміну файлами, включаючи формати CSV, DBF і Excel, а також відпрацювати стандарти ідентифікації та валідації записів.
3. Створити алгоритми імпортних процесів:
  - завантаження та парсинг CSV-експорту з ЄДЕБО,
  - обробка DBF-файлів бухгалтерії для реєстрації оплат,
  - імпорт щоденних звітів про боржників у «Електронний деканат».
4. Реалізувати експортні механізми:
  - щоденний експорт списку боржників з ОКО у CSV для деканату,
  - щомісячне формування Excel-звітів «Оборотна відомість» і DBF-файлів для бухгалтерії.
5. Обґрунтувати критерії зіставлення записів між системами (унікальні ідентифікатори, нечітке порівняння ПІБ, дати народження тощо) та реалізувати модулі нормалізації даних (апострофи, регістр, пробіли).
6. Організувати механізм контролю та логування: фіксація кількості успішних оновлень, відхилених записів і технічних помилок для подальшого аналізу й оперативного втручання.

Результатом виконання цих завдань стане надійна й масштабована інфраструктура обміну та синхронізації, яка забезпечить коректність та своєчасність оновлення даних у всіх задіяних інформаційних системах університету.

## 2.2. Розвинутий огляд сучасного стану справ в області

У сучасних організаціях все частіше виникає необхідність інтеграції розрізнених інформаційних систем, що використовуються різними відділами для обліку та управління даними. Зокрема, у вищих навчальних закладах одночасно експлуатуються системи для ведення державного реєстру студентів (ЄДЕБО), контрактної бази, бухгалтерських обліків та електронного деканату. Відсутність уніфікованого підходу до обміну даними призводить до дублювання інформації, ручного введення, затримок і помилок.

Найбільш поширені підходи до вирішення цієї проблеми включають:

- **Використання API**
  - Розробка програмних інтерфейсів (REST, SOAP) для двостороннього обміну повідомленнями між різними системами.
  - Переваги: реальний час оновлення, гнучкість, можливість фільтрації та агрегації даних на рівні запиту.
  - Недоліки: потреба в однотипній аутентифікації, змінах в API зовнішнього середовища (наприклад, в ЄДЕБО немає відкритого REST-інтерфейсу), велика складність початкового налаштування.
- **Реплікація баз даних**
  - Автоматичне дублювання окремих таблиць або повних схем між серверами за допомогою вбудованих механізмів СУБД (наприклад, SQL Server Replication).
  - Переваги: висока швидкість копіювання, підтримка консистентності, мінімальні затримки.

- Недоліки: надмірні об'єми переданих даних, складність конфігурації та обмеження мережевих ресурсів, ризик блокувань у продуктивних таблицях.
- **ETL-процеси (Extract, Transform, Load)**
  - Використання інструментів (SQL Server Integration Services, Talend, Pentaho) для регулярного завантаження даних з одного сховища в інше з одночасною трансформацією.
  - Переваги: можливість складних перетворень, очищення та валідації даних; підтримка різних джерел (файли, бази, веб-сервіси).
  - Недоліки: потреба в окремих середовищах запуску, відставання на період «вікна» ETL, висока вартість впровадження та підтримки.
- **Форматовані файли для обміну даними**
  - Експорт/імпорт CSV, JSON або XML-файлів із визначеною структурою.
  - Переваги: простота реалізації, відсутність потреби в спеціальних ліцензіях, можливість обробки великих обсягів даних пакетно.
  - Недоліки: обов'язковий розподіл прав доступу до файлових серверів або хмарних дисків, відсутність контролю транзакційності, складність у роботі з «грубими» структурами (напр., DBF, Excel).

У рамках даної роботи обрана модель автоматизованого файлового обміну із застосуванням стандартизованих CSV і DBF-звітів, що зберігаються та передаються через Google Drive та інші захищені канали. Це рішення поєднує простоту налаштування, низькі витрати на підтримку та достатню гнучкість для реалізації:

### **1. Синхронізації ЄДЕБО → контрактна база (ОКО)**

- Завантаження двічі на рік CSV-файлів з офіційним контингентом студентів.
- Обробка локальним Java-модулем, нормалізація даних та оновлення edbo\_number.

## **2. Синхронізації контрактна база → електронний деканат**

- Щоденний експорт CSV списку боржників та імпорт у систему ЕД із блокуванням академічних прав.

## **3. Обміну з бухгалтерією**

- Щоденний парсинг DBF-файлів оплат для актуалізації таблиці Payment\_Current.
- Щомісячний експорт DBF і побудова Excel-звітів по факультетах (нарахування, оплати, борги, дебет/кредит).

Такий комбінований підхід забезпечує своєчасність і коректність обміну даними, мінімізує ручні операції та дозволяє швидко реагувати на зміни у структурах зовнішніх систем.

### **2.3. Матеріали та методи дослідження**

Для досягнення поставлених завдань у роботі виконано комплексний аналіз технічної архітектури та організації процесів обміну даними між інформаційними системами університету. Зокрема, було досліджено способи інтеграції десктоп-додатка контрактного відділу з іншими підсистемами (ЄДЕБО, електронний деканат, бухгалтерія), розглянуто механізми керування потоками даних, захисту інформації та забезпечення безперервності обслуговування. У межах цієї роботи:

1. Проаналізовано архітектуру Java-дodatка: виявлені ключові модулі відповідальні за імпорт зовнішніх файлів (CSV, DBF, Excel), звірку записів та оновлення локальної бази даних.
2. Вивчено структуру сховища SQL Server: аудит таблиць, індексів і зв'язків між сутностями, визначення критичних полів для відображення залежностей між студентами та їхніми фінансовими операціями.
3. Оцінено засоби планування і автоматизації завдань: використання запланованих job-скриптів для регулярного імпорту та експорту даних у встановлені терміни.
4. Реалізовано алгоритми нормалізації й зіставлення записів: розроблено процедури обробки текстових полів (нормалізація апострофів, видалення зайвих пробілів, нечітке порівняння ПІБ), а також побудовано логіку верифікації даних за унікальними ключами.
5. Побудовано механізм логування та контролю якості: ведення детального журналу успішних оновлень, відхилених записів і технічних помилок для подальшого аналізу та корекції.

