МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

ВСП Технічний фаховий коледж

НАЦІОНАЛЬНОГО УНІВЕРСИТЕТУ " ЛЬВІВСЬКА ПОЛІТЕХНІКА"

Відділення " Інформаційних технологій та комп’ютерної техніки"

**КУРСОВА РОБОТА**

з дисципліни "Бази даних"

на тему:

«Проектування та застосування бази даних інформаційної системи

автомайстерні»

Виконав студент групи ПЗ 42 Боднарчук Віктор,

\_\_\_\_\_\_\_\_\_\_\_10001 № залікової книжки

|  |  |  |
| --- | --- | --- |
| Оцінка | Балів | Дата |
|  |  |  |

Керівник роботи Заяць М.М

Львів 2023

ЗАВДАННЯ

на курсову роботу з дисципліни "Бази даних"

студента групи ПЗ 42 Боднарчука Віктора

**Тема: «Проектування бази даних інформаційної системи**

**автомайстерні»**

**Завдання**: автоматизувати процес формування та опрацювання телефонного довідника.

**Зміст завдання та календарний план його виконання**

|  |  |  |
| --- | --- | --- |
| **№ з/п** | **Зміст завдання** | **Дата** |
| 1 | Здійснити аналітичний огляд літератури за заданою темою та обґрунтувати вибір інструментальних засобів проектування. | до 30.10. |
| 2 | Визначення та опис предметної області | до 05.11. |
| 3 | Побудова моделі «сутність-зв‘язок» | до 10.11. |
| 4 | Побудова логічної структури бази даних | до 15.11. |
| 5 | Побудова обмежень відношень бази даних. Нормалізація відношень бази даних. | до 20.11. |
| 6 | Виконання над відношеннями операцій реляційної алгебри. | до 30.11 |
| 7 | Оформити записку до курсової роботи згідно з вимогами Міжнародних стандартів, дотримуючись такого змісту:   * вступ; * теоретична частина; * визначення та опис предметної області; * концептуальна модель «сутність – зв‘язок»; * логічна структура бази даних; * нормалізація бази даних; * виконання операцій реляційної алгебри; * висновки ; * список використаних джерел; * додатки. | до 08.12 |

Завдання прийнято до виконання: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ (Боднарчук В.Р)

( підпис студента)

Керівник роботи : \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_/ Заяць М.М/

**Зміст**

**Вступ – 4**

**Теоретична частина – 5**

**Визначення та опис предметної області – 6**

**Концептуальна модель типу «Сутність-зв’язок» - 7**

**Логічна структура бази даних – 9**

**Виконання операцій реляційної алгебри – 14**

**Висновок – 21**

**Список використаних джерел – 22**

**Додатки - 23**

**Вступ**

Сучасна технологічна динаміка у сфері автосервісу потребує ефективних та точних інструментів управління та обліку. Відсутність централізованої системи контролю та управління в автомайстернях часто призводить до розпаду інформації та втрати потенційних можливостей для покращення сервісу.

Проблема в управлінні інформацією в автомайстернях залишається актуальною внаслідок великого обсягу даних, що потребують структурованого зберігання та систематизації. Необхідно розв’язати завдання пов'язані з оптимізацією процесів обслуговування, контролем запасів, обліком послуг та ремонтів, щоб підвищити ефективність роботи автомайстерень.

Для вирішення цих проблем обрано підхід розробки бази даних у середовищі SQL з використанням програмних засобів, що забезпечують надійність та гнучкість при роботі з даними. Використання інформаційної системи дозволить автомайстерням ефективно управляти процесами, забезпечуючи високу якість обслуговування автомобілів та підвищення рівня задоволеності клієнтів.

Для розробки проекту буде використана актуальна технологія, яка забезпечує потужні можливості управління даними та має великий потенціал для майбутнього розширення системи.

**Мета**

Мета даного дослідження полягає у розробці та реалізації бази даних для інформаційної системи автомайстерні, спрямованої на автоматизацію та оптимізацію управління процесами обслуговування автомобілів. Основною метою є створення ефективної та надійної системи для зберігання, організації та аналізу інформації, необхідної для оптимального функціонування автомайстерні.

