5. Programm.top (C#/.NET). URL: https://programm.top/uk/c-sharp/tutorial/recursion/ (дата звернення: 28.11.2025).

УДК 004.65

Бородій І. І., аспірант,
Осухівська Г. М., канд. техн. наук,
доцент, завідувач кафедри
комп'ютерних систем та мереж

ФОРМУВАННЯ ВХІДНИХ ДАНИХ ДЛЯ АНАЛІТИЧНИХ СИСТЕМ НА ОСНОВІ АРХІТЕКТУР DATA LAKEHOUSE

Тернопільський національний технічний університет імені Івана Пулюя, м. Тернопіль

Для побудови аналітичних систем даних провідною технологією є концепція Data Lakehouse, яка поєднує переваги Data Lake та Data Warehouse. Вона забезпечує гнучке зберігання структурованих і напівструктурованих типів даних, ефективне управління метаданими та оптимізоване виконання аналітичних запитів. У контексті реалізації найпоширенішими систем Lakehouse є табличні формати Apache Hudi, Apache Iceberg та Delta Lake.

Apache Iceberg – це продукт компанії Netflix, що є табличним форматом з відкритим кодом, призначений для роботи з великими обсягами аналітичних даних у сховищах типу Data Lakehouse. Apache Hudi є системою збереження даних, орієнтованого на роботу з великими обсягами інформації, що постійно змінюється. Delta Lake є технологією для забезпечення транзакційної цілісності та контролю версій у хмарних системах зберігання даних.

У сучасних роботах розглядають можливості Data Lakehouse архітектур, зокрема у межах порівняння їх продуктивності та особливостей обробки даних [1]. Ефективність використання Lakehouse-підходів у масштабованих аналітичних системах підтверджена в роботі [2]. Авторами дослідження [3] було показано, що вибір способу реалізації ETL-процесів суттєво впливає на продуктивність обробки даних у середовищі Apache Spark.

Метою дослідження є формування вхідних даних для аналітичних систем на основі архітектур Data Lakehouse, щоб здійснити вибір найоптимальнішої її реалізації за критеріями швидкості завантаження даних та ефективності збереження у дисковому просторі.

Формування вхідних наборів даних для аналітичних систем на основі архітектур Data Lakehouse здійснювалося з декількох джерел. Зокрема, вебресурс open-meteo надає погодинні показники температури атмосферного повітря за 2022–2024 рр. з прив'язкою до географічних координат. Цей структурований масив даних було завантажено у файл формату CSV обсягом 7 ГБ. Додатково у дослідженні використано CSV-файл із вебресурсу simplemaps.com, який містить перелік міст та їх координати, розміром приблизно 5 МБ. Напівструктуровані дані у форматі JSON отримано з вебресурсу iqair.com, що містить інформацію про показники

стану якості атмосферного повітря залежно від координат. Завантаження даних здійснювалося в окремі таблиці Data Lakehouse. Зокрема, таблиця weather_events отримує дані з open-meteo у вигляді файла формату CSV та поділена на партиції за полем дати (табл. 1). Таблиця world_cities включає інформацію з simplemaps.com у CSV-форматі та поділена на партиції за полем країни у форматі ISO 2 (табл. 2).

Таблиця 1 – Сутність weather_events

Назва поля	**Тип даних**	**Поле є партицією?**
lat	double	Ні
lng	double	Ні
temperature_ts	timestamp	Ні
temperature	double	Ні
temperature_dt	date	Так

Таблиця 2 – Сутність world_cities

Назва поля	**Тип даних**	**Поле є партицією?**
country	string	Ні
province	string	Ні
city	string	Ні
capital	string	Ні
population	bigint	Ні
lat	double	Ні
lng	double	Ні
country_iso2	string	Так

Таблиця air_quality_level зберігає дані, отримані з AirVisual API, у форматі JSON (табл. 3). Через складну структуру джерела таблиця містить поля типу struct, які відображають вкладені об’єкти, зокрема показники погоди та забруднення повітря. Таблиця партиціонується за полем dt, що забезпечує ефективність часових запитів.

Усі дані були завантажені в таблиці кожної системи Data Lakehouse з застосуванням процесу ETL для аналізу обсягу сформованих таблиць та оцінки ефективності використання дискового простору кожною системою.