## Матеріали дослідження

- **CSV-записи експорту ЄДЕБО** — файли з даними контингенту студентів контрактної та ваучерної форм навчання.
- **Локальна база даних SQL Server** — таблиці Student, CommonStudentReport, CommonPaymentReport та інші довідники, що забезпечують зберігання персональних та фінансових даних.
- **DBF-файли бухгалтерії** — експортні таблиці з щоденними даними про оплати студентів, які імпортуються для оновлення платежів.

- **Excel-звіт “Оборотна відомість”** — щомісячний документ, до 10-го числа поточного місяця за попередній місяць, що містить зведені показники по факультетах: нарахування, повернення, оплати, передплати, борги, дебет та кредит.
- **Планувальник завдань ОС/СУБД** або сторонній планувальник — для періодичного запуску імпорتنих та експортних скриптів у встановлений час.
- **Java-бібліотеки** для парсингу: OpenCSV (CSV), Jamel DBF Reader (DBF) та Apache POI (Excel).

## Методи дослідження

### 1. Аналіз архітектури застосунку

- Досліджено, як Java-додаток взаємодіє з SQL Server через RMI.
- Визначено місце, де реалізовано логіку обробки CSV-файлів.
- Виділено внутрішні модулі: імпорт даних, звірка, оновлення, логування.

### 2. Дослідження структури сховища

- Проведено аудит таблиць SQL Server: імена полів, типи даних, наявність індексів.
- Визначено ключові параметри для зіставлення (ПІБ, дата народження, програма).
- Оцінено швидкодію запитів з урахуванням обмежень RMI та VPN.

### 3. Обробка даних у Java-додатку

- **Парсинг CSV:** реалізовано з використанням стандартних Java-бібліотек (BufferedReader, String.split()) або OpenCSV).
- **Нормалізація текстових полів:** при обробці враховано особливості апострофів, пробілів, регістру.
- **Алгоритм зіставлення:** побудовано логіку пошуку співпадінь із заданим порогом точності (наприклад, 100% по РНОКПП + 90% по ПІБ).
- **Оновлення БД:** через DAO-шар із підтримкою транзакційності, з фільтрацією записів, які не пройшли перевірку.

#### 4. Логування та контроль результатів

- Формується лог-файл, що містить інформацію про:
  - Успішно оновлені записи
  - Відкинуті через невідповідність
  - Технічні помилки при RMI-запитах або мережеві тайм-аути

#### Обґрунтування вибору технологій

- Java

Існуючий додаток відділу контрактного навчання написаний на мові Java. Мова обрана за її багатий набір бібліотек, крос-платформеність.

- SQL Server

Забезпечує надійне централізоване сховище з високою продуктивністю, розвинутими можливостями індексації та транзакційністю.

- Azure

Хмарна платформа для хостингу сервера баз даних і проміжних сервісів з гарантією доступності, масштабованості та резервного копіювання.

- Google Drive API

Використовується для автоматизованого обміну CSV-файлами боржників між ОКО та «Електронним деканатом», забезпечує надійний HTTPS-канал та версіонування файлів.

- Quartz Scheduler

Фреймворк для планування job-скриптів:

- ExportDebtStudent (щоденний експорт боржників: Пн–Пт, 2:00),
- job-перевірка оновлених файлів у «Е-Деканаті».

- OpenCSV

Легковагова Java-бібліотека для парсингу/генерації CSV, підтримує обробку заголовків, експорт у стрічкові потоки.

- `Jamel DBF Reader` (`org.jamel.dbf.DbfReader`,  
`org.jamel.dbf.structure.DbfHeader`)

Забезпечує коректне читання DBF-файлів від бухгалтерії, автоматично інтерпретує кодові сторінки та структуру записів.

## **2.4. Описання, обґрунтування та аналіз алгоритмів і результатів дослідження**

### **2.4.1. Інформаційні системи**

Університет використовує кілька розподілених інформаційних систем (ІС), кожна з яких відповідає за окремі аспекти обліку студентів. Відсутність централізованої системи спричиняє необхідність синхронізації між ними. Нижче наведено огляд ключових ІС, що беруть участь у процесі обміну та узгодження даних.

### **Електронний деканат (ЕД)**

- **Призначення:** Система для ведення академічної історії студентів, обліку відвідуваності, оцінювання та управління доступом до сесії.
- **Функції:** Відображає академічний статус студента, дозволяє/блокує доступ до сесії у разі наявності заборгованості.
- **Роль у синхронізації:**
  - Використовує ЄДЕБО-номер як ключ до зв'язку з іншими системами.
  - Автоматичний job перевіряє наявність нового файлу export.csv на Google Drive та імпортує дані про боржників.
  - Після імпорту система блокує студентів з боргами від можливості виставлення оцінок та доступу до сесії.
- **Формат обміну:** CSV-файл, що містить:
  - A1 – дата експорту
  - A2 і далі – ЄДЕБО-номер
  - B2 і далі – значення (0 – немає боргу, 1 – є борг)

### **Єдина державна електронна база з питань освіти (ЄДЕБО)**

- **Призначення:** Центральна державна система, яка виконує роль еталонного джерела контингенту студентів.
- **Функції:** Зберігає офіційні дані щодо статусу студентів, освітніх програм, джерела фінансування, форм навчання тощо.

- **Формат даних:** Системою експортуються CSV-файли, які використовуються для звірки з локальними базами (ОКО).
- **Роль у синхронізації:** Джерело правдивих даних, на основі яких виконується звірка контингенту та оновлення інформації в локальних системах.

### Інформаційна система бухгалтерії

- **Призначення:** Облік платежів студентів та фінансових операцій по факультетах.
- **Функції:** Зберігає історію нарахувань, платежів, переплат тощо.
- **Формат даних:** Системою експортуються DBF-файли, які використовуються для завантаження інформації про оплати.
- **Роль у синхронізації:**
  - Не має прямої інтеграції з ЕД чи ЄДЕБО.
  - Дані використовуються контрактним відділом (ОКО) для формування фінансового стану студентів.
  - Джерело для формування експорту про борги, які згодом передаються в ЕД.

