

НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ БІОРЕСУРСІВ І
ПРИРОДОКОРИСТУВАННЯ УКРАЇНИ
ФАКУЛЬТЕТ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ

УДК 004.9:339:629.23

«ПОГОДЖЕНО»

«ДОПУСКАЄТЬСЯ ДО
ЗАХИСТУ»

Декан факультету
інформаційних технологій
Глазунова О.Г., д.пед.н., професор

Завідувач кафедри комп'ютерних
наук
Голуб Б.Л., к.тех.н., доцент

« » _____ 2021 р

« » _____ 2021 р

МАГІСТЕРСЬКА КВАЛІФІКАЦІЙНА РОБОТА

На тему: Система аналізу ринку продажів автомобілів в Україні

Спеціальність _____ 122 Комп'ютерні науки
(Код і назва)

Освітньо-професійна програма _____ Інформаційні управляючі системи та
технології

Орієнтація освітньої програми _____ (назва)
освітньо-професійна
(освітньо-професійна або освітньо-наукова)

Робота на здобуття кваліфікації магістра

Керівник магістерської роботи
доктор мед.наук, професор
(вчене звання і ступінь)

Глазунова О.Г. /
(ПІБ)

Виконав _____

(підпис)

Танасюк І.О. /

(ПІБ студента)

КИЇВ – 2021

Завідувач кафедри

комп'ютерних наук

(назва кафедри)

к.тех.н., доцент

Голуб Б.Л.

(вчене звання і ступінь)

(підпис)

(ініціали і прізвище)

«29» жовтня 2020 р.

ЗАВДАННЯ

ДО ВИКОНАННЯ МАГІСТЕРСЬКОЇ РОБОТИ СТУДЕНТУ

Танасюку Іллі Олеговичу

(прізвище, ім'я, по-батькові)

Спеціальність 122 Комп'ютерні науки

(шифр і назва)

Освітньо-професійна програма Інформаційні управляючі системи та технології

(назва)

Орієнтація освітньої програми освітньо-професійна

(освітньо-професійна або освітньо-наукова)

Тема магістерської роботи: Система аналізу ринку продажів автомобілів в Україні

Затверджено наказом ректора НУБіП від «29» жовтня 2020 р. № 1634

Термін подання завершеної роботи на кафедру 10 грудня 2020 року

(рік, місяць, число)

Вихідні дані до магістерської роботи: Дані про продажі автомобілів отримані з відкритих джерел, зокрема через API ресурсу auto.ria - <https://api.ria.com/>. Основні критерії: унікальний ідентифікатор оголошення, населений пункт, модель та виробник авто, тип кузова, тип трансмісії, тип палива, дата розміщення оголошення, дата продажу авто, ціна авто в USD, ціна авто в гривні, курс USD/UAH на момент продажу, пробіг авто

4. Перелік питань, що підлягають дослідженню:

1. Дослідити наявні на ринку продукти та інструменти аналізу ринку.
2. Сформувати та розробити pipeline передачі даних, а також систему аналізу отриманих даних.
3. Провести аналіз зібраних даних
4. Дослідити результати аналізу, та виділити метрики які впливають на кількість проданих авто.

5. Перелік графічного матеріалу (за потребами):

постер

Дата видачі завдання

«29» жовтня 2020 р.

Керівник магістерської роботи

(підпис)

Глазунова О.Г.

(прізвище та ініціали)

Завдання прийняв до виконання

(підпис)

Танасюк І.О.

(прізвище та ініціали студента)

НУБІП України

НУБІП України

НУБІП України

НУБІП України

НУБІП України

НУБІП України

НУБІП України

ЗМІСТ

Календарний план	3
Перелік умовних позначень	5
Вступ	6
1. Аналіз та проектування	10
1.1. Постановка завдання	10
1.2. Аналіз предметної області	14
1.3. Проектування системи	16
2. Методи та технології аналізу	18
2.1. Аналіз методів обробки даних	18
2.2. Загальні поняття з напрямку OLAP-технології	19
2.3. Моделювання сховища даних	21
3. Розробка системи аналізу	23
3.1. Опис вузлів системи які поставляють дані до сховища	23
3.2. Механізм вилучення, обробки і передачі даних	27
3.2.1. Опис середовища BI та створення в його середовищі проекту служби SSAS	27
3.2.2. Реалізація отриманих даних за допомогою служби SSIS та інструментів Data Flow	32
3.3. Реалізація процедури аналізу даних в розробленій системі	37
3.3.1. Побудова звітності в середовищі BI	37
3.3.2. Інтелектуальний аналіз даних Data Mining	40
3.3.3. Аналіз даних за допомогою засобів pandas та numpy	43
3.4. Результати дослідження	47
Висновки	48
Список використаних джерел	49
ДОДАТОК А	51
ДОДАТОК Б	55
ДОДАТОК В	59

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ

СД – сховище даних.

БД – база даних.

СУБД (або СКБД) – система управління (керування) базою даних.

ПЗ – програмне забезпечення.

ІС – інформаційна система.

OLAP (англ. - OnLine Analytical Processing) – аналітична обробка в реальному часі.

OLTP (англ. OnLine Transaction Processing) – обробка транзакцій в реальному часі.

SSDT (англ. - SQL Server Data Tools) – набір програмних засобів, які розширюють можливості роботи з даними в середовищі SQL Server та Visual Studio.

SSAS (англ. – SQL Server Analysis Services) – пакет ПЗ який входить до складу SSDT, надає широкі можливості аналізу даних.

SSIS (англ. – SQL Server Integration Services) – пакет ПЗ який входить до складу SSDT, надає можливості міграції даних між різними базами або сховищами даних.

EDA (англ. – Exploratory Data Analysis) – розвідувальний (початковий) аналіз даних.

API (англ. - application programming interface) – прикладний інтерфейс застосунку.

SQL (англ. Structured Query Language) – структурована мова записів використовується для доступу до даних.

Pipeline (англ. - трубопровід) – конвеєр, який являє собою набір засобів, які збирають, обробляють та передають певні дані.

ВСТУП

Автомобільний ринок відноситься до однієї з найбільш розвинутих галузей української економіки. Дослідження проблем розвитку автомобільного ринку України представляє суттєвий науковий та практичний інтерес, враховуючи зростаюче значення автомобільного бізнесу для економіки країни та посилення конкуренції серед представників даного ринку. Проте, незважаючи на таку актуальність автомобільної галузі, засобів для аналізу ринку, зокрема ринку вживаних автомобілів майже не існує. Таким чином, дослідження стану автомобільного ринку України є необхідною умовою для розуміння ситуації, що відбувається на даному ринку в цілому, а також для виявлення основних тенденцій, динаміки та ємності ринку, які в майбутньому можуть допомогти спрогнозувати тенденції на ринку, та принести користь покупцям, які поклавшись на результати дослідження змогли б зробити вибір у відповідності до своїх інтересів та вимог.

Автомобіль забезпечує високу мобільність людини, ефективність праці, визначає сучасний спосіб життя суспільства. Він є показником рівня забезпеченості матеріальними засобами як окремої людини (його власника), так і суспільства або держави в цілому (в даному випадку йдеться про автозабезпеченість населення країни).

Автомобіль належить до товарів тривалого користування, попит на які визначається двома групами факторів – суб'єктивними та об'єктивними. Об'єктивні фактори позначають фактори, які можуть бути обрахованими та прогнозованими за рахунок різних критеріїв. До об'єктивних факторів відносять доходи населення, зростання цін, курс обміну, ставки за кредитами. Суб'єктивні ж фактори є більш абстрактними поняттями, та часто не враховуються у дослідженнях, що призводить до неправильних висновків та результатів досліджень.

Також, враховуючи відносно доволі високу ціну (в порівнянні з іншими витратами, такими як одяг, техніка, побутові товари та інші категорії витрат), автомобіль може розглядатись як засіб інвестиції коштів. Проте як показує

статистика, втрата цінності нового автомобіля за перший рік складає в середньому 25% (дана цифра може збільшуватись, в залежності від таких факторів як репутація моделі, бренду, початкова ціна, ціна обслуговування, популярність авто та інші, і досягати 40%). Даний факт робить ринок вживаних автомобілів доволі вигідним з точки зору збереження фінансів, а в деяких випадках навіть прибутковим.

Слід зазначити, що ситуація на ринку нових автомобілів напряму впливає на ринок вживаних автомобілів оскільки, як показують дослідження, середній термін експлуатації нового автомобіля становить 3-5 років в європейських країнах, та 5-6 років в Україні. Це означає, що автомобілі які були популярними на ринку нових автомобілів 5 років тому, в теперішній момент починають з'являтися на вторинному ринку.

Популярність автомобіля на вторинному ринку є доволі вагомою причиною для покупки, оскільки розповсюдженість моделі напряму впливає на наявність запчастин на ринку, а також обізнаність персоналу для обслуговування даної моделі. Як і було зазначено вище, наразі інструменти для аналізу ринку відсутні, що є доволі ускладнює процес дослідження ринку, динаміку змін ціни, а також отримання ключових параметрів які впливають на формування ціни та кількості продажів.

Метою дослідження являється створення системи аналізу ринку вживаних автомобілів, яка суттєво спростить процес отримання даних щодо динаміки цін, факторів які впливають на кількість продажів, а також дослідження методів та технологій за допомогою яких і буде реалізовано систему аналізу. Реалізовані методи та технології і являтимуть собою **предмет дослідження**.

Об'єкт дослідження: система аналізу ринку вживаних автомобілів, на основі даних отриманих від найбільшого ресурсу з розміщення оголошень про продаж автомобілів auto.ria [1]. Даний ресурс було вибрано тому, що:

- Найбільший ресурс з продажу вживаних та нових автомобілів в Україні
- Наявність API
- Повнота даних та стабільність роботи сервісу

Методи та технології використані в процесі роботи: емпіричні та теоретичні наукові методи, OLAP технології, методи EDA, аналіз часових рядів, технології SSAS Data Mining.

Завданням даної роботи є: збір даних та розробка системи аналізу ринку вживаних автомобілів, яка дозволить виділити метрики від яких залежить кількість продажів вживаних автомобілів на ринку України.

Апробація постановки проблеми та аналізу рішень дослідження відбулася в рамках IV Всеукраїнської інтернет-конференції студентів і аспірантів “Теоретичні та прикладні аспекти розробки комп’ютерних систем 2021”, яка відбулася 29 квітня 2021 року, м. Київ, Україна.

Результати дослідження та доцільність впровадження системи було апробовано в рамках XII Міжнародної науково-практичної конференції молодих вчених «Інформаційні технології: економіка, техніка, освіта» 11 листопада 2021 року м. Київ, Україна та розроблено постер.

Пояснювальна записка магістерської роботи включає в себе: вступ, три розділи, висновки, літературні джерела та додатки. У вступі описуються ключові аспекти, які характеризують предметну область дослідження. Вказані мета дослідження, об’єкт, предмет та завдання.

В першому розділі проводиться аналіз постановки завдання та предметної області, більш широко розкривається постановка завдання, наводиться діаграма прецедентів та архітектура системи.

В другому розділі розглядаються методи та технології аналізу які було використано для проведення дослідження, а також обґрунтовується вибір даних технологій, розкривається зміст та сутність системи, її інформаційне забезпечення та технічна складова: вузли системи.

Третій розділ присвячено розробці системи аналізу: механізми отримання, трансформації, обробки і передачі даних, створення проекту за допомогою засобів Visual Studio, зокрема SSAS, а також побудова середовища аналізу за допомогою мови python, зокрема бібліотек pandas, numpy та scipy.

У висновках наведено оцінку отриманих результатів роботи та визначено межі використання результатів роботи.

До пояснювальної записки також додається графічний матеріал – постер,
на якому відображені основні аспекти роботи

НУБІП України

НУБІП України

НУБІП України

НУБІП України

НУБІП України

НУБІП України

НУБІП України

1. АНАЛІЗ ТА ПРОЕКТУВАННЯ

Мета процесу аналізу (предметної області) полягає у проведенні дослідження інформаційного поля, яке буде використане як основа для розробленої системи, а також в якому вона функціонуватиме. В процесу аналізу та проектування сформовано чіткі інструкції та кроки для виконання роботи.

Вирішені завдання:

- Конкретизація завдань та формування вимог.
- Розробка архітектури системи.
- Вибір технологій та механізмів реалізації системи.
- Розробка структури компонентів.

1.1. Постановка завдання

Етап постановки завдання є доволі важливим, оскільки саме на даному етапі визначаються цілі яких необхідно досягти в результаті виконання дослідження, а також яким способом ці цілі будуть досягнуті. На цьому етапі формуються точні умови, з описом вхідних та вихідних даних.

Вхідна інформація – інформація, яка подається на вхід системи, і на основі якої буде функціонувати система. Вона може бути представлена у вигляді різних джерел – документів, фото або відео інформація з камер, потік даних з бази даних, сигнали отримані з датчиків, та інші види вхідної інформації.

Оскільки дана система надає функціональність яка дозволяє аналізувати ситуацію на ринку вживаних автомобілів України, то вхідні дані для роботи системи є реальними, тобто повинні бути взяті з достовірних джерел і відповідати дійсності. В протилежному випадку цінність даних буде нульовою.

В якості основного і єдиного джерела даних виступає інтернет ресурс auto.rta.com, який являє собою найбільший майданчик з розміщення оголошень про продаж нових та вживаних автомобілів (а також не тільки автомобілів, на

сайті представлені інструменти для продажу мототехніки, а також човнів, літаків, спецтехніки). Даний веб ресурс входить до складу естонської технологічної компанії [ria.com marketplaces](http://ria.com) [2], яка володіє правами, і розвиває та підтримує наступні ресурси:

- auto.ria.com
- dom.ria.com
- ria.com
- ria.by
- domria.eu

Даний ресурс було вибрано зважаючи на декілька факторів, а саме:

1. **Популярність.** Даний веб сайт є найпопулярнішим в даному секторі в порівнянні з конкурентами (такими як rst.ua, olx.ua).

2. **Зручність роботи з даними.** Після проведеного дослідження, було

виявлено, що жоден з вищенаведених сайтів, окрім вибраного, не надає відкритого та безплатного API, що суттєво ускладнювало б процес отримання даних, їхньої обробки, а також забезпечення узгодженості даних (англ. consistency) – оскільки не всі продавці автомобілів

вказують VIN код (англ. Vehicle Identification Number – ідентифікаційний номер транспортного засобу, присвоюється виробником, та вказується фізично на кузові автомобіля), що суттєво ускладнює процес виявлення та видалення дубікатів оголошень отриманих з різних джерел.

3. **Кількість метрик.** Вибрана платформа широкий перелік параметрів автомобілів, які можуть бути використані для аналізу, в той час як конкуренти містять лише загальні

В якості робочих даних було обрано наступний перелік параметрів оголошення про продаж автомобіля: назва виробника, назва моделі автомобіля, рік випуску, пробіг (в км.), тип кузова, тип трансмісії, тип пального, місто подачі оголошення, ціна продажі в USD, ціна продажі в UAH курс валют UAH/USD на момент продажу.

Для взаємодії з API ресурсу auto.ria було розроблено скрипт мовою python, який за допомогою бібліотеки requests формував запити до сервера, та отриманий результат в форматі json парсив (англ. parse – розбір, розбиття даних, найчастіше текстових на складові, та отримання з них необхідних фрагментів) їх, та формував файли з інструкціями insert мовою SQL, які в подальшому вносяться до БД, зберігання оперативної інформації в якій є однією з вимог до системи.

Даний підхід було вибрано в угоду надійності, оскільки програмний засіб працював з зовнішнім веб сервісом, який може бути не завжди доступним через низку причин: вихід з ладу серверного обладнання (на якому запущено скрипт), неполадки з мережевим з'єднанням, внутрішні проблеми ресурсу, тощо. В разі непередбачуваної внутрішньої помилки самого скрипту та аварійного завершення його роботи, дані не втрачаються, а залишаються записані в файли на диску. Також, в разі виходу з ладу СУБД, в яку вносяться дані, маючи на диску файли з insert інструкціями, БД можна легко відновити.

База даних (БД) – організована структура, призначена для зберігання та редагування (на відміну від СД) великої кількості взаємопов'язаних структурованих даних.

У якості СУБД для даного проекту було вибрано PostgreSQL (іноді скорочується як Postgres або psql, за аналогією з консольним терміналом для даного додатку), який є одним з найбільш розповсюджених продуктів для адміністрування БД. Дана СКБД зарекомендувала себе як дуже високоефективна, швидка система з широкими можливостями для гнучкого налаштування, а також розширення функціоналу за допомогою розширень, які розробляються як спільнотою користувачів Postges, так і командою яка розробляє безпосередньо сам продукт.

На основі БД, для цілей аналізу, було створено також СД – сховище даних.

СД є предметно-орієнтованою базою даних, яка узагальнює деякі дані з БД, найчастіше за часовим виміром, та використовується для подальшого аналізу. Узагальнення даних необхідне для зменшення навантаження на обчислювальні потужності, оскільки в аналітичних даних необхідна загальна інформація про деякі процеси, а не уточнене представлення – наприклад сума продаж певного

товару за проміжок часу, а не перелік купівель за аналогічний період. На відміну від БД, в СД нормалізація відіграє децю слабшу роль, оскільки швидкодія виконання SQL запитів в ньому не відіграє ключової ролі.

Не зважаючи на той факт, що БД і СД за визначенням є досить компонентами системи, це не так. БД використовується для оперативної обробки даних, і працює за моделлю OLTP, в той час як СД створюється для аналізу даних за допомогою OLAP технологій. Порівняння даних технологій наведено в таблиці 1.

Таблиця 1. Порівняльна характеристика OLAP та OLTP

Показник	OLTP	OLAP
Частота та обсяг оновлюваних даних	Велика частота оновлення відносно малий обсяг оновлюваних даних	Низька частота оновлення і великий обсяг оновлюваних даних
Горизонт зберігання даних	Один календарний рік (як правило, за умови використання СД)	Необмежений
Ступінь агрегування даних	Первинні облікові дані	Проміжні підсумки. Зведені підсумки. консолідовані дані
Характер обробки даних	Строго регламентовані запити і звіти	Довільні запити, вибірки, зведення
Цілі створення	Облік первинних даних. Оптимізація зберігання даних. Оптимізація експлуатаційних характеристик додатків	Аналіз та моделювання даних. Формування прогнозів. Формалізація знань про систему управління

1.2. Аналіз предметної області

Аналіз предметної області проводиться перед початком проектування бази даних, для формування розуміння структури даних, їх взаємозв'язків, та способів ефективного їх зберігання та подальшої обробки.

Також, процес аналізу предметної області дозволяє виділити основні та допоміжні бізнес процеси, які забезпечують реалізацію функціональності системи аналізу. Окрім того, визначення бізнес процесів дозволяє визначитись з бізнес елементами та структурою даних яка буде безпосередньо брати участь в розробці.

Для більш точного вирішення бізнес завдань, розробнику системи необхідно глибоко занурюватись у дослідження бізнес процесів, що в подальшому дозволить краще зрозуміти завдання які необхідно виконати. Для наглядного представлення досліджуваної предметної області використовуються графічні схеми.

Стандартом для створення таких схем є UML – unified modeling language. Вона регламентує правила створення, а також види можливих діаграм та їхнє призначення. Однією з найбільш загальних діаграм які демонструють основні бізнес процеси та акторів які взаємодіють з ними, а також описує систему на концептуальному рівні є use-case діаграма, або ж діаграма прецедентів.