**Актуальність**

Завдяки стрімкому розвитку технологій та зростанню конкуренції на ринку автосервісу, потреба в ефективному управлінні та обробці великого обсягу даних набуває великої актуальності. Недоліки у системі управління даними в автомайстернях часто призводять до затримок у виконанні робіт, помилок у обліку запасів та неефективного використання ресурсів

**Цілі**

* Оптимізація обслуговування: Створення системи, що дозволить ефективно контролювати процеси обслуговування автомобілів та покращити їх швидкість та якість.
* Ефективне управління даними: Розробка бази даних, що забезпечить надійне зберігання та швидкий доступ до інформації про клієнтів, ремонти, запаси та інші аспекти автомайстерні.
* Підвищення клієнтської задоволеності: Забезпечення оперативного обслуговування, точності обліку та якості послуг для збільшення задоволеності клієнтів.

Розробка та впровадження інформаційної системи автомайстерні допоможе вирішити ці завдання та досягти поставлених цілей для підвищення ефективності та конкурентоспроможності у сфері автосервісу.

**Теоретична частина**

**Архітектура баз даних CODASYL та ANSI/X3/SPARC**

Архітектура баз даних є фундаментальним поняттям у сфері обробки даних. Два основних підходи до організації баз даних - CODASYL та ANSI/X3/SPARC - надають важливі принципи та моделі для створення ефективних інформаційних систем.

Архітектура баз даних CODASYL

Конференція Державного асоційованого комітету з систем інформаційних мов (CODASYL) зіграла велику роль у створенні стандартів організації та взаємодії з базами даних. Модель CODASYL базується на ієрархічній структурі та використовує "записи", "сегменти" та "відносини" для організації даних. Цей підхід використовується для реляційних баз даних та баз даних типу NoSQL.

Архітектура баз даних ANSI/X3/SPARC

Модель, розроблена Спеціальним комітетом з архітектури та стандартів програмного забезпечення (ANSI/X3/SPARC), визначила трьохрівневу архітектуру бази даних: зовнішній рівень, концептуальний рівень та внутрішній рівень. Це поділ дозволяє відокремити структуру даних від способу їх збереження та використання.

**CODASYL:**

CODASYL (Conference on Data Systems Languages) визначив стандарти для моделі роботи з базами даних в розрізі даних, впроваджуючи підхід ієрархічної організації. Його найвідомішою реалізацією є мова програмування COBOL та мови доступу до даних як ALGOL та PL/I. Цей підхід базувався на структурі типу дерева, де записи організовані у вигляді вузлів, а сегменти та відносини представлені як вкладені структури даних.

Принципова особливість CODASYL полягала в підтримці різних типів відношень між записами, таких як один до багатьох та багато до багатьох. Ця модель дала поштовх розвитку інших структур баз даних, зокрема реляційних баз даних.

**ANSI/X3/SPARC:**

ANSI/X3/SPARC (American National Standards Institute/Standards Planning and Requirements Committee) розробив теоретичну концепцію для організації баз даних, яка мала велике вплив на створення стандартів та моделей баз даних. Однією з найважливіших концепцій є трьохрівнева архітектура бази даних.

Внутрішній рівень: Відображає спосіб фізичного зберігання даних.

Концептуальний рівень: Визначає структуру та відношення між даними у всій базі даних, але не залежить від конкретного інтерфейсу.

Зовнішній рівень: Визначає, як користувачі бачать дані. Це відокремлення дозволяє розділити спосіб представлення даних від фізичного збереження, що полегшує роботу з базою даних для різних користувачів.

Ці два підходи стали основою для багатьох моделей та стандартів у сфері баз даних. Вони надали ключові концепції, які використовуються у сучасних базах даних та є основою для подальших досліджень у цій області.

# **Визначення та опис предметної області**

Обраною предметною областю є автомайстерня – місце, де здійснюється ремонт автомобілів та надання пов'язаних послуг. Автомайстерня надає широкий спектр послуг, включаючи технічне обслуговування, діагностику, ремонт та заміну деталей автомобіля, а також консультації та інші сервіси, пов'язані з автомобільною сферою.

Основна діяльність автомайстерень включає у себе обслуговування автотранспортних засобів різних марок і моделей. Це вимагає систематичного та якісного обліку даних про автомобілі, клієнтів, виконані послуги, використані деталі та інші аспекти.

Робота автомайстерні базується на налагодженій системі робочих процесів та управлінських рішень. Це включає в себе ведення журналу відвідувань, облік та контроль робіт, запасів деталей, планування графіку обслуговування, управління персоналом та взаємодію з клієнтами.