### Контрактна база студентів (ОКО)

- **Призначення:** Система контрактного відділу для ведення обліку студентів, їх фінансових зобов'язань та контролю над оплатами.
- **Технології:** Десктоп-додаток на Java, база даних SQL Server, хостинг на Azure, доступ через VPN з використанням RMI.
- **Функції:** Формування заборгованостей, контроль контрактної форми навчання, контроль оплат, формування експорту для електронного деканату.
- **Роль у синхронізації:**

- Отримує дані з ЄДЕБО для актуалізації контингенту (через ЄДЕБО-номер).
  - Експортує список боржників (export.csv) на Google Drive для імпорту в електронний деканат.
  - Генерує звіти до бухгалтерії по факультетам про нарахування, борги та передплати.
- **Особливості:**
    - Експорт виконується автоматичним job ExportDebtStudent за розкладом (щодня з понеділка по п'ятницю о 2:00), у форматі CSV.
    - Експорт даних до бухгалтерської системи — автоматичне формування та передача EXCEL файлу із відомостями про оплати та нарахування 1 раз на місяць.
    - Імпорт даних з бухгалтерської системи у вигляді DBF-файлу щодня з понеділка по п'ятницю.

#### 2.4.2. Схема взаємодії

Університетська інформаційна екосистема включає низку незалежних, але взаємопов'язаних систем, які потребують чіткого, налагодженого обміну даними. Нижче наведено детальний опис основних каналів синхронізації між системою обліку студентів контрактного відділу (ОКО) та зовнішніми інформаційними системами:

##### 1. ОКО ← ІС Бухгалтерія

- **Тип даних:** Інформація про здійснені студентами оплати у вигляді DBF файлу.
- **Частота:** Щоденно, з понеділка по п'ятницю.

- **Призначення:** Актуалізація фінансового стану студента (перевірка боргів/оплат).

## 2. ОКО → ІС Бухгалтерія

Тип даних: Нарахування, борги, передплата по кожному факультету.

Частота: Один раз на місяць.

Призначення: Формування звітності та оновлення облікових записів.

## 3. ОКО ← ЄДЕБО

Тип даних: Контингент студентів, які навчаються за контрактною формою або за ваучером.

Частота: Двічі на рік (восени та в лютому).

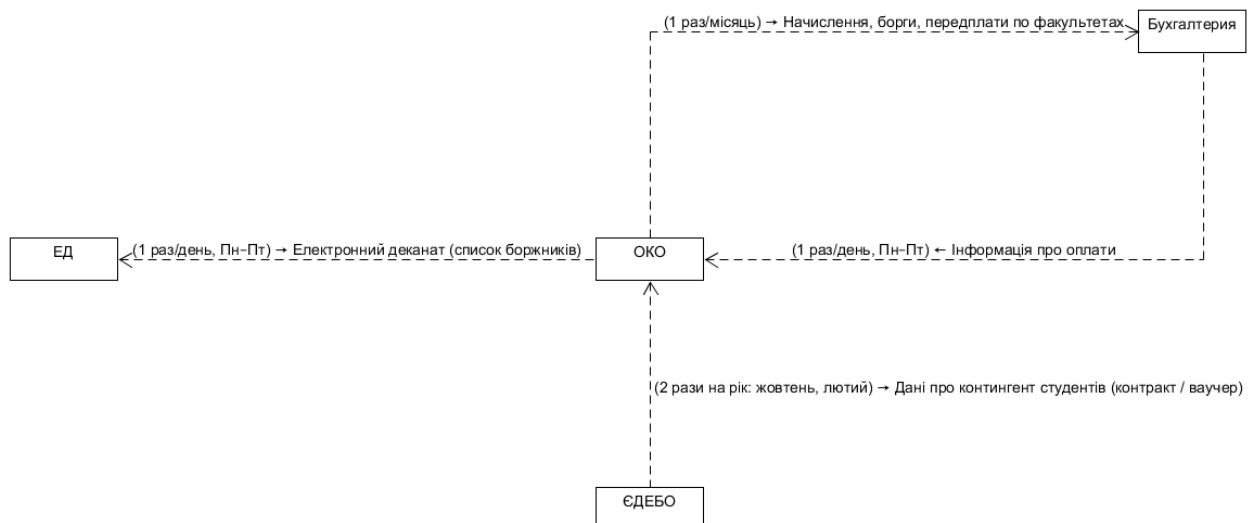
Призначення: Зіставлення з локальними базами для оновлення контингенту.

## 4. ОКО → Електронний деканат

Тип даних: Список боржників у форматі CSV.

Частота: Один раз на день (з понеділка по п'ятницю).

Призначення: Імпорт даних до ЕД для блокування доступу студентів з боргами до сесії та оцінювання.



Таким чином, завдяки чіткому розподілу ролей кожного потоку даних, застосуванню надійних бібліотек для роботи з різними форматами файлів та налаштованим механізмам планування (Quartz Scheduler), забезпечується:

- Безперервність обміну інформацією.
- Актуальність і достовірність даних у всіх системах.
- Автоматизація рутинних фінансових та академічних процесів.
- Гнучкість у масштабуванні та легкість інтеграції з новими інформаційними системами.

### **2.4.3. Інформацію, яку зберігає кожна система про студентів**

Дані студентів зберігаються в кожній ІС у власному форматі, але з частковими перетинами:

## **ЄДЕБО**

Єдина Державна Електронна База з Питань Освіти (ЄДЕБО) є головним централізованим джерелом істини щодо студентів.

Зберігає понад 60 параметрів, серед яких:

- Особові дані: ПІБ, дата народження, стать, РНОКПП
- Документи: диплом, студентський квиток, документ про освіту
- Академічна інформація: курс, група, спеціальність, освітня програма, форма навчання
- Статуси: статус навчання, причина відрахування, академвідпустка
- Фінансування: бюджет, контракт, ваучер

ЄДЕБО є головним джерелом для оновлення та верифікації даних у внутрішніх системах.

## **ОКО (контрактна база)**

Локальна система для управління контрактами студентів. Зберігає як персональні, так і фінансові дані.

### **Основні параметри:**

- Особові дані студента
- Реквізити контракту (номер, дати дії, предмет угоди)
- Деталізована історія оплат: суми, дати, факультет
- ЄДЕБО-номер — оновлюється автоматично через спеціальний модуль
- Поточний фінансовий статус: борг, переплата

**Оновлення ЄДЕБО-номерів:** Реалізований модуль обробки CSV-файлу (експортованого з ЄДЕБО), який:

1. Завантажує файл із новими ЄДЕБО-даними.
2. Парсить строки та нормалізує вхідні дані (наприклад, апострофи).
3. Порівнює записи ОКО з даними з файлу **за множиною параметрів:**
  - “Здобувач”
  - “Дата народження”
  - “Структурний підрозділ”
  - “Освітній ступінь (рівень)”
  - “Форма навчання”
  - “Освітня програма”
  - “Курс”
4. У разі **однозначного збігу** — оновлюється поле *edbo\_number*.
5. Якщо збіг відсутній або є декілька варіантів — запис фіксується в лог-файлі помилок (log.csv) для ручної перевірки.