При створенні діаграми прецедентів було дотримано наступних правил:

- Кожен прецедент відноситься як мінімум до одного актора.
- Кожен прецедент має ініціатора.
- Кожен прецедент має результат виконання.

Діаграма прецедентів даної системи наведена на рис. 1.

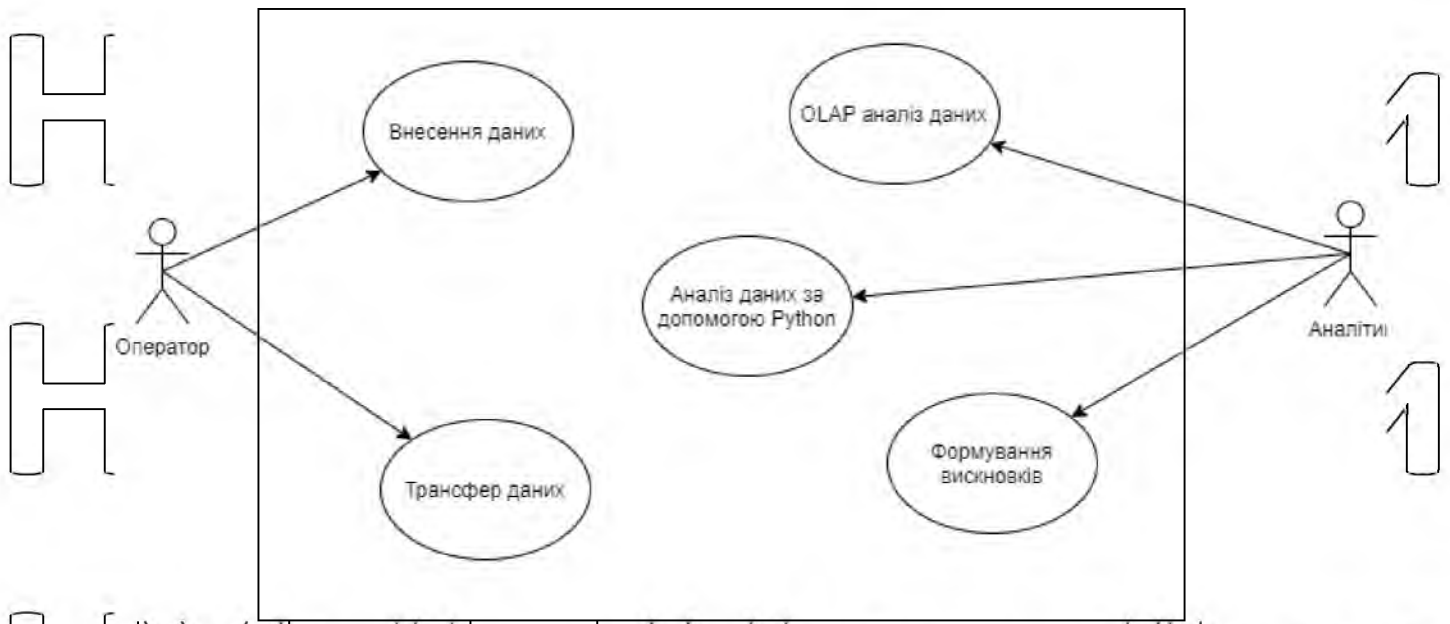


Рис. 1 – діаграма прецедентів системи

З діаграми можна зрозуміти, що до системи мають доступ 2 актори:

- Оператор.
- Аналітик.

Завданням оператора є контролювати виконання скрипта який отримує дані з зовнішнього джерела, та виконувати SQL скрипти на внесення отриманих даних до БД. Наступним завданням оператора є ініціалізація процесу data-flow (англ. потік даних) від БД до СД, та забезпечення безперебійної та безпомилкової роботи системи. Аналітик займається тим, що ініціалізує процеси аналізу ринку за допомогою інструментів SSAS та python, а також порівнює отримані результати з двох модулів аналізу, та формує результати дослідження.

Кожен з прецедентів наведених на рис.1 має свої варіанти використання, які описані в таблиці 2.

Таблиця 2. Опис прецедентів

Актор	Назва прецеденту	Опис прецеденту
Оператор	Внесення даних	Контроль виконання скрипту отримання даних, та виконання

Оператор	Трансфер даних	SQL скриптів для внесення даних до БД Ініціалізація та контроль процесу перенесення даних з БД до СД
Аналітик	OLAP аналіз даних	Ініціалізація процесу аналізу за допомогою OLAP технологій, зокрема SSAS
Аналітик	Аналіз даних за допомогою Python	Ініціалізація процесу аналізу даних за допомогою модуля розробленого мовою Python
Аналітик	Формування висновків	Порівняння результатів аналізу обох модулів, та формування висновків результатів аналізу

1.3. Проектування системи

Проектування системи аналізу ринку вживаних автомобілів було розпочато з визначення з мети, і цілей системи. Завдання системи полягає в тому, щоб після її запуску можна було забезпечити:

- Необхідну функціональність, і достатній ступінь автоматизації щоб забезпечити її функціонування з напливом нових даних
- Необхідний ступінь захищеності від помилок та стійкість до відмов
- Необхідний час реагування на запити аналітиків
- Простоту експлуатації системи
- Наглядність та зрозумілість результатів роботи системи

При проектуванні системи було охоплено три основні області:

- Проектування об'єктів, які будуть реалізовані в БД
- Проектування програмних засобів мовою Python, а також проект SSAS

- Проектування архітектури та топології системи

Наступним кроком розробки, після аналізу предметної області, став процес розробки архітектури системи. Архітектура системи являє собою схему на якій вказана послідовність потоку даних, а також кількість та тип модулів які входять

до складу системи. Іншими словами, архітектура системи – це сукупність її

основних модулів та засобів забезпечення їх взаємодії один з одним, з користувачем та з системним персоналом. Архітектуру системи наведено на рис.

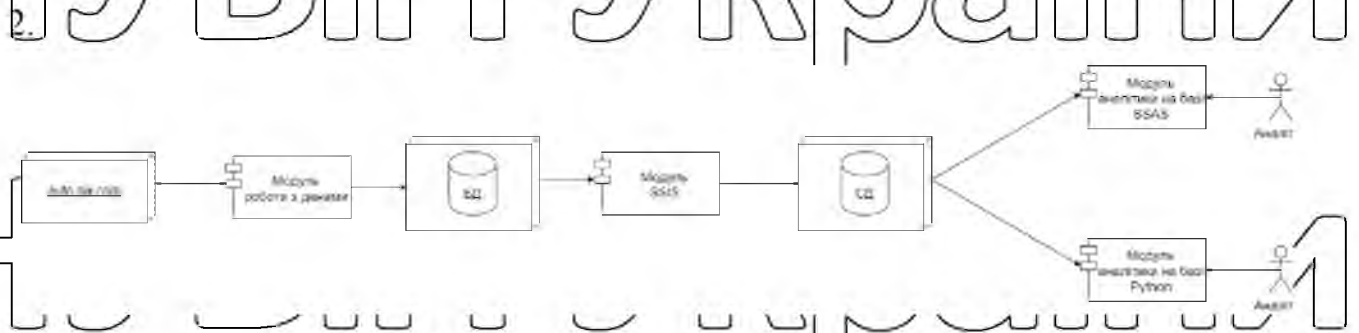


Рис. 2 – архітектура системи

Інформаційна система аналізу ринку вживаних автомобілів, з позиції користувачів, складається з аналітика, який працює з модулями аналізу, розробленими за допомогою SSAS та мови програмування Python. Також в

роботу системи вносить вклад оператор, який вносить дані, отримані скриптом з

зовнішніх джерел, до БД, а також ініціалізує процес трансферу даних з БД до СД

засобами SSIS, що дозволяє транспортувати до СД лише необхідні дані відповідно до архітектури реалізованої в СД.

2. МЕТОДИ ТА ТЕХНОЛОГІЇ АНАЛІЗУ

2.1. Аналіз методів обробки даних

Для проведення аналізу даних, вхідна інформація повинна бути структурованою, та підготовленою спеціальним чином, оскільки різні типи інформації обробляються по різному.

В науці про дані, існує два фундаментальних типи структурованих даних: числовий і категоріальний. Числові дані надходять у двох формах: безперервній, як наприклад тривалість часу, і дискретній, як-от кількість проданих автомобілів за одиницю часу. Категоріальні дані приймають лише фіксований набір значень, наприклад, тип кузова автомобіля (купе, хетчбек, родстер та інші) або тип приводу (передній, задній, повний). Двійкові дані є важливим особливим випадком категоріальних даних. Ці дані приймають лише одне з двох значень, таких як 0/1, так/ні або істина/брехня. Ще один корисний тип категоріальних даних представлений порядковими даними, у яких категорії упорядковані; їх прикладом є числова характеристика. [3]

Відомості про тип даних можуть суттєво допомогти у виборі підходящого методу обробки даних, а також налаштування БД відповідним чином, що дозволить створити індекси наприклад, або зберігати дані у відсортованому вигляді.

Після визначення типів наявних даних, рекомендується застосувати такий вид аналізу як розвідувальний аналіз, або ж EDA (англ. (exploratory data analysis)). Розвідувальний аналіз це підхід до аналізу наборів даних для узагальнення їх основних характеристик, часто з використанням статистичної графіки та інших методів візуалізації даних. EDA призначена для того, щоб побачити, що дані можуть розповісти нам за межами формального моделювання чи завдання перевірки гіпотези. Розвідувальний аналіз даних пропагується Джейном Тьюкі з 1970 року, щоб заохотити статистиків досліджувати дані та, можливо,

сформулювати гіпотези, які могли б привести до збору нових даних та експериментів.

Як слідує з вищенаведеного висновку, розвідувальний аналіз даних є доволі корисним і наглядним інструментом для початкового аналізу даних, який дозволить наглядно оцінити ситуацію на ринку, та визначити параметри які впливають на кількість продажів автомобілів. Для проведення EDA використано засоби мови програмування Python, зокрема програмні пакети pandas, numpy, scipy для аналізу та прогнозування, а також пакет matplotlib для побудови графіків та візуалізацій даних.

Об'єм даних який обробляється системою є відносно невеликим, оскільки вибрана платформа не надає архівних даних про продані автомобілі, і тому система оперує лише даними зібраними підчас проведення дослідження (листопад 2021 – листопад 2021), і складає близько 180000 записів про продані автомобілі.

2.2. Загальні поняття з напрямку OLAP-технології

OLAP-системи (On-Line Analytical Processing) представляють собою інструментарій для аналізу великих обсягів даних в режимі реального часу на основі гнучкого перегляду, отримання довільних зрізів даних і виконання аналітичних операцій деталізації, згортки, порівняння в часі і ін. Якщо системи регламентованої звітності дозволяють відповісти на питання "Який дохід підприємства за останній квартал?", то OLAP-системи дають відповідь на питання: "Якого постачальника вигідно вибрати і чому?".

Вимоги до OLAP-систем сформульовані в тесті FASMI (Fast Analysis of Shared Multidimensional Information - швидкий аналіз розділяється багатовимірної інформації):

- Fast (швидкий) - надання результатів аналізу за прийнятний час (зазвичай не більше п'яти секунд), нехай навіть ціною менш детального аналізу.

• Analysis (аналіз) - можливість здійснення будь-якого логічного і статистичного аналізу, характерного для цього додатка, і його збереження в доступному для кінцевого користувача вигляді.

- Shared (розділяється) - розрахований на багато користувачів доступ до даних з підтримкою відповідних механізмів блокувань і засобів авторизованого доступу.

• Multidimensional (багатовимірний) - багатовимірне концептуальне представлення даних, включаючи повну підтримку для ієрархій та множинних ієрархій.

• Information (інформація) - можливість звертатися до будь-якої потрібної інформації незалежно від її обсягу і місця зберігання.

Всі OLAP-системи діляться на три класи за типом вихідної БД

1. MOLAP (Multidimensional OLAP) - багатовимірний OLAP

(використовуються багатовимірні БД). Застосування доцільно за умови не дуже високих обсягів вихідних даних для аналізу, стабільному наборі інформаційних вимірювань, а також в тих випадках, коли час відповідей системи на нерегламентовані запити є найбільш критичним параметром.

2. ROLAP (Relational OLAP) - реляційний OLAP (використовуються

реляційні БД). Набули поширення дві основні схеми реалізації: схема "зірка" і схема "сніжинка". Переваги: в більшості корпоративних СД використовуються реляційні СУБД і інструменти ROLAP працюють безпосередньо над ними;

наявність великих можливостей захисту даних і розмежування прав доступу користувачів. Недолік - менша в порівнянні з MOLAP продуктивність.

3. HOLAP (Hybrid OLAP) - гібридний OLAP. Використовуються як багатовимірні, так і реляційні бази даних: найбільш затребувані агреговані бізнес-показники зберігаються в багатовимірному просторі, а ресурсоемні

детальні дані - в реляційному. Як недоліки можна виділити: складність синхронізації регламентів завантаження, модифікації алгоритмів узгодження і верифікації моделей консолідації, розподілу прав доступу до інформаційних ресурсів системи.

Перевагами OLAP-систем є:

- простота використання і сприйняття зведених таблиць.
- повнота аналітичних даних.
- повна і легка настройка звіту користувачем без залучення програміста.

- можливість деталізувати звіт в процесі аналізу даних.
- швидке формування звітів.
- несуперечність даних між звітами.
- консолідація інформації з різних БД.
- підвищений захист даних.

2.3. Моделювання сховища даних

Сховище даних або ж СД (англ. data warehouse, DWH) – предметно орієнтований набір даних, що підтримує хронологію і здатний бути джерелом достовірної узагальненої інформації для проведення аналізу даних та підтримки рішень в відповідних системах підтримки прийняття рішень (СППР). В переважній більшості СД створюється для використання в поєднанні з OLAP технологією, на відміну від БД, яка використовується з OLTP методами. Частою архітектурою є послідовність з БД, після якої є модуль трансферу даних, а потім СД. Тобто OLAP СД наповнюється даними з OLTP БД.

Дані з БД переміщуються до СД таким чином. Що при побудові звітів та аналізі не використовується транзакційна система, а також по мінімуму використовуються засоби ядра БД, які формують результуючу вибірку з декількох джерел.

Існує два головних напрямки в архітектурі сховищ даних: нормалізовані сховища, та сховища з вимірами. При використанні архітектури з вимірами, найчастішою схемою побудови сховища даних є зірка, в якій існує декілька таблиць вимірів, в центрі зірки розміщується таблиця фактів, яка пов'язує всі таблиці фактів. Саме через це СД допускає слабу нормалізацію, тобто дані в таблицях вимірів можуть бути надмірними. Перевагами даного підходу, як уже

було сказано є зменшене навантаження на ядро СУБД, і покращений час виконання запиту. Сховища побудовані по принципу нормалізації по своїй архітектурі майже не відрізняються від БД, і відповідно успадковують всі переваги та недоліки що належать й класичній нормалізований базі даних.

Є два сценарії оновлення даних в сховищі:

1. Повне оновлення. Під час кожної синхронізації всі дані з сховища видаляються, та заповнюються новими даними
2. Інкрементальне оновлення. Оновлюються тільки дані, які були змінені в OLTP джерелі.

Після проведення попереднього аналізу було прийнято рішення побудови СД в вимірах, за схемою зірка.

НУБІП України

НУБІП України

НУБІП України

НУБІП України

3. РОЗРОБКА СИСТЕМИ АНАЛІЗУ

НУБІП України

3.1. Опис вузлів системи які поставляють дані до сховища

Після отримання даних з зовнішнього джерела, дані вносяться до БД, логічна схема даних якої наведена на рис. 3.

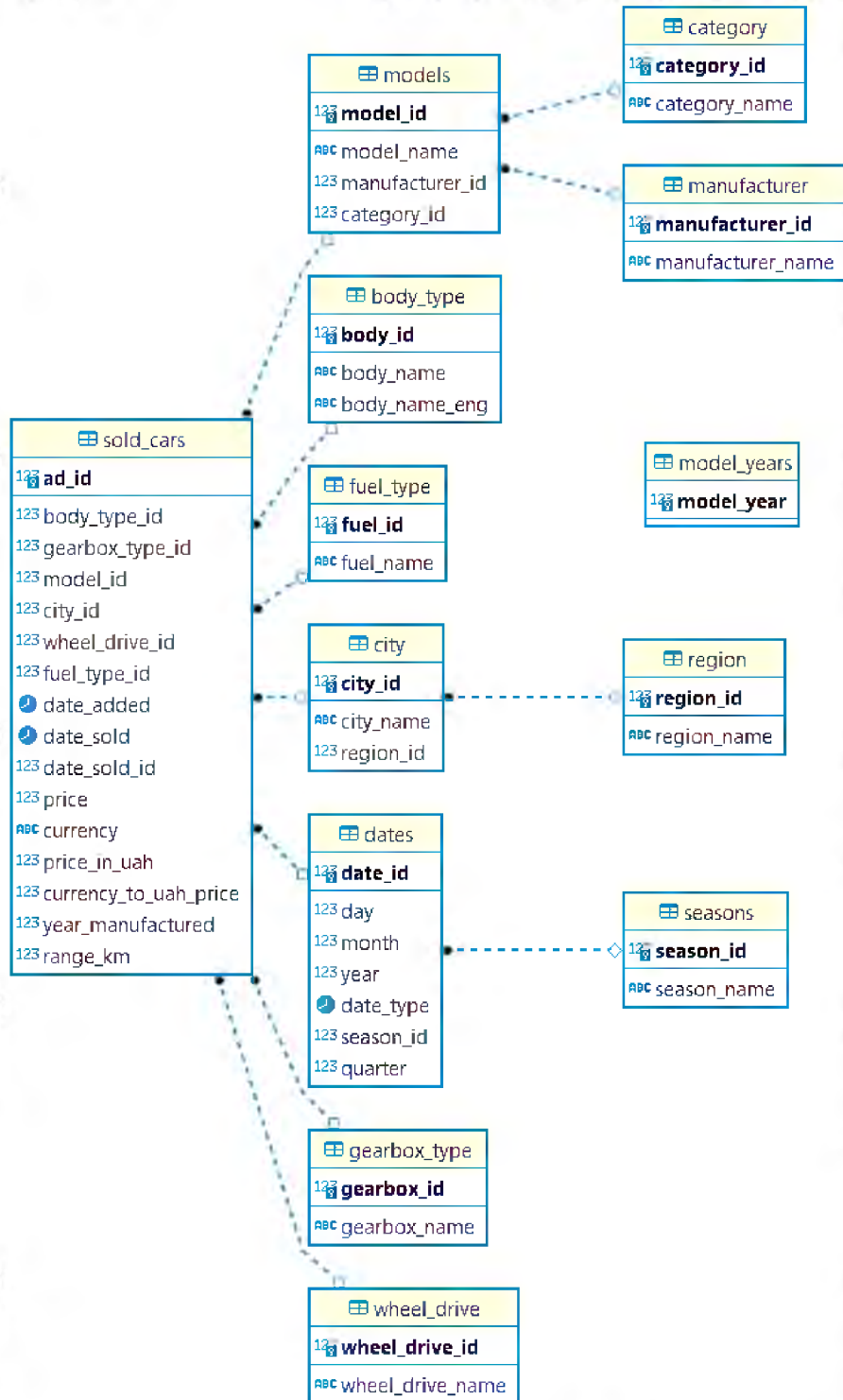


Рис. 3. – логічна схема даних БД

Дана база даних зберігає дані про продажів автомобілів на здійснені через ресурс auto.ria, і оскільки головною сутністю даної БД є продаж автомобіля, відповідно таблиця яка містить дані про продаж автомобілів є найбільшою.