Запланована база даних для автомайстерні повинна містити інформацію про різні моделі автомобілів, їх технічні характеристики, історію обслуговування та ремонту. Також, система повинна включати відомості про клієнтів, контактні дані, історію відвідувань. Додатково, важливо вести облік використаних запчастин, виконаних робіт та наданих послуг, щоб забезпечити точну фінансову звітність та підтримати аналітику ефективності.

Ця комплексна база даних дозволить автомайстерням вести точний облік робіт, покращувати якість обслуговування автотранспорту та забезпечити базу для прийняття стратегічних рішень управління.

**Концептуальна модель типу «Сутність-зв’язок»**

**Модель «сутність-зв'язок» (ER-модель)**  — модель даних, яка дозволяє описувати концептуальні схеми за допомогою узагальнених конструкцій блоків. ER-модель — це мета-модель даних, тобто засіб опису моделей даних. Існує ряд моделей для представлення знань, але одним з найзручніших інструментів уніфікованого представлення даних, незалежного від програмного забезпечення, що його реалізує, є модель «сутність-зв'язок».

Концептуальна модель «сутність – зв'язок» для автомайстерні передбачає визначення ключових сутностей та їх атрибутів, а також зв'язків між цими сутностями.

Сутності:

* Власник
* Автомобіль
* Відомість про ремонт
* Майстер
* Каталог запчастин
* Відомість використаних запчастин

**Сутності та їх атрибути:**

1. Власник:
   * Код\_Власника (INT, PRIMARY KEY)
   * ПІБ (VARCHAR(100))
   * номер\_телефону (VARCHAR(20))
   * номер\_паспорту (VARCHAR(20))
2. Автомобіль:
   * win (VARCHAR(50), PRIMARY KEY)
   * марка (VARCHAR(50))
   * номер (VARCHAR(20))
   * модель (VARCHAR(50))
   * рік\_випуску (INT)
   * код\_власника (INT, FOREIGN KEY посилається на Власник(Код\_Власника))
3. Майстер:
   * Код\_майстра (INT, PRIMARY KEY)
   * ПІБ (VARCHAR(100))
   * номер\_телефону (VARCHAR(20))
   * номер\_паспорту (VARCHAR(20))
   * спеціалізація (VARCHAR(100))
4. Каталог\_запчастин:
   * код\_запчастини (VARCHAR(50), PRIMARY KEY)
   * виробник (VARCHAR(100))
   * назва (VARCHAR(100))
   * тип (VARCHAR(50))
   * додаткове\_поле (VARCHAR(255))
5. Відомість\_наявних\_запчастин:
   * код\_запису (INT, PRIMARY KEY)
   * кількість (INT)
   * дата\_надходження (DATE)
   * дата\_виготовлення (DATE)
   * код\_запчастини (VARCHAR(50), FOREIGN KEY посилається на Каталог\_запчастин(код\_запчастини))
   * ціна (INT)
6. Відомість\_про\_ремонт:
   * код\_запису (INT, PRIMARY KEY)
   * дата (DATE)
   * win (VARCHAR(50), FOREIGN KEY посилається на Автомобіль(win))
   * код\_майстра (INT, FOREIGN KEY посилається на Майстер(Код\_майстра))
   * проблема (VARCHAR(255))
   * опис\_ремонту (TEXT)
   * ціна (DECIMAL(10, 2))
   * пробіг\_на\_момент\_ремонту (INT)
7. Відомість\_використання\_запчастин:
   * код\_запису (INT, PRIMARY KEY)
   * код\_ремонту (INT, FOREIGN KEY посилається на Відомість\_про\_ремонт(код\_запису))
   * код\_запчастини (VARCHAR(50), FOREIGN KEY посилається на Каталог\_запчастин(код\_запчастини))

Зв'язки між сутностями:

Автомобіль належить лише одному власнику, але власник може мати багато автомобілів (зв'язок "один-до-багатьох").

Кожен ремонт пов'язаний з одним автомобілем, але автомобіль може мати багато ремонтів (зв'язок "один-до-багатьох").

Кожен ремонт пов'язаний з одним майстром, і майстер може мати багато ремонтів (зв'язок "один-до-багатьох").

Відомість про ремонт використовує певні запчастини, і кожна запчастина може бути використана в багатьох ремонтах (зв'язок "багато-до-багатьох").