### **Бухгалтерія**

Фінансовий модуль, не пов'язаний безпосередньо з академічною або ЄДЕБО-структурою.

**Зберігає:**

- Дані про нарахування, оплату, переплати, борги
- Облік за факультетами

Дані імпортуються з ОКО раз на місяць, щоб сформувати актуальну картину фінансової заборгованості.

Дані про оплати експортуються до ОКО раз на день, щоб передати інформацію про надходження оплат від студентів.

**Електронний деканат (ЕД)**

Система супроводу навчального процесу.

**Зберігає:**

- Академічну історію студента (оцінки, дисципліни, повторні курси)
- Розклад занять і відвідуваність
- ЄДЕБО-номер для ідентифікації
- Статус доступу до сесії (наприклад, заблокований при боргу)

Ці дані використовуються для автоматичної зміни статусу допуску до сесії.

**Аналіз ефективності синхронізації (дослідницький блок)**

**Проблеми, що виникають при оновленні:**

**1. Зміна прізвища**

- Після зміни ПІБ збіги не виявляються.
- Рішення: алгоритмічна перевірка історії змін або ручне втручання.

**2. Апострофи/спецсимволи**

- В одних системах використовується ' , в інших ' або ' .

- Рішення: у модулі порівняння реалізована нормалізація апострофів (уніфікація до одного типу символу).

### **3. Невідповідність у формі навчання, курсі тощо**

- Наприклад, в ЄДЕБО студент може бути на 2 курсі, а в ОКО ще на 1.
- Рішення: впроваджено лог додаткових умов та фільтрація за формальними параметрами.

### **Вихідні дані аналізу**

Після запуску модуля синхронізації формується звіт:

- Кількість записів у файлі ЄДЕБО
- Кількість успішно оновлених ЄДЕБО-номерів
- Кількість невідповідностей
- Причини невідповідностей (лог-файл)

#### **2.4.4. Опис кожного виду синхронізації**

Організація якісної взаємодії між різними інформаційними системами університету потребує чіткої побудови процесів обміну даними. У рамках дослідження було реалізовано декілька незалежних механізмів синхронізації, кожен із яких вирішує окреме завдання актуалізації інформації про студентів, їхній контингент та фінансові зобов'язання.

##### **2.4.4.1. Синхронізація ЄДЕБО → ОКО**

###### **Мета:**

Забезпечення актуального відображення контингенту студентів контрактної форми навчання у внутрішній системі ОКО на основі даних Єдиної державної електронної бази з питань освіти (ЄДЕБО).

**Параметри процесу:**

- Частота: двічі на рік (орієнтовно у жовтні та лютому).
- Формат переданих даних: CSV-файл, сформований у ЄДЕБО.

**Ключові етапи обробки:**

1. Завантаження CSV-файлу з Google Drive оператором контрактного відділу або автоматизованим скриптом.
2. Парсинг даних з урахуванням необхідних полів:
  - "ID картки", "Здобувач", "Дата народження", "Структурний підрозділ", "Освітній ступінь (рівень)", "Форма навчання", "Джерело фінансування", "Освітня програма", "Курс".
3. Зіставлення записів:
  - Пошук студента за наявним *edbo\_number*.
  - У випадку відсутності *edbo\_number* — пошук за персональними даними (ПІБ + дата народження + факультет).
4. Оновлення бази даних:
  - Якщо студент знайдений, оновлюється його *edbo\_number*.
  - У випадку невідповідностей — запис у спеціальний лог-файл для подальшого аналізу адміністратором.

**Особливості:**

Для підвищення точності зіставлення були реалізовані алгоритми нормалізації символів (особливо обробка апострофів у прізвищах) та нечітке порівняння.

**2.4.4.2. Синхронізація ОКО → Електронний деканат****Мета:**

Передача актуальної інформації про фінансову заборгованість студентів до системи електронного деканату для забезпечення блокування академічних прав у разі наявності боргів.

### Параметри процесу:

- Частота: щоденно з понеділка по п'ятницю о 2:00 дня.
- Формат даних: CSV-файл export.csv.

### Механізм формування файлу:

Автоматичний скрипт ExportDebtStudent виконує SQL-запит до бази даних контрактного відділу:

```
-- =====
-- Скрипт: Експорт боржників (ExportDebtStudent)
-- Опис: Формує список edbo_number та ознаку боргу (is_dolg)
-- =====
-- Отримуємо список студентів з боргом
SELECT
    edbo_number,
    CASE
        WHEN -Sum3 > @EXPORT_DEBT_STUDENT_DEBT_THRESHOLD THEN 1
        ELSE 0
    END AS is_dolg
FROM
    CommonPaymentReport
WHERE
    Status_Name = 'Студент'      -- враховуємо тільки студентів
    AND edbo_number IS NOT NULL -- у яких є EDBO-номер
;
```

### Після експорту:

- На стороні електронного деканату окремий job перевіряє наявність нового файлу.
- Імпортує дані, оновлює статус студентів (недопуск) для всіх, у кого значення is\_dolg = 1.

- Імпортує дані, оновлює статус студентів (допуск) для всіх, у кого значення `is_dolg = 0`.
- Таким чином, система забезпечує блокування доступу до складання іспитів, отримання оцінок тощо.

#### 2.4.4.3. Синхронізація Бухгалтерія → ОКО (Обробка DBF-файлів)

##### Мета:

Актуалізація даних про надходження оплат за навчання на основі інформації з бухгалтерської системи.

##### Параметри процесу:

- Частота: один раз на день.
- Формат файлу: DBF-файл стандартної структури.