В базі даних було визначено наступні таблиці:

- **Category** – зберігає дані про категорію автомобіля, а саме: легкові, вантажні, автобуси і тд. Поля: код категорії, та назва категорії.
- **Manufacturer** – зберігає дані про виробників автомобілів, які в подальшому будуть використані в таблиці яка містить дані про моделі автомобілів. Поля: код автовиробника, та його назва.
- **Models** – зберігає дані про моделі автомобілів, комбінуючи в собі дані з двох вищеописаних таблиць, за допомогою зовнішнього ключа (англ. foreign key). Поля: код моделі, назва моделі, код виробника, код категорії.
- **Body_type** – зберігає дані про тип кузова автомобіля, наприклад хетчбек, купе, седан, і тд. Поля: код типу кузова, назва типу кузова.
- **Fuel_type** – зберігає дані про тип пального, який використовує автомобіль, наприклад бензин, дизель, гібрид і тд. Поля: код типу пального, назва типу пального.
- **Region** – зберігає дані про область, які в подальшому будуть використані в таблиці яка зберігає дані про міста. Поля: код області, назва області.
- **City** – зберігає дані про місто, в якому було здійснено продажу автомобіля. Поля: код міста, назва міста, код області.
- **Seasons** – зберігає дані про пори року, наприклад літо, зима, весна або осінь. Поля: код пори року, назва пори року.
- **Dates** – зберігає дані про дати здійснення продажу. Таку інформацію було винесено в окрему сутність, задля можливості уточнення даних (окремо зберігається інформація щодо днів, місяців, років, кварталів

та сезонів) Поля: код дня, номер дня, номер місяця, номер року, повна дата в форматі Date, код сезону, номер кварталу.

- **Gearbox_type** – зберігає дані про тип коробки перемикачів, яка встановлена на автомобілі, наприклад механічна, автоматична, варіатор, і тд. Поля: код типу коробки перемикачів, назва коробки перемикачів.

- **Wheel_drive** – зберігає дані про привід автомобіля, наприклад передній, задній, повний. Поля: код типу приводу, назва типу приводу автомобіля.

- **Sold_cars** – зберігає дані про продані автомобілі на ринку України, і агрегує дані з усіх вищенаведених таблиць. Поля: код оголошення на сайті auto.gia, код кузова автомобіля, код коробки перемикачів, код моделі, код міста в якому було здійснено продажу, код типу приводу, код типу палива, дата створення оголошення, дата продажу авто, код дати продажу, ціна авто в іноземній валюті, ціна авто в національній валюті, курс USD/UAH, рік виготовлення автомобіля, пробіг автомобіля в кілометрах.

- **Model_years** - зберігає дані про всі модельні роки автомобілів, облік яких ведеться. Створена для подальшого використання в СД, оскільки в створеному СД інформація про моделі буде також містити рік випуску для кожної моделі, таким чином буде досягнута унікальна комбінація коду моделі та року випуску.

Для подальшого аналізу було створено СД, схема даних якого наведена на рис. 4.

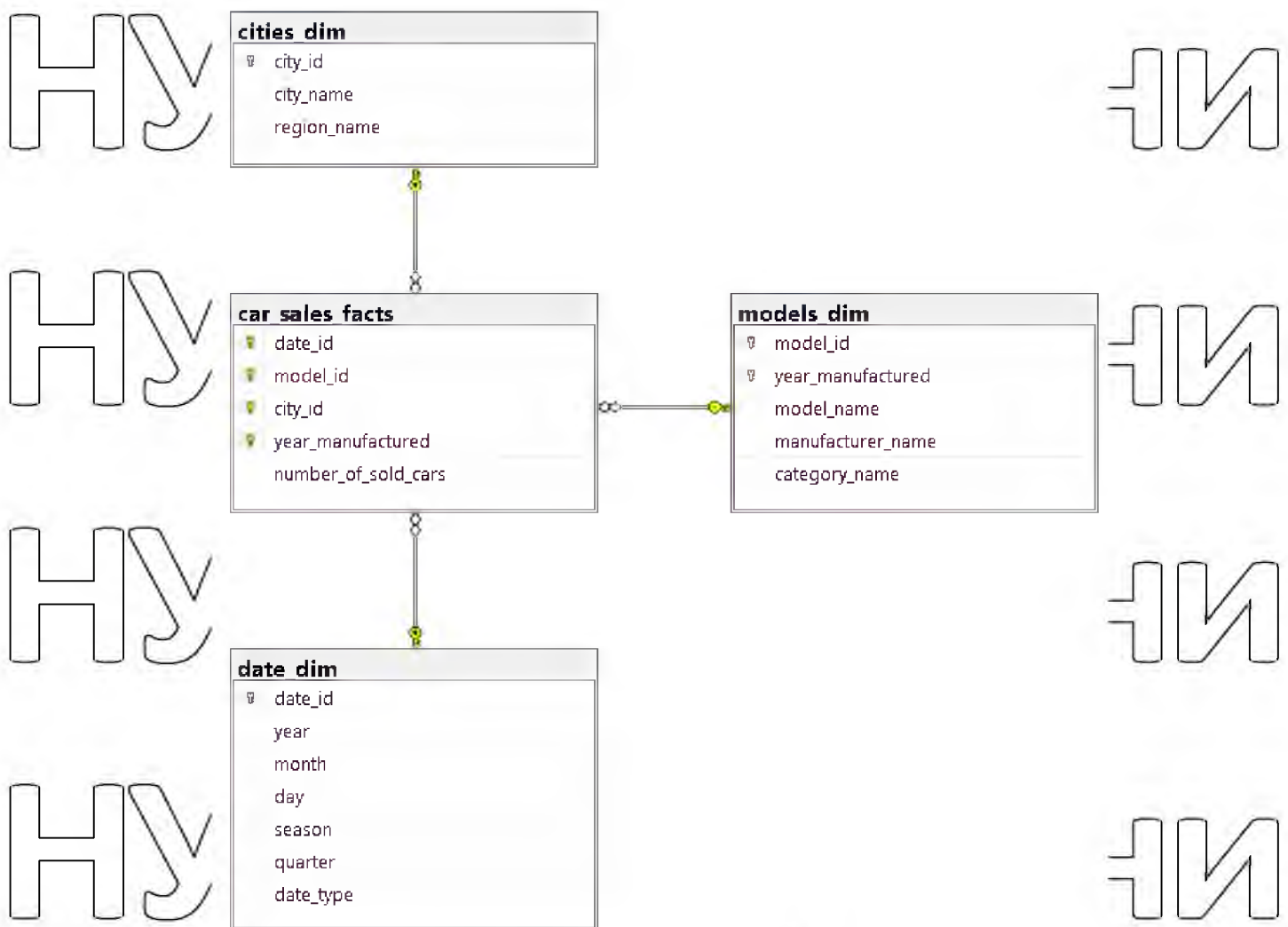


Рис. 4. – логічна схема СД

Як було описано раніше, СД було створено для процесу аналізу, оскільки використання СД є доцільнішим в зв'язку з певним агрегуванням даних.

Створене сховище даних має наступні таблиці:

- **Date_dim** – таблиця яка зберігає часовий вимір (англ. dimension), і агрегує в собі всі часові дані, які були в БД. Поля: код дати, рік, місяць, день, пара року, квартал, повна дата в форматі Date.
- **Models_dim** – таблиця яка зберігає вимір моделей, і агрегує всі дані пов'язані з моделями, виробниками, а також роками випуску автомобілів. Поля: код моделі, рік випуску авто, назва моделі, назва виробника, назва категорії. Код моделі та рік випуску формують складений ключ.

- **Cities_dim** – таблиця яка містить вимір міст та областей. Поля: код міста, назва міста, назва області.

- **Car sales facts** – таблиця яка агрегує в собі дані необхідні для аналізу, та містить посилання на таблиці вимірів – таблиця фактів.

Поля: код дати, код міста, код моделі, кількість проданих авто.

Як можна побачити з логічної схеми, та опису таблиць СД, дані в них не нормалізовані, і в порівнянні з БД є агрегованими. На відміну від БД, в даних вимірах коди тих чи інших параметрів замінені на безпосередні назви необхідних сутностей, що є не відповідністю нормальним формам, зате знімає навантаження з ядра БД при виконанні запитів, оскільки відсутні оператори join

3.2. Механізм вилучення, обробки і передачі даних

3.2.1. Опис середовища BI та створення в його середовищі проекту

служби SSAS

Analysis Services – це засіб аналітичних даних (VertiPaq), який використовується у службі підтримки прийняття рішень та бізнес-аналітики. Він надає можливості моделі семантичних даних корпоративного рівня для бізнес-аналітики (BI), аналізу даних та створення звітів, таких як Power BI, Excel, Reporting Services та інші засоби візуалізації даних. Analysis Services доступний на різних платформах:

- Azure Analysis Services, створені як ресурс Azure, Azure Analysis Services серверні ресурси підтримують табличні моделі на рівнях сумісності 1200 і вище. Крім того, також підтримуються DirectQuery, секції, безпека на рівні рядків, двонаправлені зв'язки та переклади. Щоб отримати додаткові відомості, див. Що таке Azure Analysis Services.

- Power BI Premium. підсистема VertiPaq Analysis Services забезпечує програмованість, клієнтську програму та підтримку засобів для Power BI Premium та Premium для наборів даних користувачів на рівнях сумісності 1500 і вище за допомогою клієнтських бібліотек та арі-інтерфейсів, що підтримують відкритий стандартний протокол

XMLA. Power BI Premium набори даних підтримують підключення через кінцеві точки XMLA для операцій лише для читання та для читання та запису від сторонніх клієнтських програм та засобів майкрософт. Додаткові відомості див. у розділі Analysis Services у Power BI Premium та Power BI Premium підключення набору даних до кінцевої точки XMLA.

- SQL Server Analysis Services, встановлений як локальний екземпляр сервера або екземпляра віртуальної машини, SQL Server Analysis Services підтримує табличні моделі на всіх рівнях сумісності (залежно від версії), багатовимірні моделі, інтелектуальний аналіз даних та Power Pivot для SharePoint.

Системи типу BI обов'язково містять такі наступні три основні категорії функцій:

- можливість інтеграції;
- представлення інформації;
- аналіз даних.

При розробці системи аналізу було використано середовище SQL Server Business Intelligence, в якому формувався куб, створювались потоки даних для отримання даних з джерел на заповнення створеного кубу

Для створення проекту для роботи з SSAS в середовищі Visual Studio необхідно додати даний шаблон в extension manager (рис. 5).

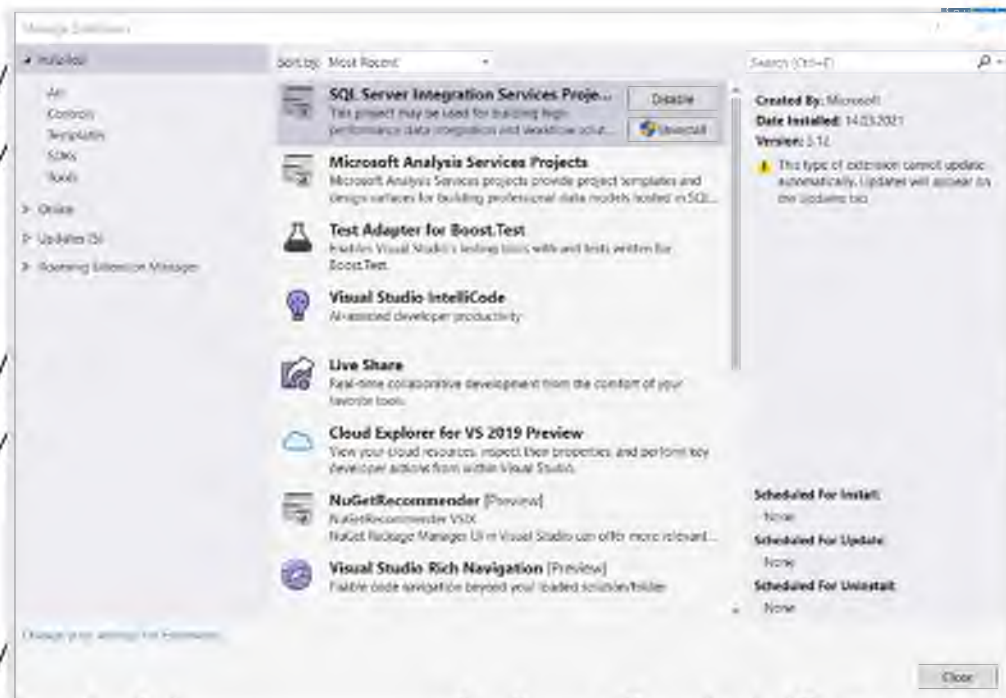


Рис. 5 – вигляд extension manager з встановленим шаблоном проекту SSAS

Створений проект має наступну структуру:

- Data sources
- Data source views
- Cubes
- Dimensions
- Mining structures
- Roles
- Assemblies
- Miscellaneous

Вигляд структури проекту наведено на рис. 6.

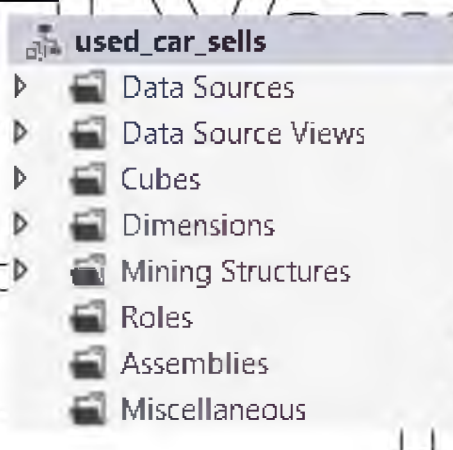


Рис. 6 – вигляд структури проекту SSAS

Для створення кубу необхідно ввести обличчя дані для підключення до серверу на якому зберігається СД, а також згенерувати data source view, який дозволяє вибрати лише необхідні виміри та таблиці фактів, а також уявлення до проекту. Таблиці включені до data view source будуть використані для наповнення кубу. Створене data source view наведено на рис. 7

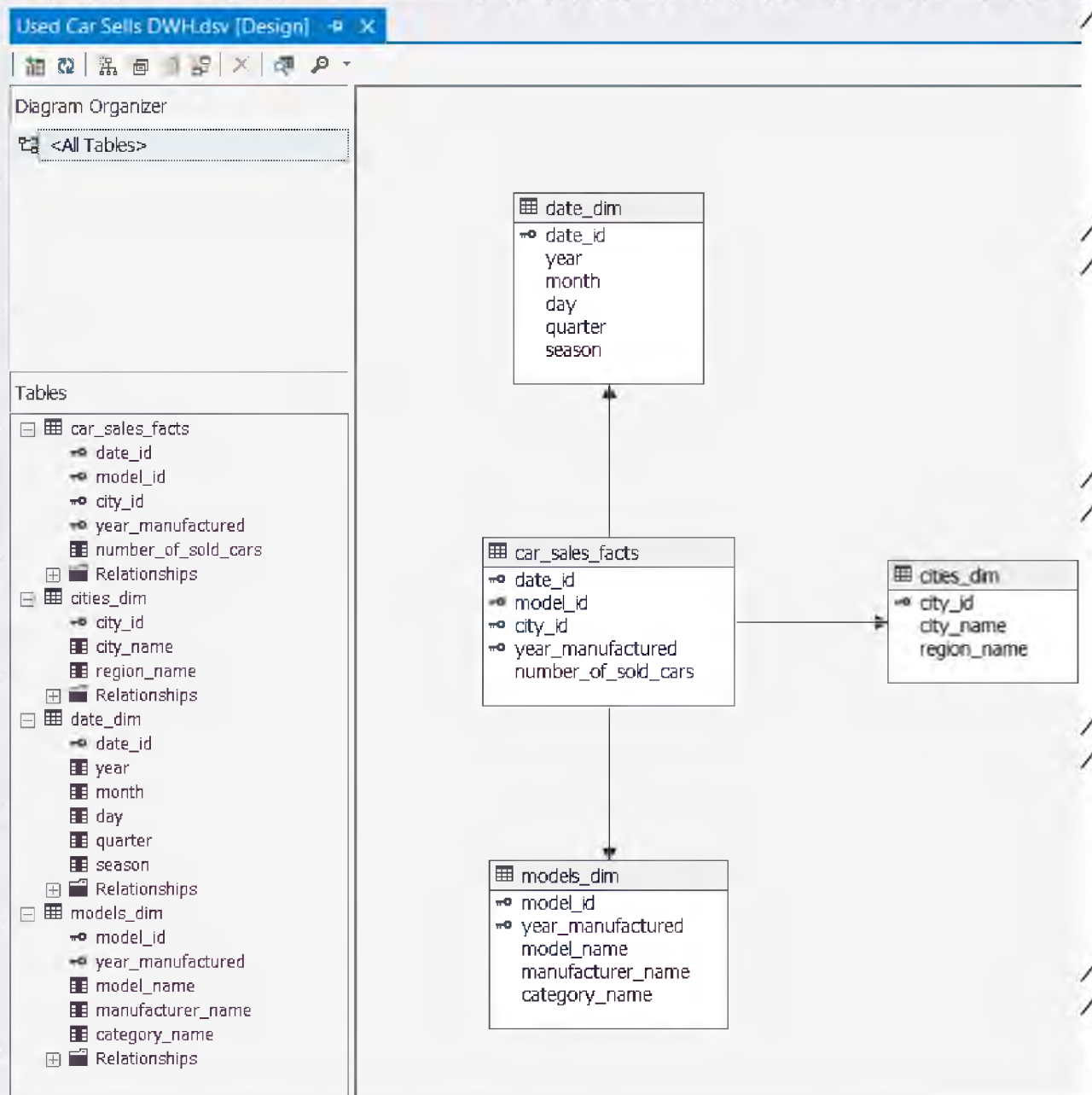


Рис.7 – створене data source view

Наступним кроком є створення та розгоргання кубу

Куб являє собою багатовимірну систему зберігання даних. Осями багатовимірної системи координат є основні атрибути аналізованого бізнес-процесу. Наприклад, для продажу це може бути товар, регіон, тип покупця. Як один із вимірів використовується час. На перетинах осей – вимірювань (Dimensions) – знаходяться дані, що кількісно характеризують процес – заходи (Measures). Це можуть бути обсяги продажу в штуках або в грошовому вираженні, залишки на складі, витрати і т.п. по тижнях) відомості та здійснювати інші маніпуляції, які йому спадають на думку в процесі аналізу.

Результати розгортання кубу та вигляд розгорнутого куба наведено на рис.8 та рис.9 відповідно.