У межах цього дослідження вилучення даних полягало у зчитуванні даних із вхідних файлів. Під час виконання етапу трансформації дані та їх структура були модифіковані відповідно до вимог дослідження за допомогою засобу обробки даних Apache Spark. Завершальний етап завантаження передбачав перенесення підготовлених даних до таблиць Data Lakehouse. Для кожної досліджуваної системи Data Lakehouse було сформовано окрему схему бази даних, у межах якої створено таблиці для завантаження даних. Кожна таблиця відповідає різним аспектам аналітики даних.

Таблиця 3 – Сутність air_quality_level

Назва поля	**Тип даних**	**Поле є партицією?**
city	string	Ні
state	string	Ні
country	string	Ні
location	struct	Ні
current	struct	Ні
lng	double	Ні
lat	double	Ні
ts	timestamp	Ні
dt	date	Так

Результатами для кожної з систем Data Lakehouse було створено окрему директорію для збереження даних із таблиць. Всередині цієї директорії розміщуються інші директорії – по одній для кожної з таблиць. Наприклад, оскільки в дослідженні використовується три таблиці, то всередині директорії будуть створені три інші піддиректорії. Організація структури таблиць для кожної з систем Lakehouse є ідентичною. Наприклад, структура для Apache Iceberg відображена на рис. 1.

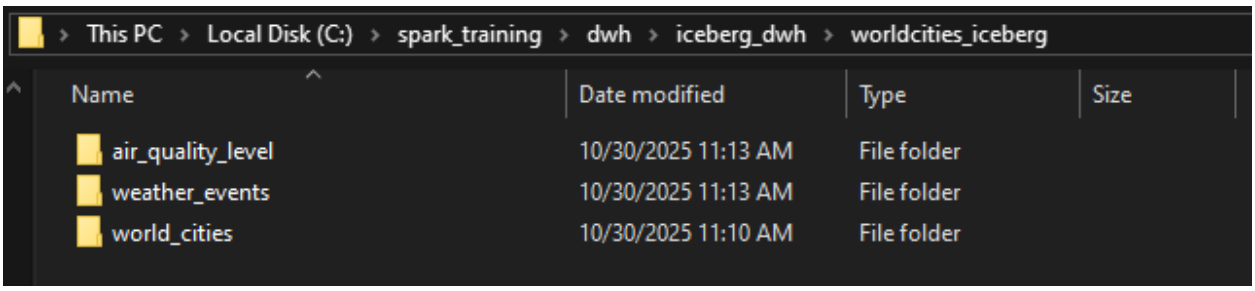

Рисунок 1 – Організація таблиць схеми Apache Iceberg

Партиції в системах Lakehouse теж організовані у вигляді файлової системи. Кожна з систем Data Lakehouse має свої особливості організації та збереження даних, які необхідно враховувати під час побудови аналітичних систем.

У цьому дослідженні описано формування вхідних даних для аналітичних систем на основі архітектур Data Lakehouse з метою порівняння продуктивності табличних форматів Apache Hudi, Apache Iceberg та Delta Lake для вибору найоптимальнішої архітектури.

Список використаних джерел

1. Analyzing and Comparing Lakehouse Storage Systems / P. Jain, P. Kraft, C. Power, T. Das, I. Stoica, M. Zaharia. *Conference on Innovative Data Systems Research (CIDR).* 2023. P. 1–12. URL: https://mail.vldb.org/cidrdb/papers/2023/p92-jain.pdf (дата звернення: 25.11.2025).

2. Begoli E., Goethert I., Knight K. A Lakehouse Architecture for the Management and Analysis of Heterogeneous Data for Biomedical Research and Mega-biobanks. *IEEE International Conference on Big Data.* 2021. P. 967–975. DOI: 10.1109/BigData 52589.2021.9671534.

3. Comparative analysis of large data processing in Apache Spark using Java, Python and Scala / I. Borodii, I. Fedorovych, H. Osukhivska, D. Velychko, R. Butsii. *Proceedings of the 3rd International Workshop on Computer Information Technologies in Industry 4.0 (CITI 2025)* (м. Тернопіль, Україна, 11–12 June 2025). P. 189–198. URL: https://ceur-ws.org/Vol-4057/paper13.pdf (дата звернення: 25.11.2025).