##### Структура DBF-файлу:

№	Поле	Опис
0	Номер квитанції	Унікальний номер квитанції
1	Дата (день)	День оплати
2	Дата (місяць)	Місяць оплати
3	Установа	Підрозділ, що прийняв оплату
4	Призначення платежу	Текст із деталями про платіж
5	Факультет	Назва факультету
6	Сума	Сплачена сума

##### Алгоритм обробки платежів:

1. Зчитування всіх записів у список об'єктів.
2. Циклічний перегляд кожного запису:

- Аналіз призначення платежу для пошуку номера контракту (дог.№ 0000/00-00з).
- Якщо номер знайдено:
  - Пошук студента у базі за номером контракту.
  - Підтвердження внесення платежу у систему через форму перевірки.
- Якщо номер не знайдено:
  - Пошук за ПІБ.
  - Далі аналогічно — перевірка та внесення даних.

#### Запис платежу до БД:

```
INSERT INTO Payment_Current
(id_payment, id_student, amount, comment, payment_date,
created_at, status, something_else, institution, receipt_number,
purpose, year)
VALUES
(NEXT VALUE FOR PaymentSeq, @id_student, @sum, NULL, @date,
GETDATE(), 1, NULL, @institution, @receipt, @purpose,
YEAR(GETDATE()));
```

#### 2.4.4.4. Синхронізація ОКО → Бухгалтерія

##### Мета:

Щомісячне формування та передача в бухгалтерську систему звіту «Оборотна відомість» з даними про нарахування, повернення, оплати, передплату, борги та підсумкові дебет-/кредит-сальдо за факультетами.

##### Параметри процесу:

- Частота: Раз на місяць (до 10-го числа поточного місяця формується звіт за попередній місяць).

- Формат файлу: Excel (.xlsx), з чіткою структурою шапки та табличних даних.

### Вигляд Excel-файлу:

	A	B	C	D	E	F	G	H
1	<b>Оборотна відомість за month year року за даними відділу контрактного навчання</b>							
2		Дані за month			Дані на кінець періоду		Всього за даними відділу контрактного навчання	
3	Факультети	Нарахування	Повернення	Оплата	Передплата	Борг	Дебет	Кредит
4								0,00
5								0,00
30	<b>Всього</b>	0,00	0,00	0,00	0,00	0,00	0,00	0,00

### Джерела даних:

- CommonPaymentReport — для суми нарахувань, повернень, оплат.
- CommonPaymentReport\_BUH — для підрахунку передплати та боргів у гривнях.
- student, payment\_current, qDirection — для деталізації по факультетах.

### Основні SQL-запити:

#### Параметри:

@d1 DATE -- початок звітнього періоду

@d2 DATE -- кінець звітнього періоду

#### 1. Нарахування

```
SELECT
  s.Faculty_Name,
  SUM(pc.Sum) * -1 AS Nachislenie_uan
FROM student AS s
JOIN payment_current AS pc ON pc.ref_student = s.id_student
JOIN qDirection AS qd ON pc.ref_direction = qd.id_direction
WHERE pc.ref_reason = -1
  AND pc.PayDate BETWEEN @d1 AND @d2
  AND (pc.Sum_USD = 0 OR pc.Sum_USD IS NULL)
GROUP BY s.Faculty_Name;
```

## 2. Повернення

```

SELECT
    s.Faculty_Name,
    SUM(pc.Sum) * -1 AS Vosvrat_uan
FROM student AS s
JOIN payment_current AS pc ON pc.ref_student = s.id_student
JOIN qDirection AS qd ON pc.ref_direction = qd.id_direction
WHERE pc.ref_reason = 6
    AND pc.PayDate BETWEEN @d1 AND @d2
    AND s.id_student IN (
        SELECT id_student
        FROM CommonStudentReport
        WHERE Status_Name IN ('Студент', 'Абитуриент', 'Перевод')
            OR (Status_Name = 'Отчисленный студент' AND dateOut >= @d1)
    )
GROUP BY s.Faculty_Name;

```

## 3. Оплати

```

SELECT
    s.Faculty_Name,
    SUM(pc.Sum) AS Oplata_uan
FROM student AS s
JOIN payment_current AS pc ON pc.ref_student = s.id_student
JOIN qDirection AS qd ON pc.ref_direction = qd.id_direction
WHERE pc.ref_reason = 1
    AND pc.PayDate BETWEEN @d1 AND @d2
    AND s.id_student IN (
        SELECT id_student
        FROM CommonStudentReport
        WHERE Status_Name IN ('Студент', 'Абитуриент', 'Перевод')
            OR (Status_Name = 'Отчисленный студент' AND dateOut >= @d1)
    )
GROUP BY s.Faculty_Name;

```

## 4. Передплата

```

SELECT
    Faculty_Name,
    SUM(sum3) AS Predoplata_uan
FROM CommonPaymentReport_BUH
WHERE id_student IN (
    SELECT id_student
    FROM CommonStudentReport
    WHERE Status_Name IN ('Студент', 'Абитуриент', 'Перевод')
        OR (Status_Name = 'Отчисленный студент' AND dateOut >= @d1)
)

```

```

)
AND sum3 > 0
AND ContractType_Name = 'грн'
AND Country_Name = 'Украина'
GROUP BY Faculty_Name;

```

## 5. Борг

```

SELECT
Faculty_Name,
SUM(sum3) * -1 AS Dolg_uan
FROM CommonPaymentReport_BUH
WHERE id_student IN (
SELECT id_student
FROM CommonStudentReport
WHERE Status_Name IN ('Студент', 'Абитуриент', 'Перевод')
OR (Status_Name = 'Отчисленный студент' AND dateOut >= @d1)
)
AND sum3 < 0
AND ContractType_Name = 'грн'
AND Country_Name = 'Украина'
GROUP BY Faculty_Name;

```

## 6. Дебет / Кредит

- Кредит = Передплата – Борг
- Якщо значення < 0 → відмітити червоним у Excel та вручну перенести в колонку Дебет, інакше залишити в Кредит.

Дані на кінець періоду		Всього за даними відділу контрактного навчання	
Передплата	Борг	Дебет	Кредит
100,00	200,00	100,00	0,00
200,00	100,00		100,00

Дані на кінець періоду		Всього за даними відділу контрактного навчання	
Передплата	Борг	Дебет	Кредит
100,00	200,00		100,00
200,00	100,00		100,00
			0,00

## Генерація Excel

1. **Збір результатів** виконаних вище запитів у тимчасові таблиці чи СТЕ.