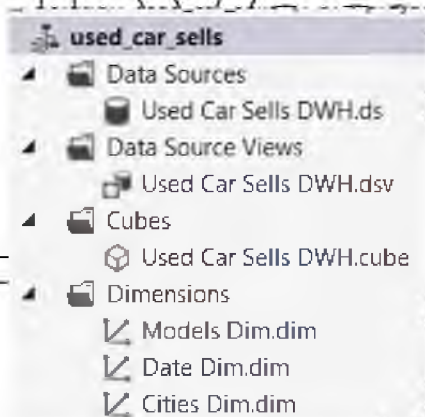


Рис. 8 – результати розгортання кубу – в середовищі Visual Studio

сформувались виміри (dimensions)

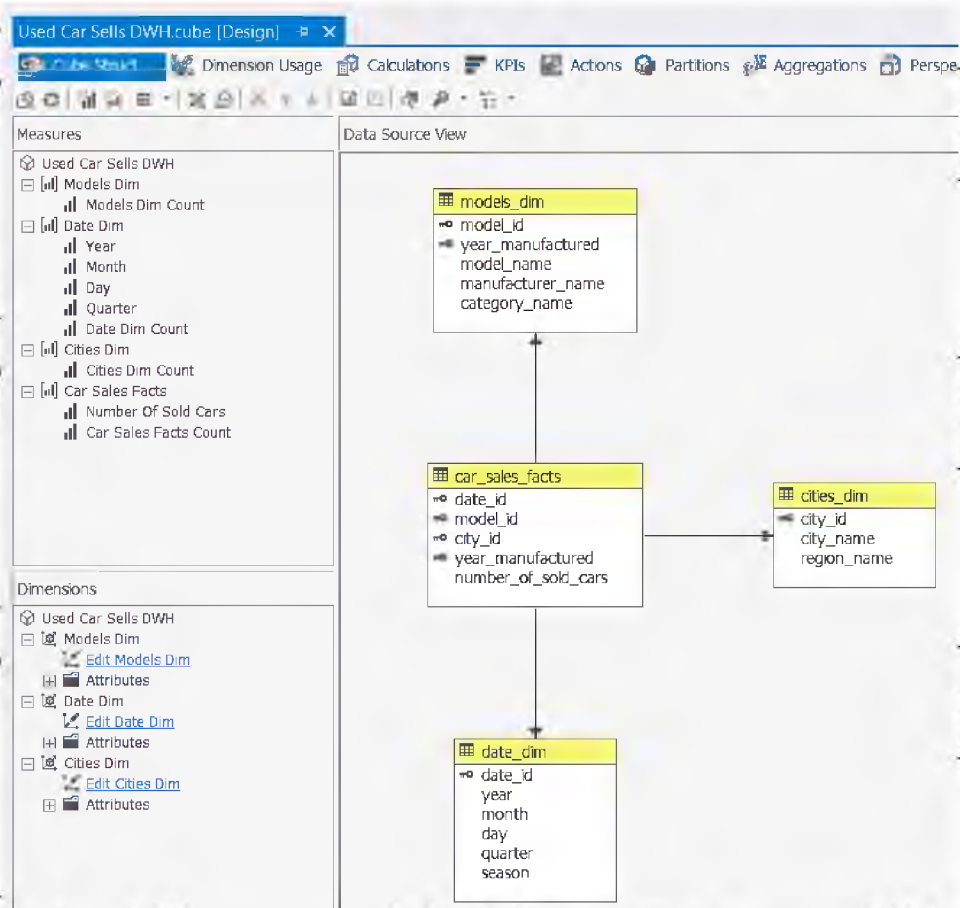


Рис. 9 – вигляд розгорнутого кубу

3.2.2. Реалізація отриманих даних за допомогою служби SSIS та інструментів Data Flow

SSIS (SQL Server Integration Services) – це інструмент, що дозволяє у зручному вигляді реалізувати інтеграцію, тобто реалізувати процес перенесення даних з одного джерела до іншого. Цей процес іноді називають ETL (від англ. Extract, Transform, Load - дослівно "витягнення, перетворення, завантаження").

Для створення проекту SSIS в середовищі Visual Studio необхідно встановити даний шаблон засобами extension manager, аналогічно до SSAS. Після створення проекту, необхідно виконати декілька кроків, а саме:

- Створити з'єднання з серверами на яких зберігаються БД та СД, для подальшого трансферу даних.
- Сформувані data flow task'и, в послідовності, в якій служба зможе внести дані (спочатку виміри, потім таблиці фактів, які посилаються на дані з вимірів).

Створені екземпляри підключення до БД та СД, а також сформовані data flow task'и для процесу трансферу даних наведені на рис. 10.

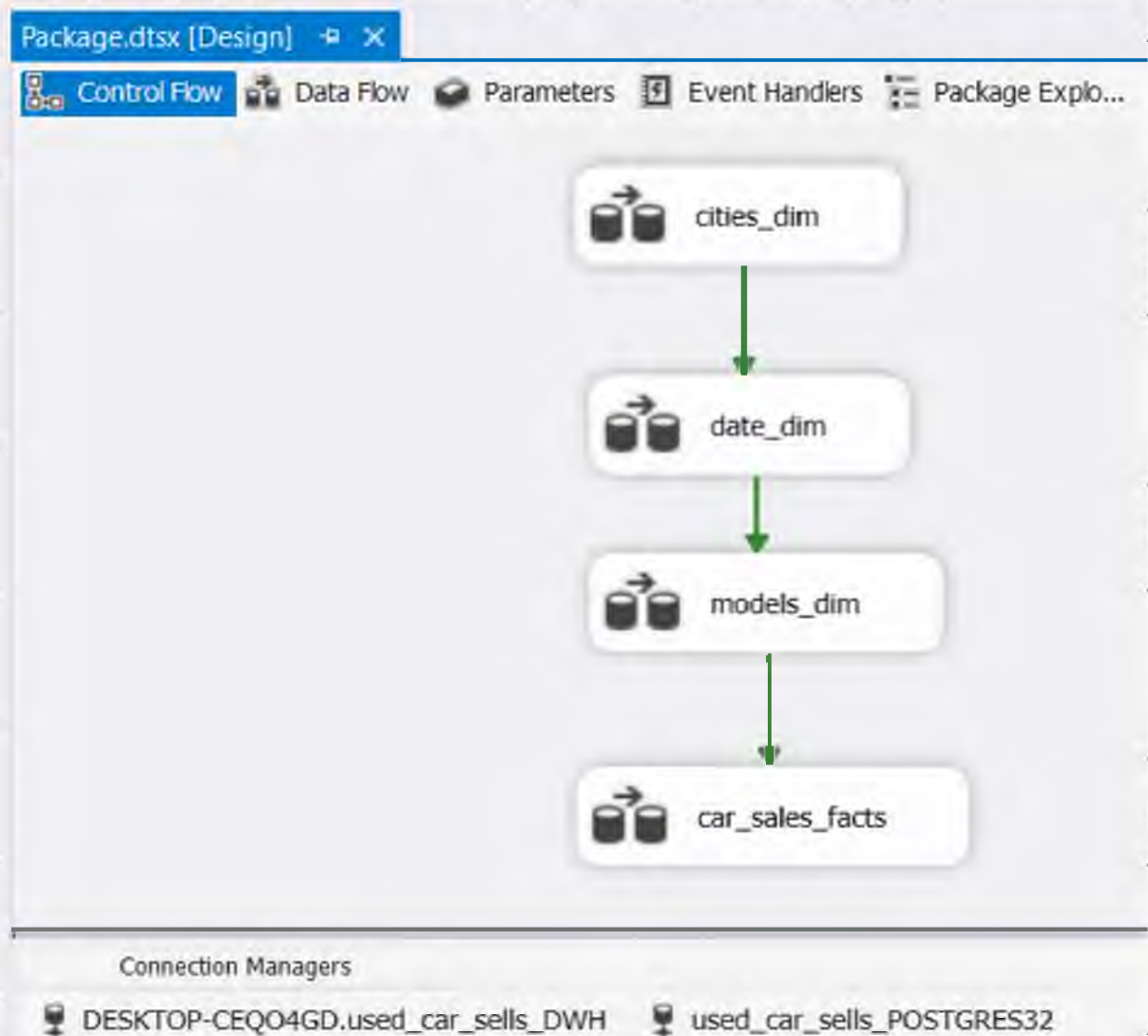


Рис. 10 – сформована послідовність data flow task'ів

На подальших рисунках (рис.11 – рис.14) зображені відповідності між таблицями в БД, та відповідними їм таблицями в СД, а також деякі запити мовою SQL, які сформувавши необхідні набори даних.

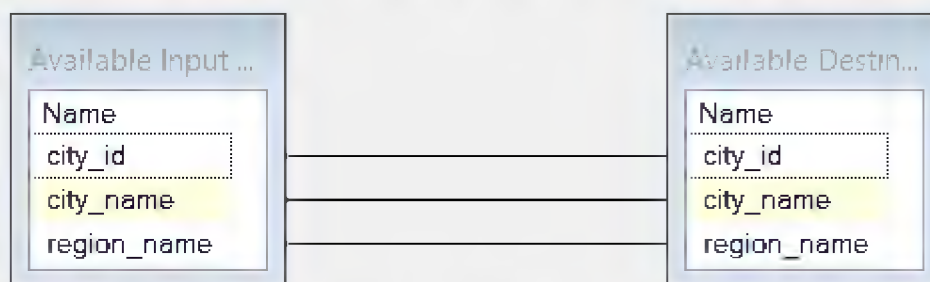


Рис. 11 – відповідність атрибутів між таблицею в БД, та відповідною їй в СД

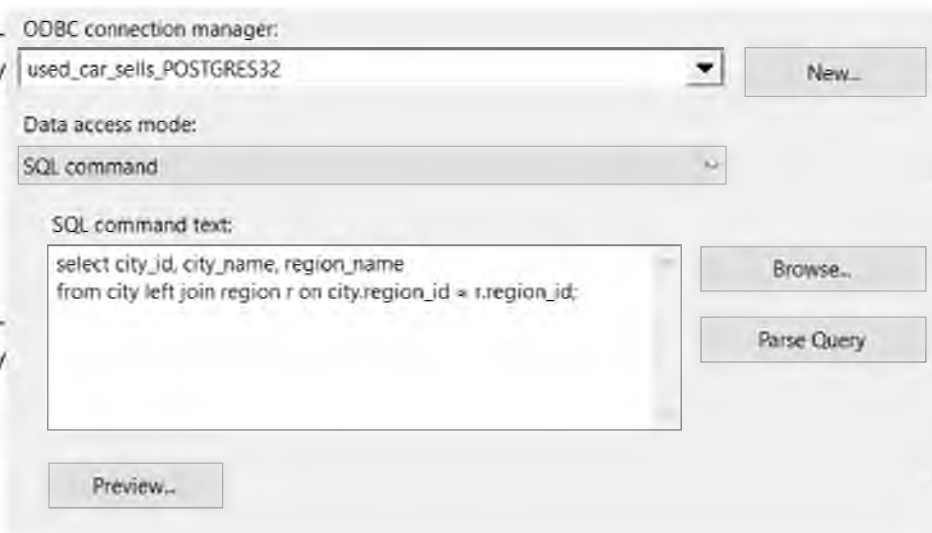


Рис. 12 – вигляд вікна властивостей ODBC Source та SQL запит для формування відповідного набору даних для таблиці СД



Рис. 13 – відповідність набору даних з БД до таблиці date dim в СД

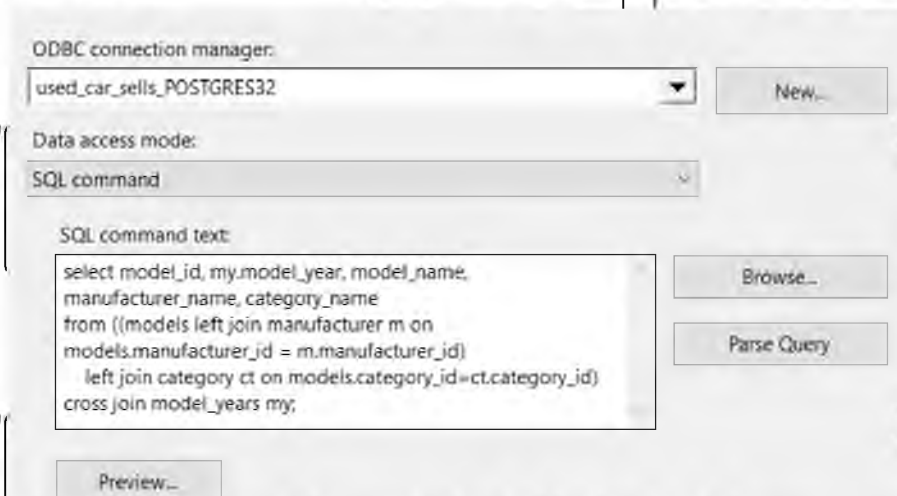


Рис. 14 – вигляд вікна властивостей ODBC Source та SQL запит для формування набору даних для таблиці models_dim

Після моделювання заповнення всіх data flow task'ів завданнями перенесення даних, а також співставлення всіх ODBC Source джерел (оскільки в якості СУБД використовувалась PostgreSQL) з SQL Server Destination (оскільки SQL Server було використано для зберігання СД) було запущено проект. Успішні результати виконання наведено на рис. 15 – рис. 17.

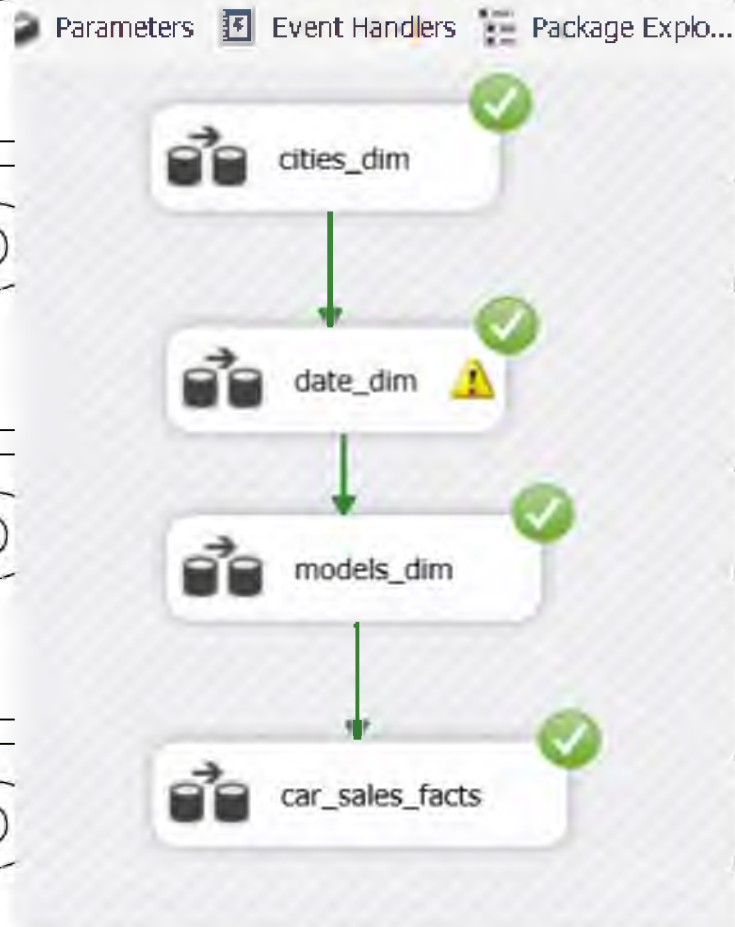


Рис. 15 – результат виконання проекту SSIS

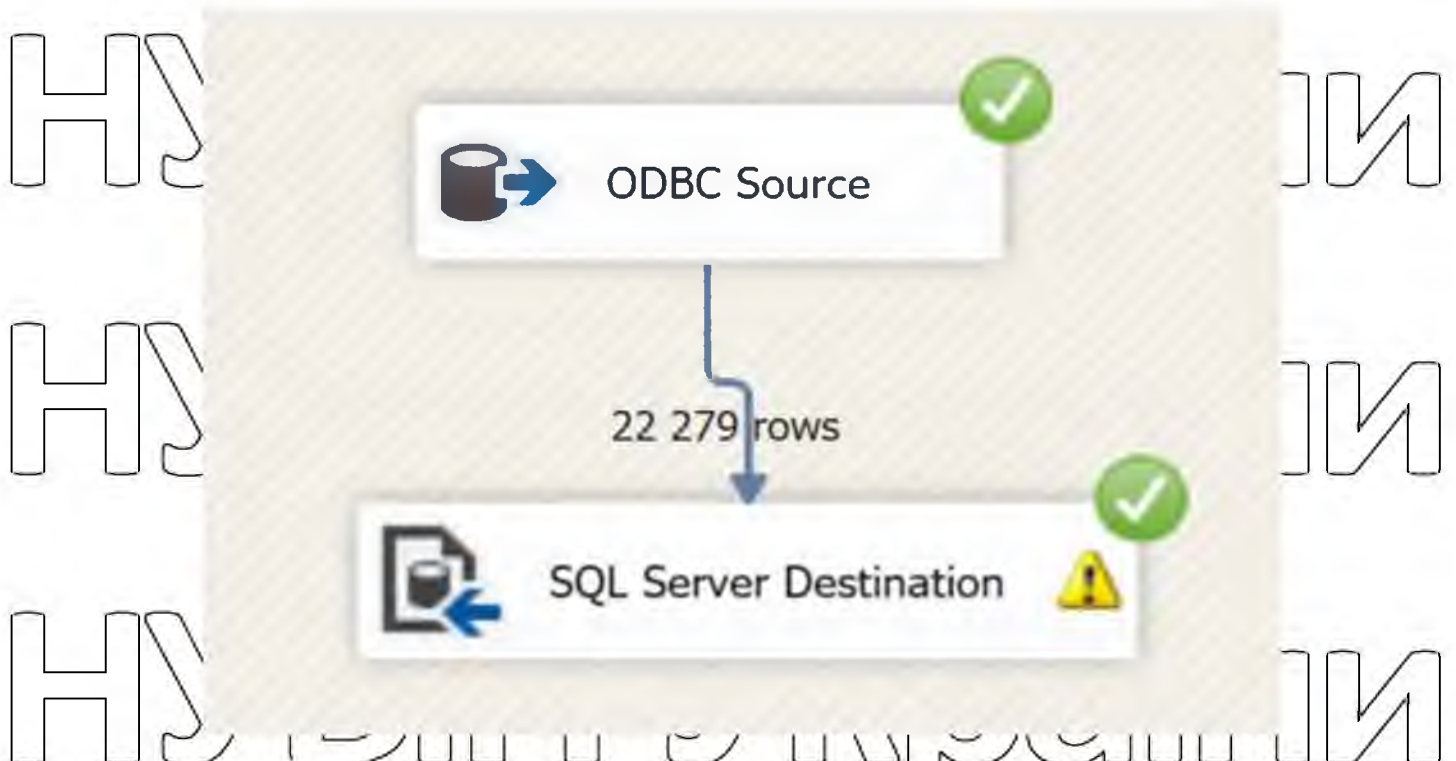


Рис. 16 – результат виконання трансферу даних всередині data flow task для date_dim



Рис. 17 - результат виконання трансферу даних всередині data flow task для model_dim

На рис.16 в модулі SQL Server Destination відображено жовтий знак оклику, що являє собою попередження про не співпадіння розмірів текстових слів в початковій таблиці, та в таблиці призначення. Виконання трансферу даних пройшло вдало не зважаючи на попередження.