Ця модель дозволяє логічно відобразити співвідношення між сутностями та їх атрибутами, а також зв'язками між ними. Це стане основою для подальшої розробки бази даних, що спрямована на збереження та ефективне управління інформацією в автомайстерні.

**Логічна структура бази даних**

Розглянемо логічну структуру бази даних для інформаційної системи "Автомайстерня":

* Власник: Зберігає дані про власників автомобілів, такі як ПІБ, номер телефону, номер паспорту.
* Автомобіль: Містить дані про автомобілі, включаючи VIN-код, марку, номер, модель, рік випуску та зв'язок з власником.
* Майстер: Тут зберігаються дані про майстрів: ПІБ, номер телефону, номер паспорту, спеціалізація.
* Каталог\_запчастин: Містить інформацію про запчастини: код запчастини, виробника, назву, тип, додаткові характеристики.
* Відомість\_наявних\_запчастин: Зберігає дані про наявність запчастин: кількість, дату надходження, дату виготовлення, ціну та зв'язок з каталогом запчастин.
* Відомість\_про\_ремонт: Містить інформацію про ремонти: дату, зв'язок з автомобілем, майстром, опис проблеми, опис ремонту, ціну, пробіг на момент ремонту.
* Відомість\_використання\_запчастин: Тут зберігається інформація про використання запчастин в ремонтах: зв'язок з ремонтом та запчастиною.

Це загальна логічна структура бази даних, яка включає в себе таблиці для зберігання інформації про власників, автомобілі, майстрів, запчастини, наявність запчастин, ремонти та використання запчастин у ремонтах. Кожна таблиця має поля, що відповідають конкретним характеристикам об'єктів цієї сутності та зв'язки між ними для забезпечення взаємозв'язку даних у системі.

**Структура даних**

Таблиця Власник:

|  |  |  |
| --- | --- | --- |
| **Назва** | **Тип поля** | **Розмірність та додаткові обмеження** |
| Код\_власника | INT | PRIMARY KEY |
| ПІБ | VARCHAR | 100 |
| Номер\_телефону | VARCHAR | 20 |
| Номер\_паспорту | VARCHAR | 20 |

Таблиця Автомобіль:

|  |  |  |
| --- | --- | --- |
| **Назва** | **Тип поля** | **Розмірність та додаткові обмеження** |
| win | VARCHAR | 50, PRIMARY KEY |
| марка | VARCHAR | 50 |
| номер | VARCHAR | 20 |
| модель | VARCHAR | 50 |
| Рік\_випуску | INT |  |
| Код\_власника | INT | Зовнішній ключ |

Таблиця Майстер:

|  |  |  |
| --- | --- | --- |
| **Назва** | **Тип поля** | **Розмірність та додаткові обмеження** |
| Код\_майстра | INT | PRIMATY KEY |
| ПІБ | VARCHAR | 100 |
| Номер\_телефону | VARCHAR | 20 |
| Номер\_паспорту | VARCHAR | 20 |
| Спеціалізація | VARCHAR | 100 |

Таблиця Каталог\_запчастин:

|  |  |  |
| --- | --- | --- |
| **Назва** | **Тип поля** | **Розмірність та додаткові обмеження** |
| Код\_запчастини | VARCHAR | 50, PRIMARY KEY |
| виробник | VARCHAR | 100 |
| назва | VARCHAR | 100 |
| тип | VARCHAR | 50 |
| Додаткове\_поле | VARCHAR | 255 |

Таблиця Відомість наявних запчастин:

|  |  |  |
| --- | --- | --- |
| **Назва** | **Тип поля** | **Розмірність та додаткові обмеження** |
| Код\_запису | INT | PRIMATY KEY |
| Кількість | INT |  |
| Дата\_виготовлення | DATE | Дата |
| Дата\_надходження | DATE | Дата |
| Код\_запчастини | VARCHAR | 50, зовнішній ключ |
| Ціна | INT |  |

Таблиця Відомість\_використання\_запчастин:

|  |  |  |
| --- | --- | --- |
| **Назва** | **Тип поля** | **Розмірність та додаткові обмеження** |
| Код запису | INT | PRIMARY KEY |
| Код\_ремонту | INT | Зовнішній ключ |
| Код\_запчастини | VARCHAR | 50, Зовнішній ключ |

Створення таблиць з використанням SQL:

CREATE TABLE Власник (

Код\_Власника INT PRIMARY KEY,

ПІБ VARCHAR(100),

номер\_телефону VARCHAR(20),

номер\_паспорту VARCHAR(20)

);

describe Власник;

CREATE TABLE Автомобіль (

win VARCHAR(50) PRIMARY KEY,

марка VARCHAR(50),

номер VARCHAR(20),

модель VARCHAR(50),

рік\_випуску INT,

код\_власника INT,

FOREIGN KEY (код\_власника) REFERENCES Власник(Код\_Власника)

);

describe Автомобіль;

CREATE TABLE Майстер (

Код\_майстра INT PRIMARY KEY,

ПІБ VARCHAR(100),

номер\_телефону VARCHAR(20),

номер\_паспорту VARCHAR(20),

спеціалізація VARCHAR(100)

);

describe Майстер;

CREATE TABLE Каталог\_запчастин (

код\_запчастини VARCHAR(50) PRIMARY KEY,

виробник VARCHAR(100),

назва VARCHAR(100),

тип VARCHAR(50),

додаткове\_поле VARCHAR(255)

);

describe Каталог\_запчастин;

CREATE TABLE Відомість\_наявних\_запчастин (

код\_запису INT PRIMARY KEY,

кількість INT,

дата\_надходження DATE,

дата\_виготовлення DATE,

код\_запчастини VARCHAR(50),

ціна INT,

FOREIGN KEY (код\_запчастини) REFERENCES Каталог\_запчастин(код\_запчастини)

);

describe Відомість\_наявних\_запчастин;

CREATE TABLE Відомість\_про\_ремонт (

код\_запису INT PRIMARY KEY,

дата DATE,

win VARCHAR(50),

код\_майстра INT,

проблема VARCHAR(255),

опис\_ремонту TEXT,

ціна DECIMAL(10, 2),

пробіг\_на\_момент\_ремонту INT,

FOREIGN KEY (win) REFERENCES Автомобіль(win),

FOREIGN KEY (код\_майстра) REFERENCES Майстер(Код\_майстра)

);

describe Відомість\_про\_ремонт;

CREATE TABLE Відомість\_використання\_запчастин (

код\_запису INT PRIMARY KEY,

код\_ремонту INT,

код\_запчастини VARCHAR(50),

FOREIGN KEY (код\_ремонту) REFERENCES Відомість\_про\_ремонт(код\_запису),

FOREIGN KEY (код\_запчастини) REFERENCES Каталог\_запчастин(код\_запчастини)

);

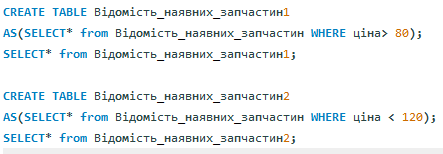
describe Відомість\_використання\_запчастин;

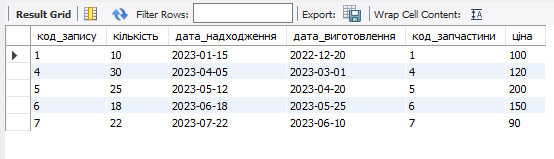
**Виконання операцій реляційної алгебри**

У цьому розділі будуть виконані операції над відношеннями реляційної алгебри:

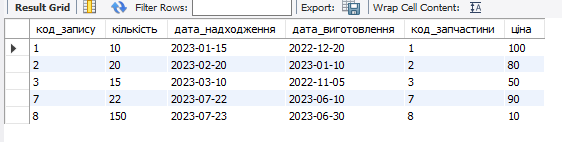
* перетин, об‘єднання, різниця відношення з його копією;
* декартів добуток відношень;
* селекція, проекція одного відношення;
* натуральне та умовне з‘єднання двох відношень;
* включення, вилучення кортежів, зміна значень атрибутів у одному з відношень;
* додавання нового атрибута, вилучення атрибута, зміна параметрів атрибута у одному з відношень.