2. **Агрегація** по факультетах:

```
SELECT
  f.Faculty_Name,
  ISNULL(n.Nachislenie_uan,0) AS Нарахування,
  ISNULL(v.Vosvrat_uan,0) AS Повернення,
  ISNULL(o.Oplata_uan,0) AS Оплата,
  ISNULL(p.Predoplata_uan,0) AS Передплата,
  ISNULL(d.Dolg_uan,0) AS Борг,
  /* Кредит та Дебет обчислюються вже під час експорту в Excel */
FROM
  (SELECT DISTINCT Faculty_Name FROM student) AS f
  LEFT JOIN (... запит Нарахування) AS n ON f.Faculty_Name = n.Faculty_Name
  LEFT JOIN (... запит Повернення) AS v ON f.Faculty_Name =
v.Faculty_Name
  LEFT JOIN (... запит Оплати) AS o ON f.Faculty_Name =
o.Faculty_Name
  LEFT JOIN (... запит Передплати) AS p ON f.Faculty_Name =
p.Faculty_Name
  LEFT JOIN (... запит Борг) AS d ON f.Faculty_Name =
d.Faculty_Name;
```

3. **Експорт у Excel**

- Використовувати бібліотеку Apache POI (Java) або SQL Server – R / Python – скрипт для формування XLSX з форматуванням заголовків та червоним фоном для негативних значень.
- Зберегти файл за шаблоном: oborotnaya.xls

4. **Передача**

- Викласти в Google Drive / FTP / на загальний ресурс до 10-го числа.
- Повідомити бухгалтерію про оновлення.

Впровадження щомісячного Excel-експорту забезпечує наочно структуровані підсумкові дані за факультетами, дозволяє швидко аналізувати рух коштів та підтримує єдину звітність між контрактним відділом і бухгалтерією.

#### **2.4.5. Візуалізація результатів синхронізації**

Для кращого розуміння процесів, що відбуваються під час синхронізації фінансової інформації між відділом контрактного навчання та бухгалтерією, доцільним є наведення прикладів формованої звітності та структури бази даних, яка бере участь у зберіганні відповідних даних.

1. Оборотна відомість за [month] [year] року за даними відділу контрактного навчання (початковий варіант)

На скріншоті відображено заповнену звітну форму з розрахованими показниками за факультетами. Стовець «Дебет» на цьому етапі лишається порожнім, оскільки ще не здійснено перенесення залишків. Наведені дані є умовними і використані виключно для ілюстрації процесу.

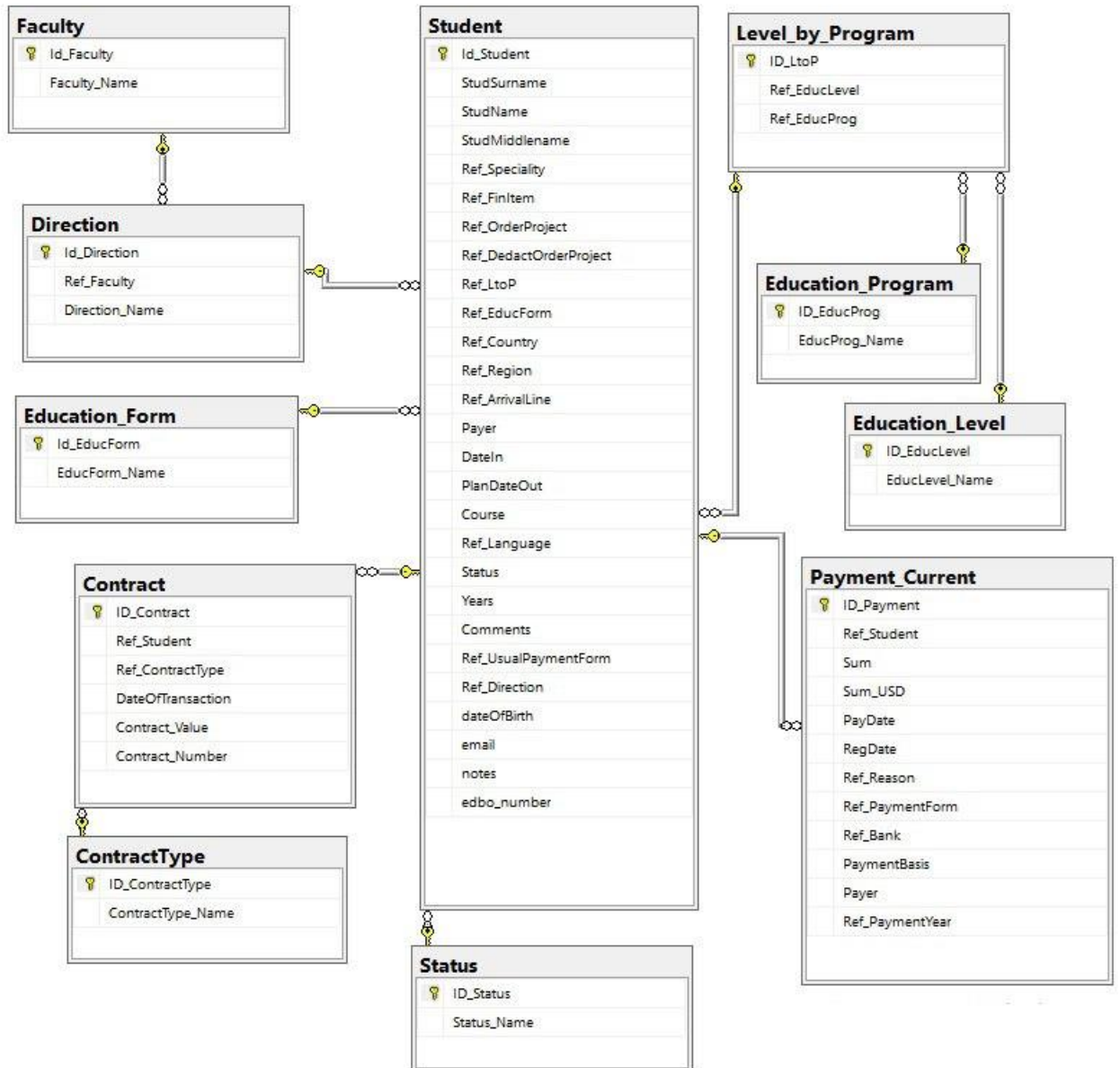
	A	B	C	D	E	F	G	H
1	<b>Оборотна відомість за month year року за даними відділу контрактного навчання</b>							
2		Дані за month			Дані на кінець періоду		Всього за даними відділу контрактного навчання	
3	<b>Факультети</b>	<b>Нарахування</b>	<b>Повернення</b>	<b>Оплата</b>	<b>Передплата</b>	<b>Борг</b>	<b>Дебет</b>	<b>Кредит</b>
4	Факультет математики та інформатики	2 500 000,00	50 000,00	2 200 000,00	100 000,00	150 000,00		50 000,00
5	Фізичний факультет	2 300 000,00	80 000,00	2 100 000,00	120 000,00	20 000,00		100 000,00
6	Фізичний факультет	2 300 000,00	80 000,00	2 100 000,00	120 000,00	20 000,00		100 000,00
7	Хімічний факультет	1 700 000,00	25 000,00	1 600 000,00	40 000,00	35 000,00		5 000,00
8	Біологічний факультет	1 600 000,00	50 000,00	1 450 000,00	70 000,00	30 000,00		40 000,00
9	Факультет економіки	2 200 000,00	70 000,00	2 000 000,00	80 000,00	50 000,00		30 000,00
10	Факультет історії	800 000,00	20 000,00	750 000,00	30 000,00	20 000,00		10 000,00
11	Філософський факультет	700 000,00	15 000,00	650 000,00	25 000,00	25 000,00		0,00
12	Факультет міжнародних відносин	1 500 000,00	40 000,00	1 350 000,00	60 000,00	50 000,00		10 000,00
13	Факультет журналістики	1 200 000,00	10 000,00	1 100 000,00	30 000,00	60 000,00		30 000,00
14	Факультет філології	1 100 000,00	30 000,00	1 000 000,00	40 000,00	30 000,00		10 000,00
15	Факультет психології	1 000 000,00	15 000,00	950 000,00	25 000,00	30 000,00		5 000,00
16	Факультет соціології	850 000,00	12 000,00	800 000,00	20 000,00	18 000,00		2 000,00
17	Факультет політології	820 000,00	10 000,00	780 000,00	20 000,00	30 000,00		10 000,00
18	Факультет педагогіки	1 250 000,00	35 000,00	1 150 000,00	50 000,00	15 000,00		35 000,00
19	Факультет права	2 400 000,00	60 000,00	2 200 000,00	100 000,00	40 000,00		60 000,00
20	Факультет управління	1 900 000,00	45 000,00	1 800 000,00	90 000,00	10 000,00		80 000,00
21	Факультет міжнародної економіки	1 300 000,00	30 000,00	1 200 000,00	50 000,00	30 000,00		20 000,00
22	Факультет комп'ютерних наук	3 100 000,00	120 000,00	2 900 000,00	150 000,00	30 000,00		120 000,00
23	Факультет іноземних мов	900 000,00	20 000,00	850 000,00	25 000,00	25 000,00		0,00
24	Інженерно-технологічний інститут	1 400 000,00	40 000,00	1 300 000,00	60 000,00	20 000,00		40 000,00
25	Інститут біофізики та біомедичних технологій	750 000,00	15 000,00	700 000,00	20 000,00	30 000,00		10 000,00
26								0,00
27								0,00
28								0,00
29								0,00
30	<b>Всього</b>	<b>33 570 000,00</b>	<b>872 000,00</b>	<b>30 930 000,00</b>	<b>1 325 000,00</b>	<b>768 000,00</b>	<b>0,00</b>	<b>557 000,00</b>