НУБІП України

3.3. Реалізація процедури аналізу даних в розробленій системі

3.3.1. Побудова звітності в середовищі BI

Power BI — комплексне програмне забезпечення бізнес-аналізу (BI) компанії Microsoft, що об'єднує кілька програмних продуктів, що мають загальний технологічний та візуальний дизайн, з'єднувачі (шлюзи), а також web-сервіси. Power BI відноситься до класу self-service BI, та BI з резидентним обчисленням (англ. in-memory computing). Є частиною єдиної платформи Microsoft Power Platform.

Ключовий і найбільш розповсюджений продукт лінійки Power BI Desktop було використано для формування наступних звітів, наведених на рис. 18-рис.22.

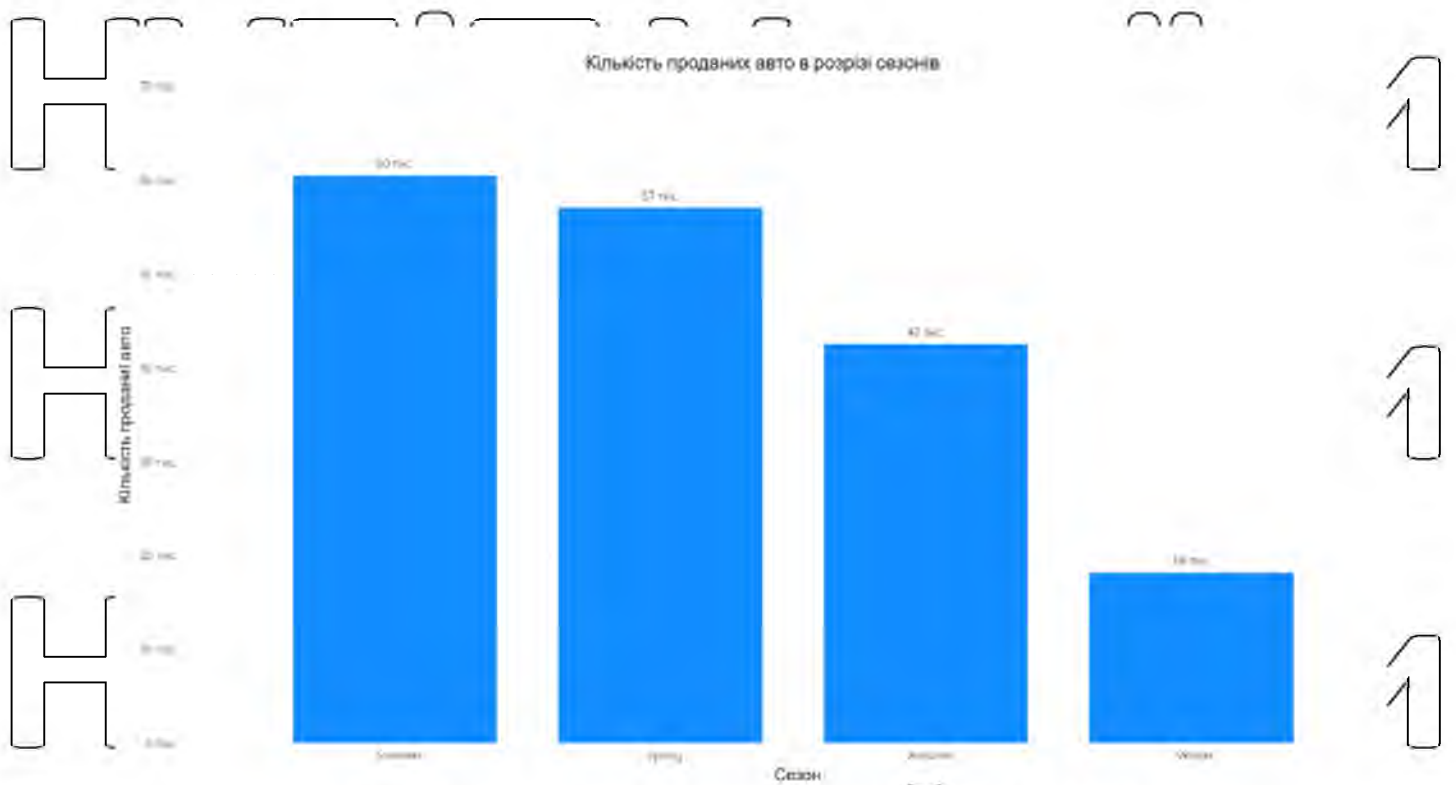


Рис. 18 – кількість проданих авто в розрізі сезонів

Дана картина сформована фактом того, що під час зимового сезону найменша кількість даних було зібрано, тому даний сезон не є репрезентативним.

Дане правило також частково застосовуються і до осіннього сезону, оскільки

кінець дослідження припав на осінь, і тому дані було зібрано не в повному обсязі. Дані за літній і весняний період були зібрані в повному обсязі, і тому графіки даних сезонів можна оцінювати об'єктивно.

Як бачимо з графіків, кількість проданих автомобілів в літній період дещо вища, що може бути пов'язано з збільшеною кількістю продажів кабриолетів.

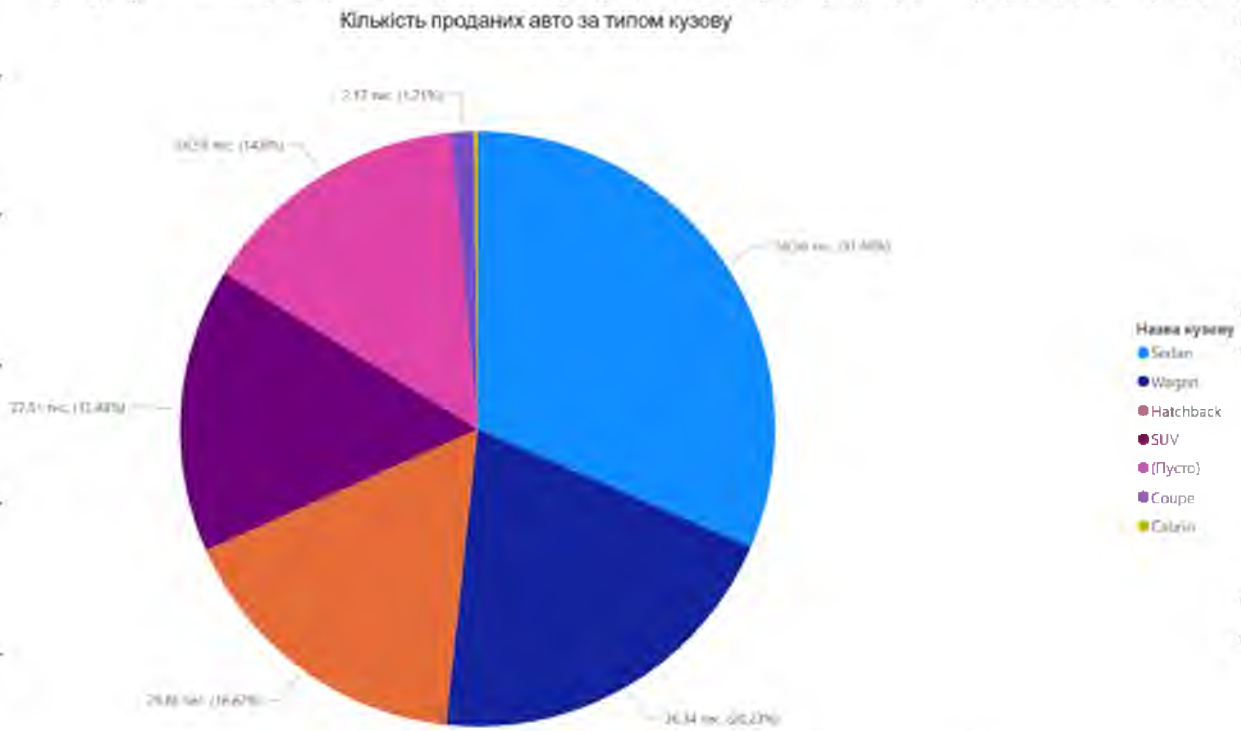


Рис. 19 – кількість проданих автомобілів за типом кузова

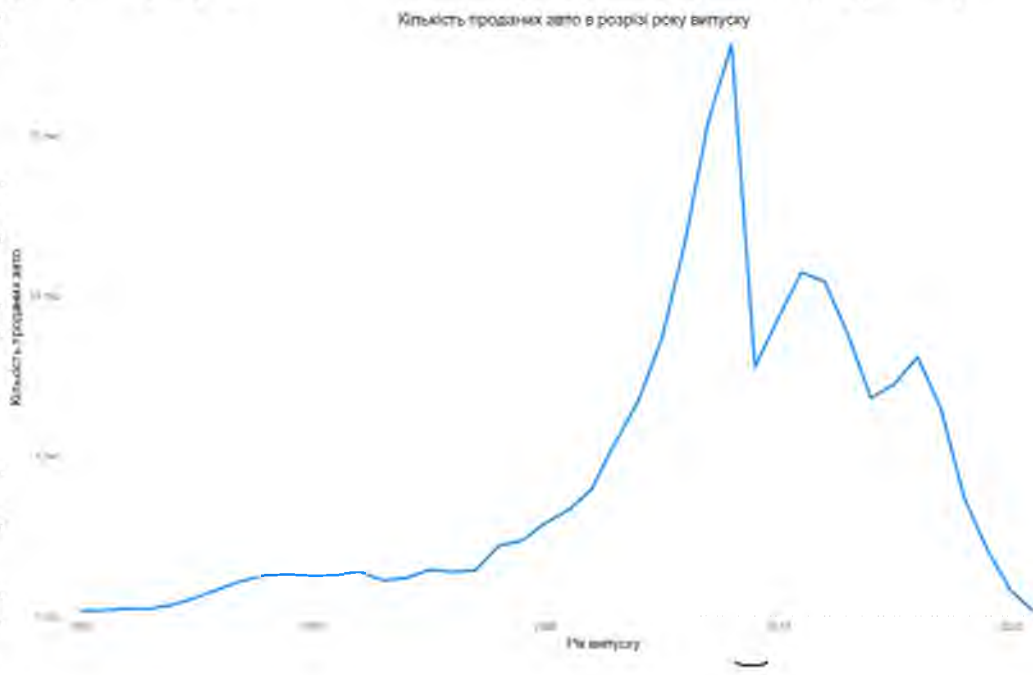


Рис. 20 — кількість проданих автомобілів в розрізі років випуску

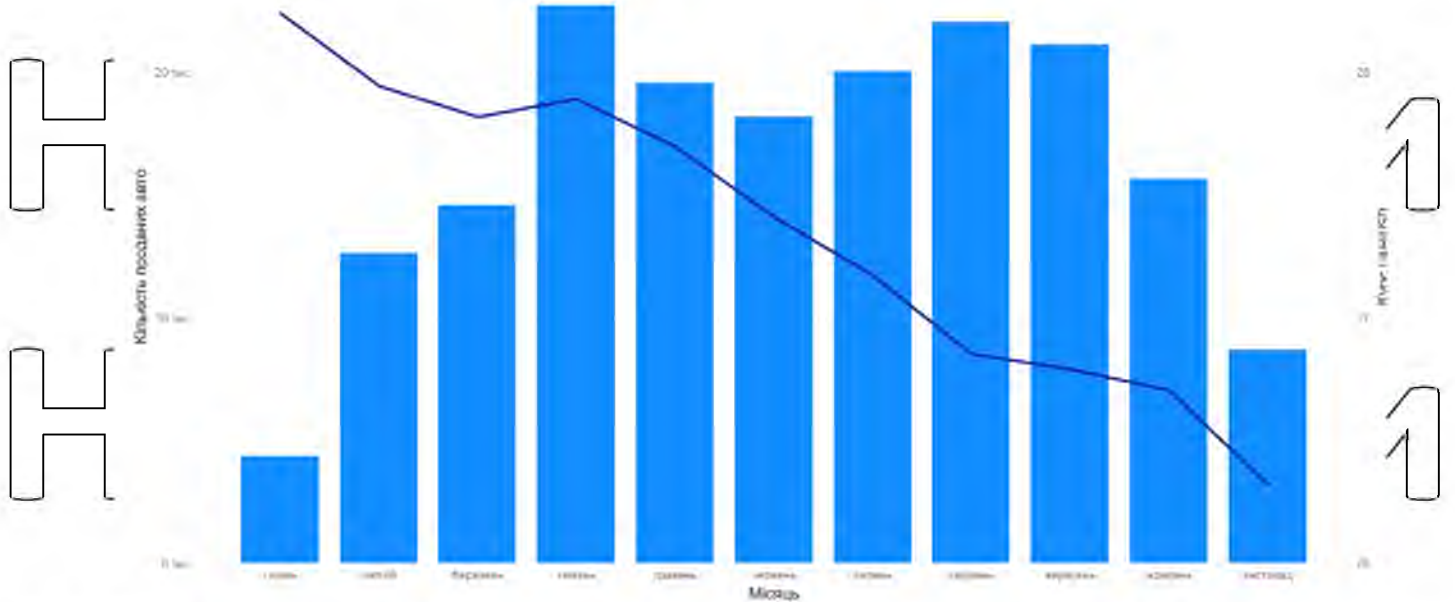
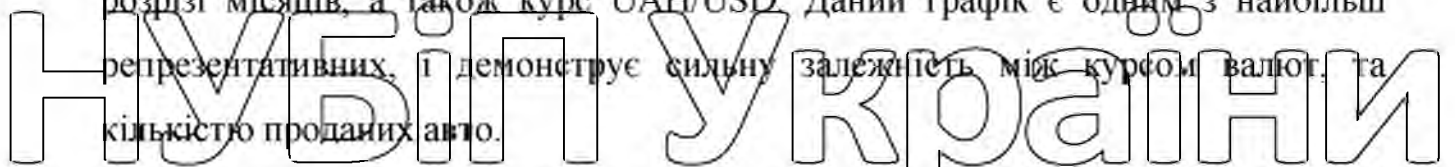


Рис. 21- поєднаний графік кількості проданих автомобілів в розрізі місяців, та графік зміни курсу USD/UAH



На рис. 21 зображено поєднаний графік кількості продажів автомобілів в розрізі місяців, а також курс UAH/USD. Даний графік є одним з найбільш репрезентативних, і демонструє сильну залежність між курсом валют, та кількістю проданих авто.



Вплив даного параметру на кількість проданих авто є очікуваним і передбачуваним, оскільки купівельна спроможність великих товарів в Україні напряму пов'язана з курсом долара, оскільки більшість населення надає перевагу зберіганню фінансів в валюті.



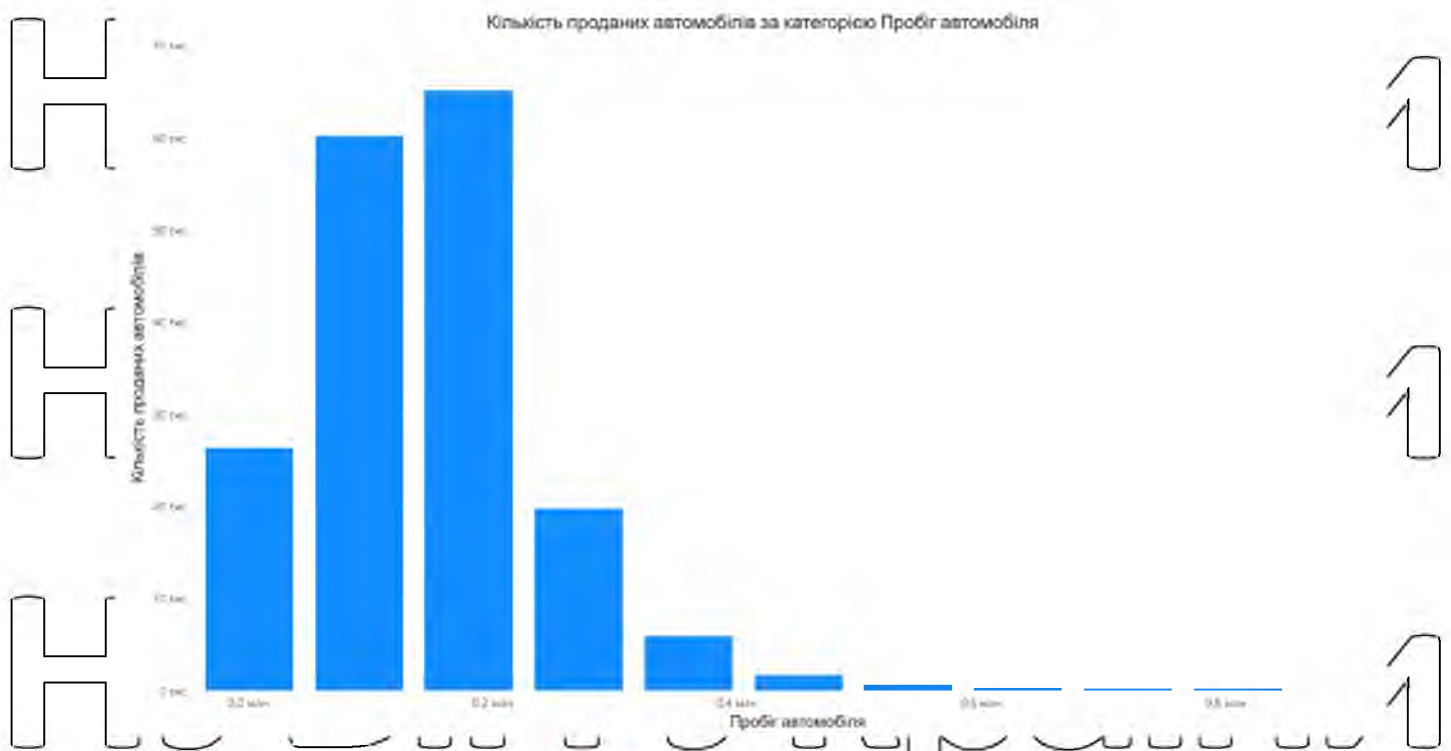


Рис. 22 – графік кількості проданих автомобілів в розрізі пробігу

3.3.2. Інтелектуальний аналіз даних Data Mining

Data Mining - це методологія і процес виявлення у великих масивах даних, що накопичуються в інформаційних системах компаній, раніше невідомих, нетривіальних, практично корисних та доступних для інтерпретації знань, необхідних для прийняття рішень у різних сферах людської діяльності. Data Mining є одним із етапів більш масштабної методології Knowledge Discovery in Databases.

Знання, виявлені у процесі Data Mining, мають бути нетривіальними і раніше невідомими. Нетривіальність передбачає, такі знання неможливо знайти виявлено шляхом простого візуального аналізу. Вони повинні описувати зв'язок між властивостями бізнес-об'єктів, передбачати значення одних ознак з урахуванням інших тощо. Знайдені знання мають бути застосовні і до нових об'єктів.

Практична корисність знань обумовлена можливістю їх використання у процесі підтримки прийняття управлінських рішень та вдосконалення діяльності компанії.

Знання мають бути представлені у вигляді, зрозумілому для користувачів, які не мають спеціальної математичної підготовки. Наприклад, найпростіше сприймаються людиною логічні конструкції «якщо, то». Більш того, такі правила можуть бути використані в різних СУБД як SQL-запитів. У разі, коли видобуті знання непрозорі для користувача, повинні існувати методи пост обробки, що дозволяють привести їх до виду, що інтерпретується.

Data Mining - це не один метод або технологія, а сукупність великої кількості різних методів виявлення знань. Всі завдання, які вирішуються методами Data Mining, можна умовно розбити на шість видів:

- Класифікація;
- Регресія;
- кластеризація;
- Асоціація;
- Послідовні шаблони;
- Аналіз відмилень.

Засоби SSAS, окрім можливостей використання OLAP кубу, також надають можливості використання технологій DataMining з використанням вищезгаданих методів. Список доступних для використання методів Data Mining в середовищі Visual Studio наведено на рис. 23.

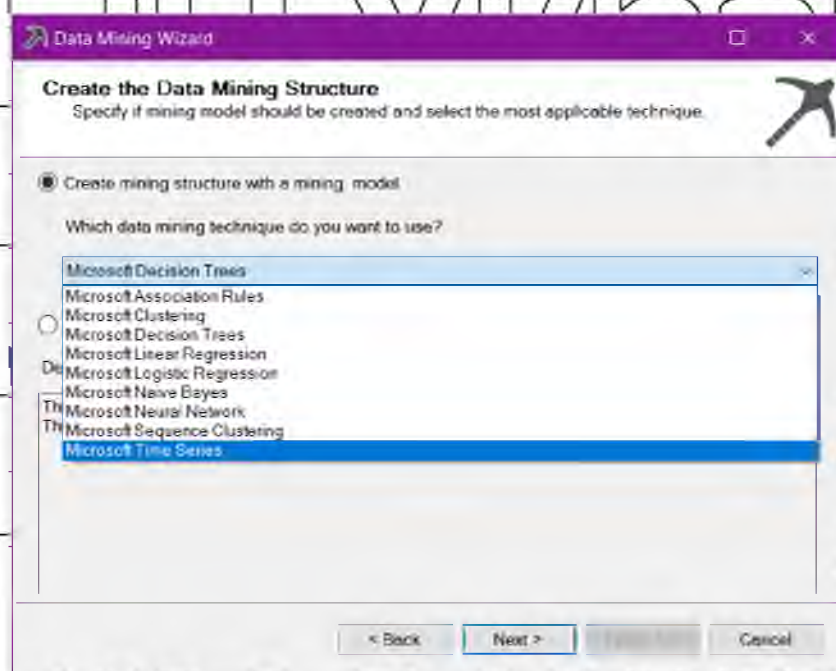


Рис. 23 – перелік доступних методів data mining

Оскільки продажі автомобілів прив'язані до часового виміру, було прийнято рішення використати методику Time Series, тобто аналіз на основі часових рядів. Для зручного сприйняття даних, було створено уявлення, яке містило в собі дані про продажі автомобілів за кожен день, незалежно від міста продажу. Дане рішення було зумовлене тим, що кількість даних, які було отримано, не є достатньою для проведення повноцінного процесу data mining, оскільки які не враховувати міста з великою кількістю населення (т.з. «мільйонники»), то порядок проданих автомобілів за день, в одному місті може складати десятки, або сотні. Також, враховуючи, що дані зібрано за один не певний рік, виходить що модель повинна була використати мізерний об'єм даних для роботи (а враховуючи той факт, що для підвищення точності передбачення, вхідний набір даних розділяється на навчальну та контрольну вибірку, то кількість даних для тренування моделі була б мізерною), що є контрпродуктивним при використанні методів data mining.

В результаті виконання mining structure, було отримано наступний графік (рис. 24), який демонструє кількість проданих автомобілів за день, а також прогнозований рівень продажів на наступний тиждень.

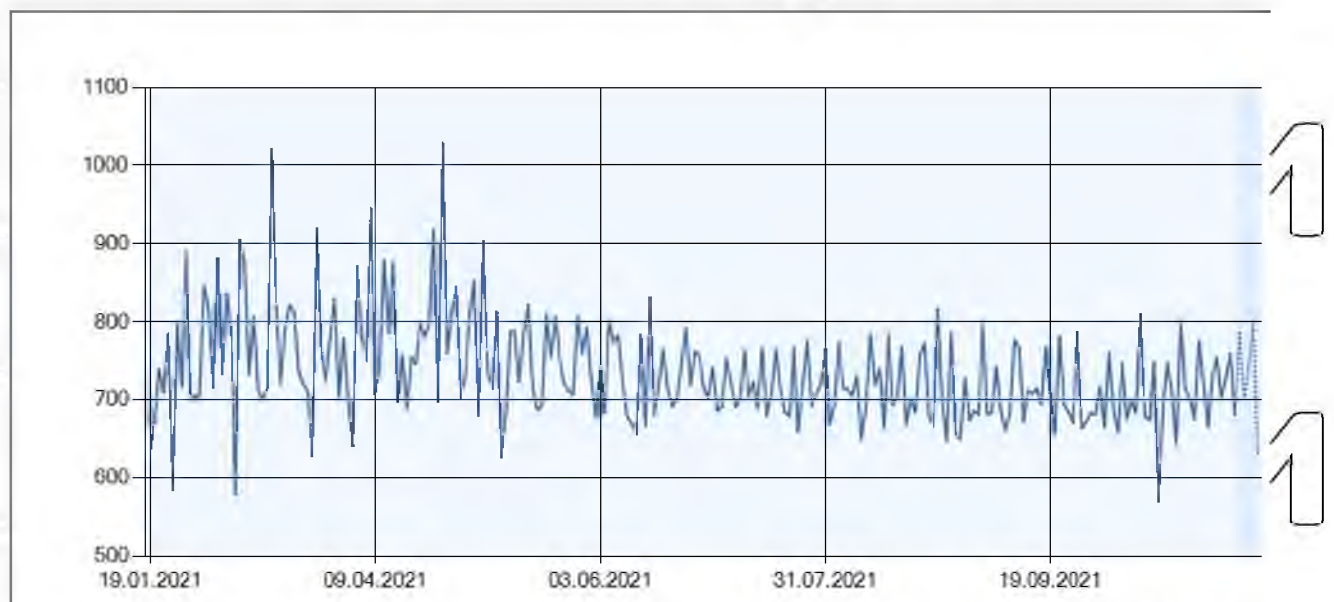


Рис. 24 – графік продажів автомобілів та прогноз на основі методу Time Series

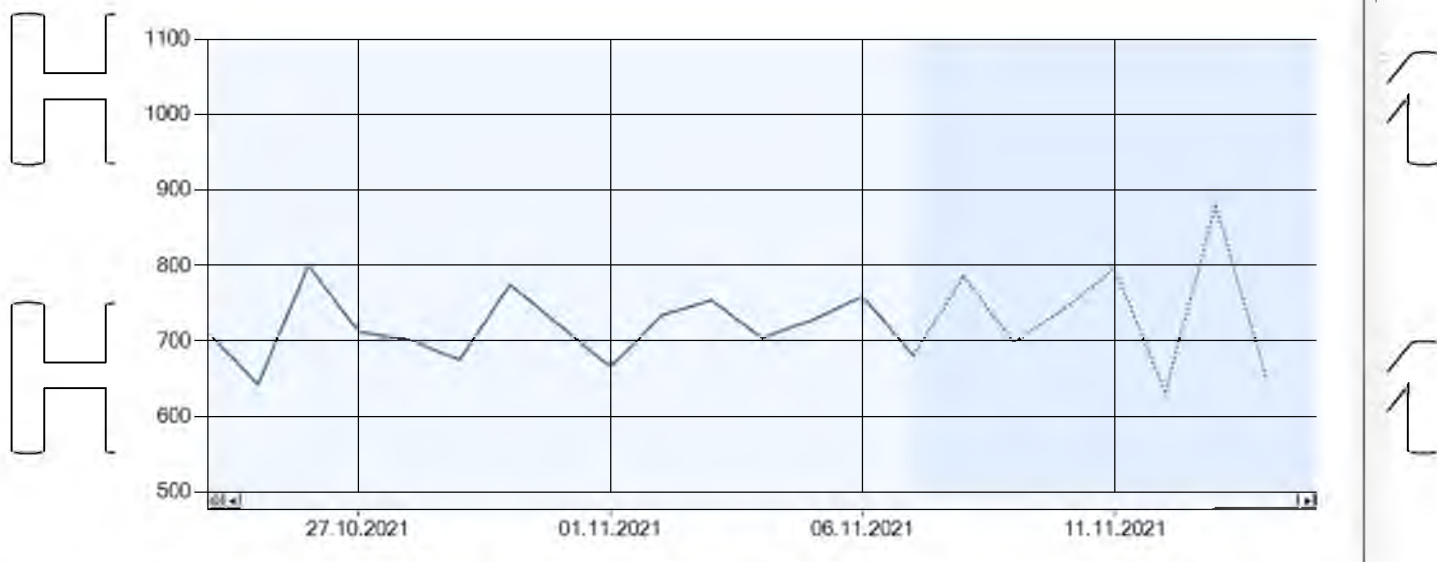


Рис. 15 — прогноз продажу автомобілів на основі моделі Time Series

Як і було зазначено вище, оскільки дані були зібрано за відносно малій (за мірками Data Mining) проміжок часу, передбачення може бути не зовсім точним, а також дещо спотвореним, через викиди, які видно на рис. 24, спричинені збоjami в роботі скрипта отримання даних з зовнішнього джерела.

3.3.3. Аналіз даних за допомогою засобів pandas та numpy

Pandas — програмна бібліотека, написана для мови програмування Python для маніпулювання даними та їхнього аналізу. Вона, зокрема, пропонує структури даних та операції для маніпулювання чисельними таблицями та часовими рядами. pandas є вільним програмним забезпеченням, що випускається за ліцензією BSD. Ця назва походить від терміну «панельні дані» (англ. panel data), який в економетрії позначає багатовимірні структуровані набори даних.

Numpy розширення мови Python, що додає підтримку великих багатовимірних масивів і матриць, разом з великою бібліотекою високорівневих математичних функцій для операцій з цими масивами. Попередник Numpy, Numeric, був спочатку створений Jim Hugunin. Numpy — відкрите програмне забезпечення і має багато розробників.

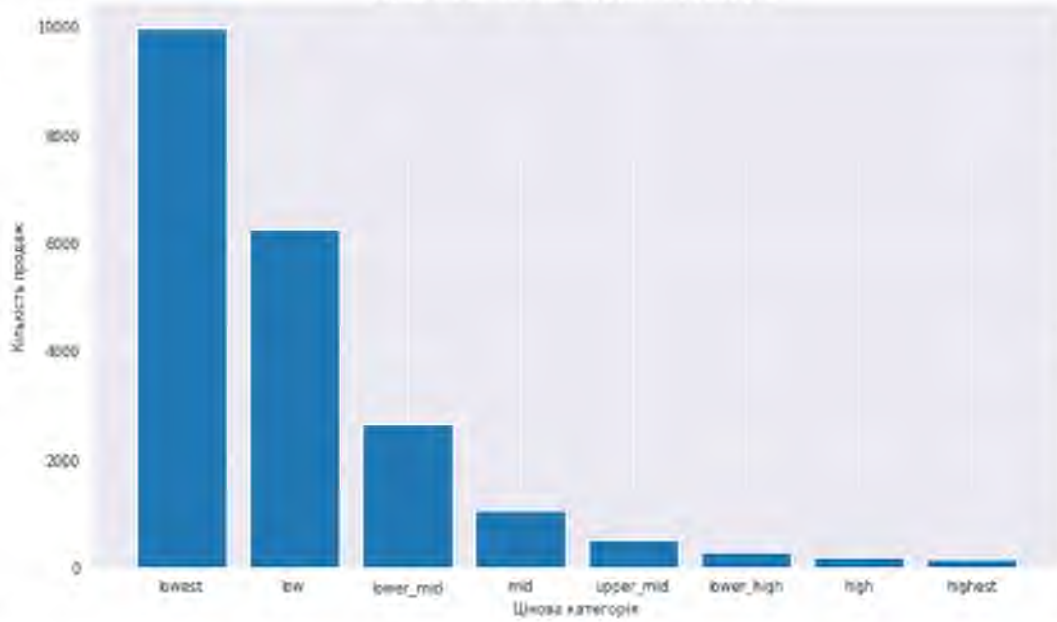
Оскільки Python — інтерпретована мова, математичні алгоритми, часто працюють в ньому набагато повільніше ніж у компільованих мовах, таких як C

або навіть Java. NumPy намагається вирішити цю проблему для великої кількості обчислювальних алгоритмів забезпечуючи підтримку багатовимірних масивів і безліч функцій і операторів для роботи з ними. Таким чином будь-який алгоритм, який може бути виражений в основному як послідовність операцій над масивами і матрицями, працює так само швидко, як еквівалентний код, написаний на C.

NumPy можна розглядати як гарну вільну альтернативу MATLAB, оскільки мова програмування MATLAB зовні нагадує NumPy: обидві вони інтерпретовані, і обидві дозволяють користувачам писати швидкі програми поки більшість операцій проводяться над масивами або матрицями, а не над скалярами. Перевага MATLAB у великій кількості доступних додаткових інструментів, включаючи такі як пакет Simulink. Основні пакети, що доповнюють NumPy, це: SciPy — бібліотека, що додає більше MATLAB-подібної функціональності; Matplotlib — пакет для створення графіки в стилі MATLAB. Внутрішньо як MATLAB, так і NumPy базуються на бібліотеці LAPACK, призначеній для вирішення основних задач лінійної алгебри.

Через вищеписані переваги, дані пакети, як і сама мова програмування Python є доволі популярними у сфері Data Analysis, оскільки мають потужні інструменти роботи з великими даними, а також велику спільноту розробників, відкритого коду, та документації.

Продажі в залежності від цінних категорій



lowest = 100.0 - 6338.0
 low = 6338.0 - 12575.0
 lower_mid = 12575.0 - 18812.0
 mid = 18812.0 - 25050.0
 upper_mid = 25050.0 - 31288.0
 lower_high = 31288.0 - 37525.0
 high = 37525.0 - 43762.0
 highest = 43762.0 - 50000.0

Рис. 26 – кількість проданих автомобілів за цінною категорією

Продажі по сезонах в розрізі кузовів

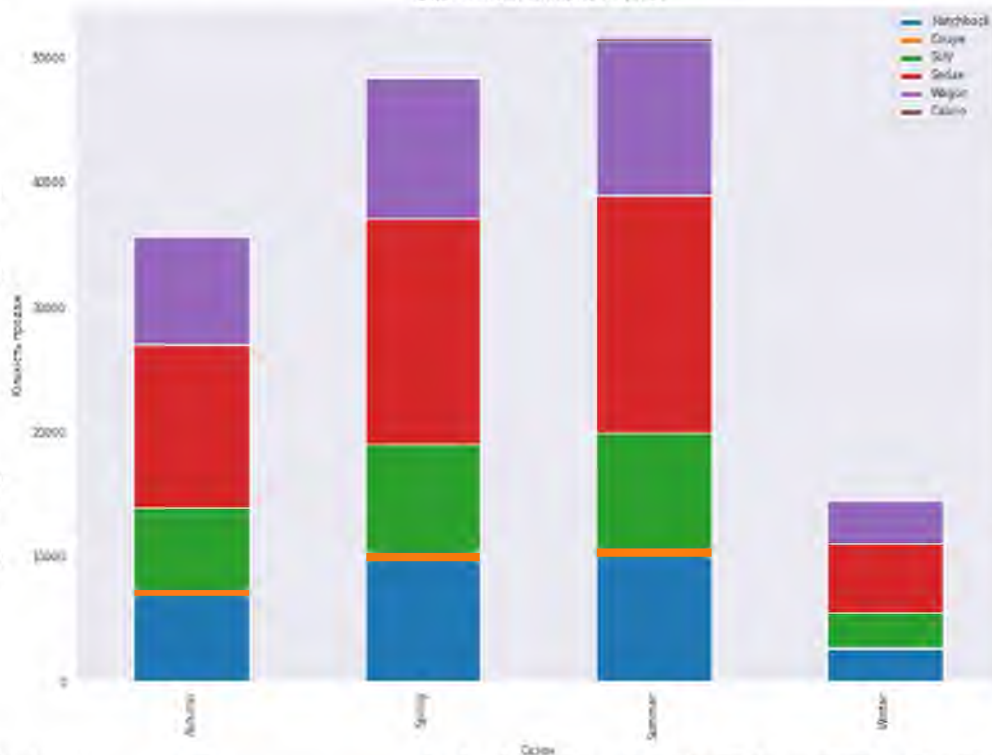
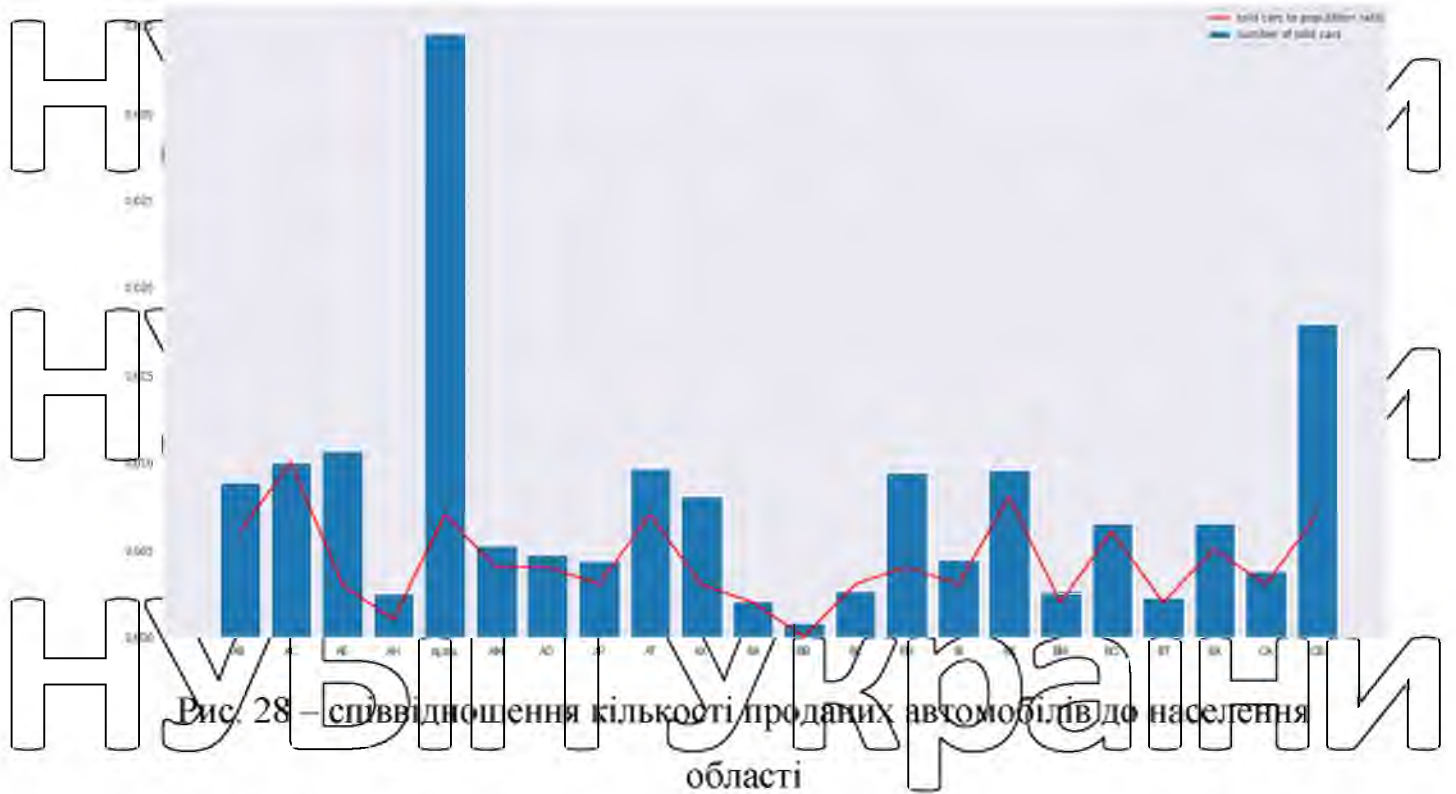


Рис. 27 – графік продажів авто за типом кузова в розрізі сезону



На рис.28 наведено цікавий графік, який відображає кількість проданих автомобілів (синя гістограма) поділені на 1,000,000 (для масштабу, та можливості накладення супутнього графіку), а також графік співвідношення кількості проданих автомобілів до кількості населення. Таким чином вдалось нівелювати кількість населення, і оцінити реальну кількість продаж на душу населення області.