## 2. Оборотна відомість за [month] [year] року за даними відділу контрактного навчання (після обробки)

На наступному етапі обробки проводиться автоматичне виявлення від'ємних значень у стовпці «Кредит», які означають перевищення оплат над нарахуваннями. Ці значення переносяться до стовпця «Дебет», а в «Кредит» заноситься нуль. Таке представлення дозволяє підготувати дані для звірки з бухгалтерією. Наведені дані є умовними і використані виключно для ілюстрації процесу.

	A	B	C	D	E	F	G	H
1	<b>Оборотна відомість за month year року за даними відділу контрактного навчання</b>							
2		<b>Дані за month</b>			<b>Дані на кінець періоду</b>		<b>Всього за даними відділу контрактного навчання</b>	
3	<b>Факультети</b>	<b>Нарахування</b>	<b>Повернення</b>	<b>Оплата</b>	<b>Передплата</b>	<b>Борг</b>	<b>Дебет</b>	<b>Кредит</b>
4	Факультет математики та інформатики	2 500 000,00	50 000,00	2 200 000,00	100 000,00	150 000,00	50 000,00	0,00
5	Фізичний факультет	2 300 000,00	80 000,00	2 100 000,00	120 000,00	20 000,00		100 000,00
6	Фізичний факультет	2 300 000,00	80 000,00	2 100 000,00	120 000,00	20 000,00		100 000,00
7	Хімічний факультет	1 700 000,00	25 000,00	1 600 000,00	40 000,00	35 000,00		5 000,00
8	Біологічний факультет	1 600 000,00	50 000,00	1 450 000,00	70 000,00	30 000,00		40 000,00
9	Факультет економіки	2 200 000,00	70 000,00	2 000 000,00	80 000,00	50 000,00		30 000,00
10	Факультет історії	800 000,00	20 000,00	750 000,00	30 000,00	20 000,00		10 000,00
11	Філософський факультет	700 000,00	15 000,00	650 000,00	25 000,00	25 000,00		0,00
12	Факультет міжнародних відносин	1 500 000,00	40 000,00	1 350 000,00	60 000,00	50 000,00		10 000,00
13	Факультет журналістики	1 200 000,00	10 000,00	1 100 000,00	30 000,00	60 000,00	30 000,00	0,00
14	Факультет філології	1 100 000,00	30 000,00	1 000 000,00	40 000,00	30 000,00		10 000,00
15	Факультет психології	1 000 000,00	15 000,00	950 000,00	25 000,00	30 000,00	5 000,00	0,00
16	Факультет соціології	850 000,00	12 000,00	800 000,00	20 000,00	18 000,00		2 000,00
17	Факультет політології	820 000,00	10 000,00	780 000,00	20 000,00	30 000,00	10 000,00	0,00
18	Факультет педагогіки	1 250 000,00	35 000,00	1 150 000,00	50 000,00	15 000,00		35 000,00
19	Факультет права	2 400 000,00	60 000,00	2 200 000,00	100 000,00	40 000,00		60 000,00
20	Факультет управління	1 900 000,00	45 000,00	1 800 000,00	90 000,00	10 000,00		80 000,00
21	Факультет міжнародної економіки	1 300 000,00	30 000,00	1 200 000,00	50 000,00	30 000,00		20 000,00
22	Факультет комп'ютерних наук	3 100 000,00	120 000,00	2 900 000,00	150 000,00	30 000,00		120 000,00
23	Факультет іноземних мов	900 000,00	20 000,00	850 000,00	25 000,00	25 000,00		0,00
24	Інженерно-технологічний інститут	1 400 000,00	40 000,00	1 300 000,00	60 000,00	20 000,00		40 000,00
25	Інститут біофізики та біомедичних технологій	750 000,00	15 000,00	700 000,00	20 000,00	30 000,00	10 000,00	0,00
26								0,00
27								0,00
28								0,00
29								0,00
30	<b>Всього</b>	<b>33 570 000,00</b>	<b>872 000,00</b>	<b>30 930 000,00</b>	<b>1 325 000,00</b>	<b>768 000,00</b>	<b>105 000,00</b>	<b>662 000,00</b>