НУБІП України

НУБІП України

НУБІП України

	reg_name	population	sold_cars	sc2pop_ratio	regpop2allpop_ratio	overalsc2regsc_ratio	car_code
1	Vinnytska	1545416	8905	0.006	3.77	5.25	AB
10	Volynska	1031421	10039	0.010	2.52	5.91	AC
0	Dnipropetrovska	3176648	10680	0.003	7.75	6.29	AE
11	Donetska	4131808	2524	0.001	10.08	1.49	AH
17	Kyivska	4748404	34582	0.007	11.58	20.37	AI/AA
14	Zhytomyrska	1208212	5222	0.004	2.95	3.08	AM
4	Zakarpatska	1253791	4739	0.004	3.06	2.79	AO
9	Zaporizka	1687401	4370	0.003	4.12	2.57	AP
21	Ivano-Frankivska	1368097	9666	0.007	3.34	5.69	AT
7	Kharkivska	2658461	8070	0.003	6.48	4.75	AX
3	Kirovohradska	933109	2034	0.002	2.28	1.20	BA
13	Luhanska	2135913	772	0.000	5.21	0.45	BB
8	Chernihivska	991294	2668	0.003	2.42	1.57	BC
18	Odeska	2377230	9377	0.004	5.80	5.52	BH
22	Poltavska	1386978	4414	0.003	3.38	2.60	BI
12	Rivnenska	1152961	9580	0.008	2.81	5.64	BK
15	Mykolajivska	1119862	2620	0.002	2.73	1.54	BM
5	Sumska	1068247	2493	0.002	2.61	1.47	BN
16	Ternopilska	1038695	6483	0.006	2.53	3.82	BO
2	Khersonska	1027913	2279	0.002	2.51	1.34	BT
6	Khmelnytska	1254702	6510	0.005	3.06	3.83	BX
19	Cherkaska	1192137	3771	0.003	2.91	2.22	CA
20	Lvivska	2512084	17967	0.007	6.13	10.58	CB

Рис. 29 – таблиця відношень кількості проданих авто до кількості населення в

Як бачимо з таблиці та графіку, найвищі значення в Волинській, Рівненській, Львівській, Івано-Франківській та Київській областях, що свідчить що найбільша кількість проданих авто на душу населення знаходиться на заході країни, а також в столиці.

3.4. Результати дослідження

В результаті виконання дослідження було впрокремлено наступні критерії, які впливають на кількість продажів вживаних автомобілів: курс іноземної валюти, пробіг автомобіля, місце розташування. Також за допомогою засобів SSAS та бібліотеки scipy було сформульоване прогнозу щодо продажу автомобілів на найближчий час.

ВИСНОВКИ

В ході виконання дипломної роботи магістра було спроектовано і створено сховище даних на основі реляційної бази даних для аналізу ринку вживаних автомобілів в Україні. Для аналізу було використано дані отримані з сайту auto.nia.com

Для роботи було використано наступні інструменти

- Visual studio code – текстовий редактор
- Інтерпретатор Python 3.9 для запуску скриптів отримання даних та аналізу мовою python
- DBeaver – SQL клієнт, засіб управління БД
- MS SQL Server Management Studio – для роботи з сховищем даних, а також роботи з об'єктами Analysis Services
- MS Visual Studio – для роботи з проектами SSAS та SSIS
- Power BI Desktop – для створення звітності

Всі дані мігрували з бази в створене за схемою «зірка» сховище даних у СУБД MS SQL Server. Було розгорнуто куб, створено потоки та заповнено куб даними за допомогою Data Flow в середовищі SSIS. Після цього було звіти в середовищі Power BI від Microsoft. Також було проведено аналіз даних за допомогою методу Data Mining, а також розвідувальний аналіз даних за допомогою бібліотек pandas та numpy.

На основі результатах дослідження було виділено ті показники, які на пряму впливають на кількість продажів автомобілів.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Голуб Б.Л., Ящук Д.Ю. Організація сховища даних: навч. посіб. Київ: Національний університет біоресурсів і природокористування України, 2018. 150 с.

2. Голуб Б.Л., Ящук Д.Ю. Методичні вказівки до виконання курсового проекту з дисципліни «Організація сховища даних» Київ: Національний університет біоресурсів і природокористування України, 2018. 20 с.

3. Andrew Bruce, Peter Bruce, Practical Statistics for Data Scientists: 50 Essential Concepts, O'Reilly Media, Inc., 2017.

4. Roger Peng, Elizabeth Matsui, The Art of Data Science, Lulu.com., 2016.

5. Jake VanderPlas, Python Data Science Handbook, O'Reilly Media, Inc., 2017.

6. Барсегян А. А., Куприянов М. С., Степаненко В. В., Холод И. И. Методы и модели анализа данных. OLAP и Data Mining. СПб.: БХВ-Петербург, 2004. — 336 с..

7. James Rumbaugh, Ivar Jacobson, Grady Booch (1999). The unified modeling language reference manual (англ.). Addison Wesley Longman Inc. ISBN 0-201-30998-X.

8. Сховища та простори даних: монографія / Н. Б. Шаховська, В. В. Насичник ; М-во освіти і науки України, Нац. ун-т «Львів. політехніка». Ль. : Вид-во Нац. ун-ту «Львів. політехніка», 2009. — 240 с. — Бібліогр. : с. 230–240

(207 назв) — ISBN 978-966-553-796-0

9. SQL Server Business intelligence [Електронний ресурс]. — Режим доступу: <https://www.microsoft.com/ru-ru/sql-server/sql-business-intelligence>

10. Бабич А. В. Введение в UML. ISBN 978-5-94774-878-9, 6. Лекция: Диаграммы прецедентов: крупным планом / А. В. Бабич // Национальный открытый университет «Интуит». Режим доступу:

<https://intuit.ru/studies/courses/1007/229/lecture/5962>

НУБІП України

НУБІП України

НУБІП України

НУБІП України

НУБІП України

НУБІП України

НУБІП України

НУБІП України
ДОДАТОК А

НУБІП України

НУБІП України
Тези на тему:
Аналіз автомобільного ринку України.

Матеріали з XI Міжнародної науково-практичної конференції студентів,
аспірантів та молодих вчених «ІНФОРМАЦІЙНІ ТЕХНОЛОГІЇ: ЕКОНОМІКА,
ТЕХНІКА, ОСВІТА», 10-11 листопада 2020 року, НУБІП України, Київ.

НУБІП України

НУБІП України

НУБІП України

НУБІП України

НУВБІП України

АНАЛІЗ АВТОМОБІЛЬНОГО РИНКУ УКРАЇНИ

Ганаєтук І.О., науковий керівник Улазунова О.Г.

Автомобільний ринок відноситься до однієї з найбільш розвинутих галузей української економіки. Дослідження проблем розвитку автомобільного ринку України представляє суттєвий науковий та практичний інтерес, зважаючи зростаюче значення автомобільного бізнесу для економіки країни та посилення конкуренції серед представників даного ринку. Проте, незважаючи на таку актуальність автомобільної галузі, засобів для аналізу ринку, зокрема ринку вживаних автомобілів майже не існує. Таким чином, дослідження стану автомобільного ринку України є необхідною умовою для розуміння ситуації, що відбувається на даному ринку в цілому, а також для виявлення основних тенденцій, динаміки та ємності ринку, які в майбутньому можуть допомогти спрогнозувати тенденції на ринку, та принести користь покупцям, які, поклавшись на результати дослідження змогли б зробити висірі у відповідності до своїх інтересів та вимог.

Згідно дослідження, проведеного компанією GfK Ukraine, під час кризи на автомобільному ринку України важливість суб'єктивних чинників при ухваленні рішення щодо здійснення купівлі зросла (рис. 1).

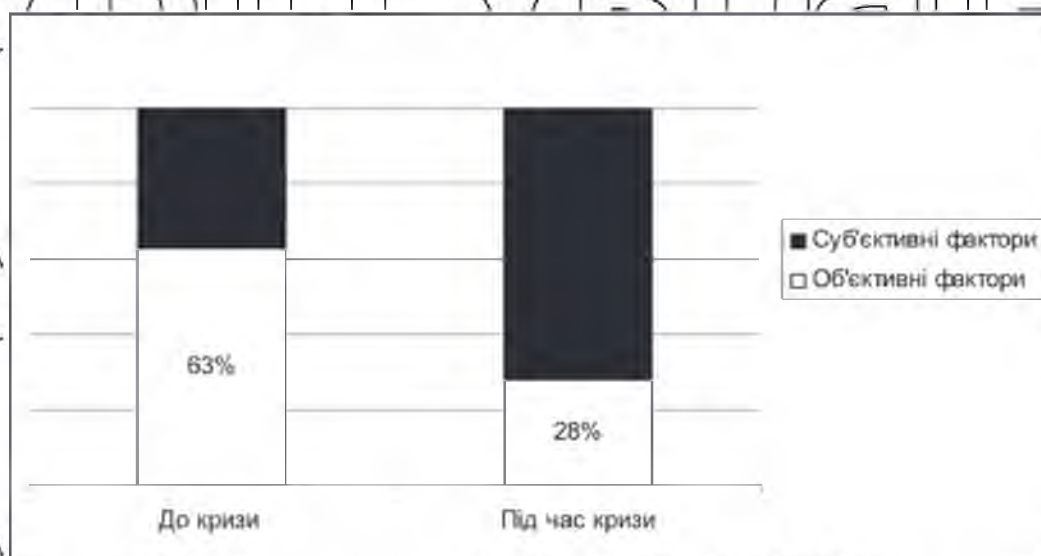


Рис. 1- графік впливу об'єктивних та суб'єктивних факторів на рішення про

покупку нового автомобіля

Слід зазначити, що ситуація на ринку нових автомобілів напряду впливає на ринок вживаних автомобілів оскільки, середній термін експлуатації нового автомобіля становить 3-5 років в європейських країнах, та 5-6 років в Україні. Це означає, що автомобілі які були популярними на ринку нових автомобілів 5 років тому, в теперішній момент починають з'являтися на вторинному ринку.

Дану проблему можна вирішити за допомогою відкритих даних щодо оголошень розміщених на веб-ресурсах для продажу автомобілів. Одним з ресурсів, який надає такого роду інформацію є Auto.RIA. Дані надаються в форматі JSON за допомогою API до якого можна здійснити 1000 звернень на годину. Завдяки цим даним можна будувати графіки зміни ціни (рис. 2) та кількості продаж конкретної моделі за певний період, для проведення подальшого кореляційного аналізу.

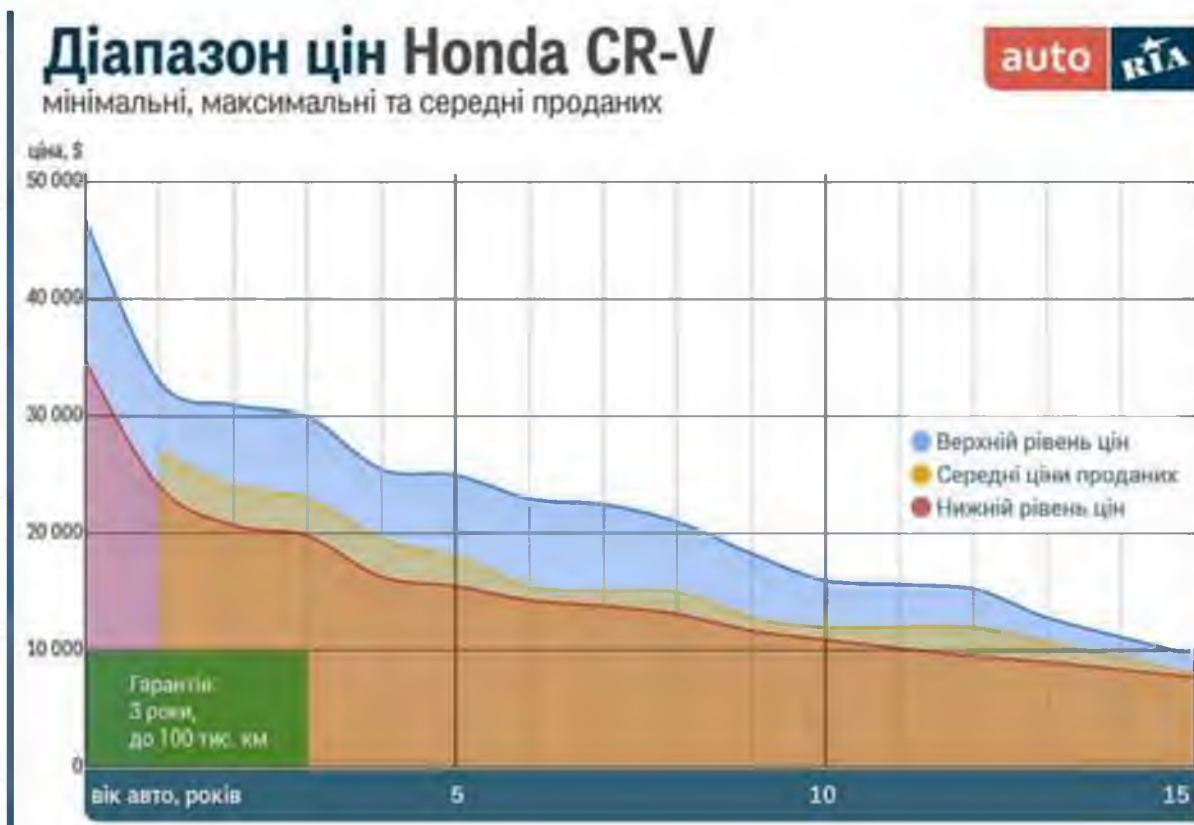


Рис 2 – графік зміни ціни авто в залежності від віку, на прикладі моделі Honda

Після накопичення великої кількості даних про продажі автомобілів в базу даних, постає питання про інструменти аналізу. З плином часу, в індустрії IT сформувалися

деякі методики та правила щодо роботи з великими обсягами даних, а саме їхнім зберіганням та аналізом. Дану область ІТ так і називають, Big Data.

Одним з найбільш поширених та простих для використання з точки зору користувача рішень в даному секторі, являється використання серії продуктів від компанії Microsoft під назвою BI (Business intelligence). Дане поняття об'єднує в собі інструменти, які дозволяють організувати сховища даних (СД), а також проводити аналіз на основі СД, з можливостями прогнозування за різними алгоритмами, візуалізації тенденцій в даних, і розрахунків KPI (Key performance indicator – ключовий показник ефективності).

Використання СД передбачає особливу структуру зберігання накопичених даних, найчастіше передбачається групування масивів даних за певним показником (найчастіше за часовим виміром). Дані в СД, на відміну від БД не передбачають можливості швидкого доступу і зміни, оскільки дані операції потребують значних ресурсів СУБД.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Луценко О. Автоексперт - 2009: результати експертного опитування топ-менеджменту українських автомобільних компаній / Оксана Луценко // Маркетинг в Україні. - 2009. - № 1. - С. 13-14.

2. Автомобільна статистика України (стислний огляд) за 2008 рік [Електронний ресурс] // Офіційний сайт асоціації автовиробників України.

Режим

доступу:

URL:http://ukrautoprom.com.ua/index.php?option=com_docman&task=cat_view&gid=27&dir=DESC&order=name&Itemid=79&limit=5&limitstart=65 – Заголовок з екрану.

3. API Portal ria.com для розробників [Електронний ресурс] // портал для розробників групи ресурсів ria.com. Режим доступу: <https://developers.ria.com/>

НУБІП України

ДОДАТОК Б

НУБІП України

НУБІП України

SQL код створення бази даних

НУБІП України

НУБІП України

НУБІП України

НУБІП України


```
CREATE TABLE public.body_type (  
  body_id int4 NOT NULL,  
  body_name varchar(50) NULL,  
  body_name_eng varchar(100) NULL,  
  CONSTRAINT body_type_pkey PRIMARY KEY (body_id)  
);
```

```
CREATE TABLE public.category (  
  category_id int4 NOT NULL,  
  category_name varchar(50) NULL,  
  CONSTRAINT category_pkey PRIMARY KEY (category_id)  
);
```

```
CREATE TABLE public.fuel_type (  
  fuel_id int4 NOT NULL,  
  fuel_name varchar(50) NULL,  
  CONSTRAINT fuel_type_pkey PRIMARY KEY (fuel_id)  
);
```

```
CREATE TABLE public.gearbox_type (  
  gearbox_id int4 NOT NULL,  
  gearbox_name varchar(50) NULL,  
  CONSTRAINT gearbox_type_pkey PRIMARY KEY (gearbox_id)  
);
```

```
CREATE TABLE public.manufacturer (  
  manufacturer_id int4 NOT NULL,  
  manufacturer_name varchar(50) NULL,  
  CONSTRAINT manufacturer_pkey PRIMARY KEY (manufacturer_id)  
);
```

```
CREATE TABLE public.model_years (  
  model_year int4 NOT NULL,  
  CONSTRAINT model_years_pkey PRIMARY KEY (model_year)  
);
```

```
CREATE TABLE public.region (  
  region_id int4 NOT NULL,  
  region_name varchar(50) NULL,  
  CONSTRAINT region_pkey PRIMARY KEY (region_id)  
);
```

```
CREATE TABLE public.seasons (  
  season_id int4 NOT NULL,  
  season_name varchar(50) NULL,  
  CONSTRAINT seasons_pkey PRIMARY KEY (season_id)  
);
```

```
CREATE TABLE public.wheel_drive (  
  wheel_drive_id int4 NOT NULL,  
  wheel_drive_name varchar(50) NULL,  
  CONSTRAINT wheel_drive_pkey PRIMARY KEY (wheel_drive_id)  
);
```

```
CREATE TABLE public.city (  
  city_id int4 NOT NULL,  
  city_name varchar(50) NULL,  
  region_id int4 NULL,  
  CONSTRAINT city_pkey PRIMARY KEY (city_id),  
  CONSTRAINT fk_city_region_id FOREIGN KEY (region_id)  
REFERENCES public.region(region_id)  
);
```

```
CREATE TABLE public.dates (  
  date_id int4 NOT NULL,  
  "day" int4 NULL,  
  "month" int4 NULL,  
  "year" int4 NULL,  
  date_type date NULL,  
  season_id int4 NULL,  
  quarter int4 NULL,  
  CONSTRAINT dates_pkey PRIMARY KEY (date_id),  
  CONSTRAINT fk_dates_season_id FOREIGN KEY (season_id)  
REFERENCES public.seasons(season_id)  
);
```

```
CREATE TABLE public.models (  
  model_id int4 NOT NULL,  
  model_name varchar(50) NULL,  
  manufacturer_id int4 NULL,  
  category_id int4 NULL,  
  CONSTRAINT models_pkey PRIMARY KEY (model_id),  
  CONSTRAINT fk_models_category_id FOREIGN KEY (category_id)  
REFERENCES public.category(category_id),
```

```
CONSTRAINT fk_models_manufacturer_id FOREIGN KEY  
(manufacturer_id) REFERENCES public.manufacturer(manufacturer_id);
```

```
CREATE TABLE public.sold_cars (
```

```
ad_id int4 NOT NULL,
```

```
body_type_id int4 NULL,
```

```
gearbox_type_id int4 NULL,
```

```
model_id int4 NULL,
```

```
city_id int4 NULL,
```

```
wheel_drive_id int4 NULL,
```

```
fuel_type_id int4 NULL,
```

```
date_added date NULL,
```

```
date_sold date NULL,
```

```
date_sold_id int4 NULL,
```

```
price float8 NULL,
```

```
currency varchar(5) NULL,
```

```
price_in_uah float8 NULL,
```

```
currency_to_uah_price float8 NULL,
```

```
year_manufactured int4 NULL,
```

```
range_km int4 NULL,
```

```
CONSTRAINT sold_cars_pkey PRIMARY KEY (ad_id),
```

```
CONSTRAINT fk_sold_cars_body_id FOREIGN KEY (body_type_id)
```

```
REFERENCES public.body_type(body_id),
```

```
CONSTRAINT fk_sold_cars_city_id FOREIGN KEY (city_id)
```

```
REFERENCES public.city(city_id),
```

```
CONSTRAINT fk_sold_cars_date_sold_id FOREIGN KEY  
(date_sold_id) REFERENCES public.dates(date_id),
```

```
CONSTRAINT fk_sold_cars_fuel_id FOREIGN KEY (fuel_type_id)  
REFERENCES public.fuel_type(fuel_id),
```

```
CONSTRAINT fk_sold_cars_gearbox_id FOREIGN KEY  
(gearbox_type_id) REFERENCES public.gearbox_type(gearbox_id),
```

```
CONSTRAINT fk_sold_cars_model_id FOREIGN KEY (model_id)  
REFERENCES public.models(model_id),
```

```
CONSTRAINT fk_sold_cars_wheel_drive_id FOREIGN KEY  
(wheel_drive_id) REFERENCES public.wheel_drive(wheel_drive_id);
```