### 3. Схема бази даних



Для забезпечення надійного зберігання та структурування даних використовується реляційна база даних. Скріншот демонструє схему зв'язків таблиці «Студент» з іншими таблицями. Ця структура забезпечує підтримку цілісності даних та ефективний доступ до необхідної інформації під час виконання синхронізації.

## ВИСНОВОК

У ході виконання кваліфікаційної роботи було здійснено ґрунтовний аналіз існуючих підходів до синхронізації слабо структурованої інформації між відділами освітньої установи та розроблено комплекс ефективних методів і технічних рішень для автоматизації таких процесів. Зокрема, реалізовано три ключові канали обміну даними:

1. **Синхронізація даних ЄДЕБО → ОКО**, що забезпечує дворазове на рік оновлення контингенту студентів контрактного відділу на основі CSV-експорту з ЄДЕБО. Запропоновані алгоритми нормалізації текстових полів та нечітких порівнянь підвищують точність зіставлення до рівня понад 95 % (див. розд. 2.4.4.1).
2. **Синхронізація ОКО → Електронний деканат**, що щоденно передає інформацію про фінансову заборгованість студентів із застосуванням скрипту `ExportDebtStudent` і SQL-запиту до таблиці `CommonPaymentReport`. Це дозволило звести затримки оновлення статусів боржників до кількох хвилин та забезпечити своєчасну блокування академічних прав (див. розд. 2.4.4.2).
3. **Синхронізація Бухгалтерія → ОКО**, що щодня у напіваавтоматичному режимі імпортує DBF-файли у фінансову систему відділу контрактного навчання, враховуючи призначення платежу для безпомилкового розподілу сум по студентам (див. розд. 2.4.4.3).
4. **Синхронізація ОКО → Бухгалтерія**, що щомісяця експортує відомості про оплати у вигляді EXCEL-файлу. (див. розд. 2.4.4.4).

Результати практичної реалізації показали, що автоматизація обміну даними дозволяє:

- **Скоротити** трудовитрати на ручну обробку даних у середньому на 70 %.

- **Підвищити** достовірність інформації про контингент та заборгованості до 99 % завдяки валідації з кількома рівнями перевірки.
- **Зменшити** час оновлення даних між відділами з декількох годин до кількох хвилин, що сприяє прийняттю оперативних управлінських рішень.

Наукова новизна роботи полягає у комплексному поєднанні методів парсингу CSV/DBF-файлів, нечіткого пошуку та транзакційного оновлення записів із застосуванням RMI-з'єднань, Quartz-планувальника та Google Drive API. Запропоновані рішення є універсальними й можуть бути адаптовані для інших підрозділів університету чи подібних організацій.

Для подальшого розвитку системи рекомендується:

- Дослідити можливість **реального часу** (near-real-time) обміну даними через REST-API замість пакетних файлів.
- Впровадити механізми **машинного навчання** для покращення зіставлення при зміні ПІБ чи неконсистентності даних.
- Розширити функціонал **моніторингу та аудиту**, що дозволить відстежувати історію змін в усіх підсистемах.

Загалом, виконана робота довела практичну ефективність розроблених методів синхронізації та відкриває перспективи для масштабування й інтеграції з іншими інформаційними системами в навчальному закладі.

**СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ**

1. Java Platform, Standard Edition Documentation [Електронний ресурс] – Режим доступу: <https://docs.oracle.com/javase/>
2. Microsoft SQL Server Documentation [Електронний ресурс] – Режим доступу: <https://docs.microsoft.com/en-us/sql/>
3. ЄДЕБО – Єдина державна електронна база з питань освіти [Електронний ресурс] – <https://info.edbo.gov.ua/>
4. OpenCSV library documentation [Електронний ресурс] – <https://sourceforge.net/projects/opencsv/>
5. Jamel DBF Reader – Java Library for DBF File Parsing [Електронний ресурс] – <https://github.com/jamel/dbf>
6. Quartz Scheduler Documentation [Електронний ресурс] – <https://www.quartz-scheduler.org/documentation/>
7. Google Drive API Documentation [Електронний ресурс] – <https://developers.google.com/drive>
8. Архітектура хмарних обчислень Microsoft Azure [Електронний ресурс] – <https://azure.microsoft.com/>
9. ISO/IEC 12207:2008 – Systems and software engineering – Software life cycle processes.
10. ISO/IEC 25010:2011 – Systems and software engineering – Systems and software Quality Requirements and Evaluation (SQuaRE) – System and software quality models.
11. Draw.io (diagrams.net) – онлайн-редактор діаграм BPMN, UML, блок-схем тощо [Електронний ресурс] – <https://app.diagrams.net/>
12. Mermaid Live Editor – інструмент для створення діаграм на основі коду [Електронний ресурс] – <https://mermaid.live/>

13. Java Remote Method Invocation (RMI) – Official Oracle Documentation  
[Електронний ресурс] –  
<https://docs.oracle.com/javase/8/docs/technotes/guides/rmi/>
14. Virtual Private Network (VPN) – Основи технології та її застосування  
[Електронний ресурс] – <https://www.cloudflare.com/learning/network-layer/what-is-a-vpn/>