НУБІП України

НУБІП України

SQL код внесення даних до бази даних

НУБІП України

НУБІП України

НУБІП України

НУБІП України

НУБІП України

```
insert into sold_cars (ad_id, body_type_id, gearbox_type_id, model_id, city_id,
wheel_drive_id, fuel_type_id, date_added, date_sold, date_sold_id, price, currency,
price_in_uah, currency_to_uah_price, year_manufactured, range_km) values ('31579079', 3, 1,
856, 269, 2, 1, '2021-11-23 19:11:02', '2021-11-27 20:06:02', '11654', 1650, 'USD', 71444,
26.96, 2002, 160000);
```

```
insert into sold_cars (ad_id, body_type_id, gearbox_type_id, model_id, city_id,
wheel_drive_id, fuel_type_id, date_added, date_sold, date_sold_id, price, currency,
price_in_uah, currency_to_uah_price, year_manufactured, range_km) values ('31545749', 3, 3,
535, 12, 2, 4, '2021-11-19 13:16:20', '2021-11-27 16:47:44', '11654', 4000, 'USD', 107840,
26.96, 2002, 216000);
```

```
insert into sold_cars (ad_id, body_type_id, gearbox_type_id, model_id, city_id,
wheel_drive_id, fuel_type_id, date_added, date_sold, date_sold_id, price, currency,
price_in_uah, currency_to_uah_price, year_manufactured, range_km) values ('31536614', 307,
1, 3011, 15, 1, 4, '2021-11-18 09:29:33', '2021-11-27 12:41:51', '11654', 4999, 'USD',
134773, 26.95999199839368, 2008, 240000);
```

```
insert into sold_cars (ad_id, body_type_id, gearbox_type_id, model_id, city_id,
wheel_drive_id, fuel_type_id, date_added, date_sold, date_sold_id, price, currency,
price_in_uah, currency_to_uah_price, year_manufactured, range_km) values ('31545785', 4, 1,
872, 10, 2, 1, '2021-11-19 13:16:35', '2021-11-27 14:25:14', '11654', 950, 'USD', 25612,
26.96, 2003, 170000);
```

```
insert into sold_cars (ad_id, body_type_id, gearbox_type_id, model_id, city_id,
wheel_drive_id, fuel_type_id, date_added, date_sold, date_sold_id, price, currency,
price_in_uah, currency_to_uah_price, year_manufactured, range_km) values ('31560620', 4, 1,
853, 303, 2, 1, '2021-11-21 15:10:12', '2021-11-27 08:23:01', '11654', 500, 'USD', 40440,
26.96, 1992, 130000);
```

```
insert into sold_cars (ad_id, body_type_id, gearbox_type_id, model_id, city_id,
wheel_drive_id, fuel_type_id, date_added, date_sold, date_sold_id, price, currency,
price_in_uah, currency_to_uah_price, year_manufactured, range_km) values ('31604962', 2, 1,
3008, 1, 2, 1, '2021-11-27 08:04:05', '2021-11-27 15:42:53', '11654', 5700, 'USD', 153672,
26.96, 2005, 201000);
```

```
insert into sold_cars (ad_id, body_type_id, gearbox_type_id, model_id, city_id,
wheel_drive_id, fuel_type_id, date_added, date_sold, date_sold_id, price, currency,
price_in_uah, currency_to_uah_price, year_manufactured, range_km) values ('31457463', 8, 1,
2036, 23, 2, 2, '2021-11-07 17:30:15', '2021-11-26 23:43:50', '11654', 6600, 'USD', 17.936,
26.96, 2003, 105000);
```

```
insert into sold_cars (ad_id, body_type_id, gearbox_type_id, model_id, city_id,
wheel_drive_id, fuel_type_id, date_added, date_sold, date_sold_id, price, currency,
price_in_uah, currency_to_uah_price, year_manufactured, range_km) values ('31516369', 4, 1,
30786, 10, 2, 4, '2021-11-15 15:26:26', '2021-11-27 15:02:17', '11654', 3900, 'USD', 105144,
26.96, 1998, 400000);
```

```
insert into sold_cars (ad_id, body_type_id, gearbox_type_id, model_id, city_id,
wheel_drive_id, fuel_type_id, date_added, date_sold, date_sold_id, price, currency,
price_in_uah, currency_to_uah_price, year_manufactured, range_km) values ('31567635', 4, 1,
2990, 484, 2, 1, '2021-11-22 13:00:15', '2021-11-27 12:00:16', '11654', 6150, 'USD', 165804,
26.96, 2007, 180000);
```

```
insert into sold_cars (ad_id, body_type_id, gearbox_type_id, model_id, city_id,
wheel_drive_id, fuel_type_id, date_added, date_sold, date_sold_id, price, currency,
price_in_uah, currency_to_uah_price, year_manufactured, range_km) values ('31025814', 56,
1, 3354, 10, 4, 0, '2021-11-14 08:20:10', '2021-11-27 19:54:41', '11654', 8150, 'USD',
214724, 26.96, 2013, 0);
```

```
insert into sold_cars (ad_id, body_type_id, gearbox_type_id, model_id, city_id,
wheel_drive_id, fuel_type_id, date_added, date_sold, date_sold_id, price, currency,
price_in_uah, currency_to_uah_price, year_manufactured, range_km) values ('31198247', 9, 2,
31575, 9, 1, 1, '2021-11-22 10:08:57', '2021-11-27 16:51:36', '11654', 35900, 'USD', 967864,
26.96, 2016, 183000);
```

```
insert into sold_cars (ad_id, body_type_id, gearbox_type_id, model_id, city_id,
wheel_drive_id, fuel_type_id, date_added, date_sold, date_sold_id, price, currency,
price_in_uah, currency_to_uah_price, year_manufactured, range_km) values ('31558251', 3, 1,
43, 538, 2, 0, '2021-11-21 11:04:31', '2021-11-27 13:43:08', '11654', 1200, 'USD', 32352,
26.96, 1985, 100000);
```

```
insert into sold_cars (ad_id, body_type_id, gearbox_type_id, model_id, city_id,
wheel_drive_id, fuel_type_id, date_added, date_sold, date_sold_id, price, currency,
price_in_uah, currency_to_uah_price, year_manufactured, range_km) values ('31606276', 307,
1, 3011, 10, 1, 4, '2021-11-27 12:04:07', '2021-11-27 17:20:31', '11654', 9000, 'USD',
242640, 26.96, 2012, 147000);
```

```
insert into sold_cars (ad_id, body_type_id, gearbox_type_id, model_id, city_id,
wheel_drive_id, fuel_type_id, date_added, date_sold, date_sold_id, price, currency,
```



```
price_in_uah, currency_to_uah_price, year_manufactured, range_km) values ('31046213', 5, 2,
2630, 133, 2, 4, '2021-11-14 18:19:19', '2021-11-27 12:49:59', '11654', 9500, 'USD', 256120,
26.96, 2013, 145000 );
insert into sold_cars (ad_id, body_type_id, gearbox_type_id, model_id, city_id,
wheel_drive_id, fuel_type_id, date_added, date_sold, date_sold_id, price, currency,
price_in_uah, currency_to_uah_price, year_manufactured, range_km) values ('30647927', 254,
1, 584, 70, 2, 2, '2021-11-10 10:43:52', '2021-11-27 19:11:22', '11654', 9800, 'USD',
264208, 26.96, 2014, 184000 );
insert into sold_cars (ad_id, body_type_id, gearbox_type_id, model_id, city_id,
wheel_drive_id, fuel_type_id, date_added, date_sold, date_sold_id, price, currency,
price_in_uah, currency_to_uah_price, year_manufactured, range_km) values ('31561366', 8, 1,
37785, 5, 2, 1, '2021-11-21 16:16:05', '2021-11-27 16:33:26', '11654', 7850, 'USD', 211636,
26.96, 2010, 212000 );
insert into sold_cars (ad_id, body_type_id, gearbox_type_id, model_id, city_id,
wheel_drive_id, fuel_type_id, date_added, date_sold, date_sold_id, price, currency,
price_in_uah, currency_to_uah_price, year_manufactured, range_km) values ('31593846', 2, 1,
586, 12, 2, 2, '2021-11-25 18:18:58', '2021-11-27 17:26:48', '11654', 6800, 'USD', 183328,
26.96, 2010, 217000 );
insert into sold_cars (ad_id, body_type_id, gearbox_type_id, model_id, city_id,
wheel_drive_id, fuel_type_id, date_added, date_sold, date_sold_id, price, currency,
price_in_uah, currency_to_uah_price, year_manufactured, range_km) values ('31451410', 5, 2,
167, 213, 1, 4, '2021-11-06 20:26:06', '2021-11-27 20:19:44', '11654', 8250, 'USD', 222420,
26.96, 2006, 196000 );
insert into sold_cars (ad_id, body_type_id, gearbox_type_id, model_id, city_id,
wheel_drive_id, fuel_type_id, date_added, date_sold, date_sold_id, price, currency,
price_in_uah, currency_to_uah_price, year_manufactured, range_km) values ('31442207', 2, 2, 38063, 1, 2, 2,
'2021-11-05 15:55:33', '2021-11-27 14:04:40', '11654', 11950, 'USD', 322172, 26.96, 2012, 190000 );
insert into sold_cars (ad_id, body_type_id, gearbox_type_id, model_id, city_id,
wheel_drive_id, fuel_type_id, date_added, date_sold, date_sold_id, price, currency,
price_in_uah, currency_to_uah_price, year_manufactured, range_km) values ('31441682', 195,
1, 31254, 18, 2, 2, '2021-11-05 14:55:06', '2021-11-27 17:25:58', '11654', 15555, 'USD',
419363, 26.96, 2012, 292000 );
insert into sold_cars (ad_id, body_type_id, gearbox_type_id, model_id, city_id,
wheel_drive_id, fuel_type_id, date_added, date_sold, date_sold_id, price, currency,
price_in_uah, currency_to_uah_price, year_manufactured, range_km) values ('21514519', 4, 1,
239, 210, 2, 2, '2021-11-19 17:10:07', '2021-11-27 17:45:02', '11654', 7500, 'USD', 202200,
26.96, 2012, 120000 );
insert into sold_cars (ad_id, body_type_id, gearbox_type_id, model_id, city_id,
wheel_drive_id, fuel_type_id, date_added, date_sold, date_sold_id, price, currency,
price_in_uah, currency_to_uah_price, year_manufactured, range_km) values ('31526882', 8, 1,
588, 4, 2, 2, '2021-11-16 20:40:25', '2021-11-27 13:46:13', '11654', 9500, 'USD', 256120,
26.96, 2012, 186000 );
insert into sold_cars (ad_id, body_type_id, gearbox_type_id, model_id, city_id,
wheel_drive_id, fuel_type_id, date_added, date_sold, date_sold_id, price, currency,
price_in_uah, currency_to_uah_price, year_manufactured, range_km) values ('31604629', 8, 1,
1524, 18, 2, 1, '2021-11-27 00:23:11', '2021-11-27 17:06:12', '11654', 4050, 'USD', 109188,
26.96, 2001, 190000 );
insert into sold_cars (ad_id, body_type_id, gearbox_type_id, model_id, city_id,
wheel_drive_id, fuel_type_id, date_added, date_sold, date_sold_id, price, currency,
price_in_uah, currency_to_uah_price, year_manufactured, range_km) values ('30938835', 5, 2,
2544, 4, 2, 1, '2021-11-25 18:42:12', '2021-11-27 05:50:10', '11654', 11000, 'USD', 296550,
26.96, 2014, 167000 );
insert into sold_cars (ad_id, body_type_id, gearbox_type_id, model_id, city_id,
wheel_drive_id, fuel_type_id, date_added, date_sold, date_sold_id, price, currency,
price_in_uah, currency_to_uah_price, year_manufactured, range_km) values ('31592555', 3, 1,
1060, 7, 2, 4, '2021-11-25 16:28:10', '2021-11-27 13:22:08', '11654', 3950, 'USD', 106492,
26.96, 2004, 153000 );
insert into sold_cars (ad_id, body_type_id, gearbox_type_id, model_id, city_id,
wheel_drive_id, fuel_type_id, date_added, date_sold, date_sold_id, price, currency,
price_in_uah, currency_to_uah_price, year_manufactured, range_km) values ('31091884', 5, 2,
824, 10, 1, 2, '2021-11-22 15:17:50', '2021-11-27 17:06:39', '11654', 22500, 'USD', 606600,
26.96, 2013, 180000 );
insert into sold_cars (ad_id, body_type_id, gearbox_type_id, model_id, city_id,
wheel_drive_id, fuel_type_id, date_added, date_sold, date_sold_id, price, currency,
```



```
price_in_uah, currency_to_uah, price, year_manufactured, range_km) values ('31026679', 8, 1, 3153, 7, 2, 4, '2021-11-13 15:55:48', '2021-11-27 10:30:33', '11654', 5600, 'USD', 150976, 26.96, 2007, 261000 );
```

```
insert into sold_cars (ad_id, body_type_id, gearbox_type_id, model_id, city_id, wheel_drive_id, fuel_type_id, date_added, date_sold, date_sold_id, price, currency, price_in_uah, currency_to_uah, price, year_manufactured, range_km) values ('31461155', 5, 1, 30503, 15, 2, 2, '2021-11-08 10:06:42', '2021-11-27 09:41:37', '11654', 17500, 'USD', 471800, 26.96, 2019, 31000 );
```

```
insert into sold_cars (ad_id, body_type_id, gearbox_type_id, model_id, city_id, wheel_drive_id, fuel_type_id, date_added, date_sold, date_sold_id, price, currency, price_in_uah, currency_to_uah, price, year_manufactured, range_km) values ('31496071', 4, 1, 3008, 10, 2, 1, '2021-11-12 21:12:10', '2021-11-27 18:48:40', '11654', 4493, 'USD', 121293, 26.95999110915363, 2003, 240000 );
```

```
insert into sold_cars (ad_id, body_type_id, gearbox_type_id, model_id, city_id, wheel_drive_id, fuel_type_id, date_added, date_sold, date_sold_id, price, currency, price_in_uah, currency_to_uah, price, year_manufactured, range_km) values ('30816564', 4, 1, 1556, 16, 2, 1, '2021-11-16 12:02:08', '2021-11-27 15:32:05', '11654', 4500, 'USD', 121320, 26.96, 2007, 10000 );
```

```
insert into sold_cars (ad_id, body_type_id, gearbox_type_id, model_id, city_id, wheel_drive_id, fuel_type_id, date_added, date_sold, date_sold_id, price, currency, price_in_uah, currency_to_uah, price, year_manufactured, range_km) values ('31006433', 2, 2, 1524, 35, 2, 1, '2021-11-27 12:26:09', '2021-11-27 19:38:46', '11654', 8300, 'USD', 22768, 26.96, 2009, 120000 );
```

```
insert into sold_cars (ad_id, body_type_id, gearbox_type_id, model_id, city_id, wheel_drive_id, fuel_type_id, date_added, date_sold, date_sold_id, price, currency, price_in_uah, currency_to_uah, price, year_manufactured, range_km) values ('31599365', 307, 1, 2087, 76, 2, 4, '2021-11-26 14:01:34', '2021-11-26 22:21:00', '11653', 3000, 'USD', 80880, 26.96, 2008, 90000 );
```

```
insert into sold_cars (ad_id, body_type_id, gearbox_type_id, model_id, city_id, wheel_drive_id, fuel_type_id, date_added, date_sold, date_sold_id, price, currency, price_in_uah, currency_to_uah, price, year_manufactured, range_km) values ('31564388', 4, 2, 2990, 20, 2, 1, '2021-11-21 21:44:09', '2021-11-27 19:48:22', '11654', 8200, 'USD', 221072, 26.96, 2008, 144000 );
```

```
insert into sold_cars (ad_id, body_type_id, gearbox_type_id, model_id, city_id, wheel_drive_id, fuel_type_id, date_added, date_sold, date_sold_id, price, currency, price_in_uah, currency_to_uah, price, year_manufactured, range_km) values ('31503385', 5, 5, 2228, 15, 1, 1, '2021-11-13 21:45:43', '2021-11-27 14:28:34', '11654', 16499, 'USD', 444813, 26.959997575610643, 2017, 127000 );
```

```
ah_price, year_manufactured, range_km) values ('31442066', 17, 1, 55521, 5, 6, 1, '2021-11-05 15:39:10', '2021-11-27 09:59:26', '11654', 3100, 'USD', 83576, 26.96, 2000, 0 );
```

```
insert into sold_cars (ad_id, body_type_id, gearbox_type_id, model_id, city_id, wheel_drive_id, fuel_type_id, date_added, date_sold, date_sold_id, price, currency, price_in_uah, currency_to_uah, price, year_manufactured, range_km) values ('31579616', 4, 1, 853, 527, 2, 1, '2021-11-23 20:26:08', '2021-11-27 20:29:21', '11654', 1300, 'USD', 35048, 26.96, 1990, 100000 );
```

НУБІП України

НУБІП України