Створимо 2 додаткових таблиці, над якими будемо виконувати деякі операції реляційної алгебри:



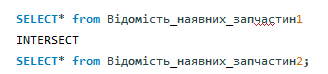


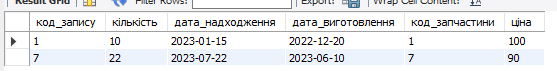
Таблиця відомість\_наявних\_запчастин1



Таблиця Відомість\_наявних\_запчастин2

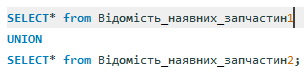
**Запит на перетин таблиць**

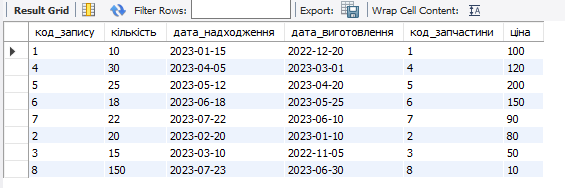
****

****

Даний запит виводить спільні дані з 2-х таблиць.

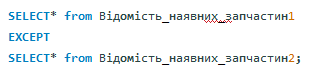
**Запит на об’єднання таблиць**

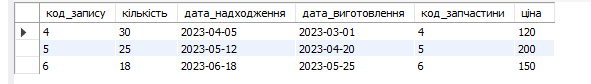




Результат виконання об’єднання таблиць

**Запит на різницю відношення таблиць**

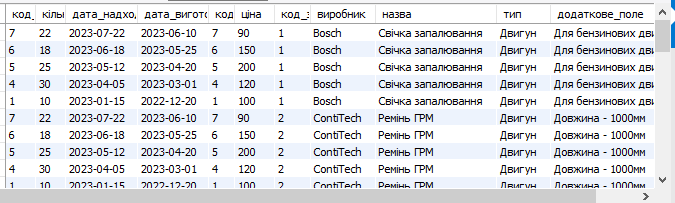
****

****

Результат виконання різниці відношень.

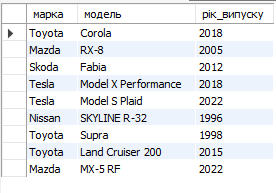
**Декартів добуток**





**Проекція відношення**

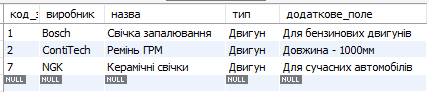




Результат вибору з таблиці Автомобіль стовпців марка, модель, рік\_випуску

**Селекція відношення**

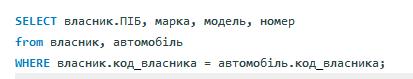
****



Результат вибору з таблиці каталог\_запчастин полів, де тип – двигун

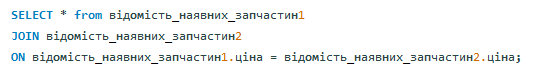
**Натуральне з’єднання відношень**

Виведемо власників автомобілів та їхні автомобілі.





**Умовне з’єднання відношень**

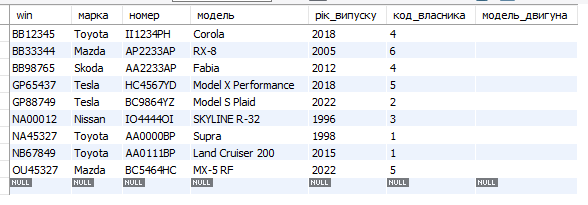




**Додавання нового атрибуту**

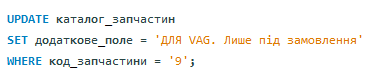
Добавимо до таблиці Автомобіль атрибут модель\_двигуна





**Зміна параметрів атрибута**

Змінимо додаткове\_поле в таблиці каталог\_запчастин

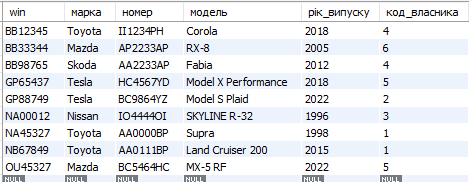




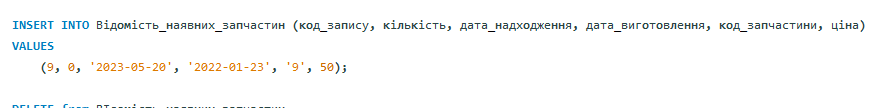
**Вилучення атрибута**

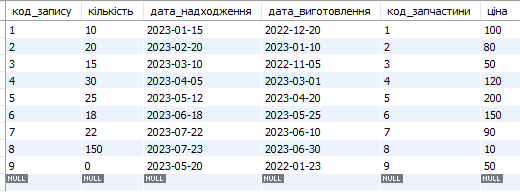
Вилучимо атрибут модель\_двигуна з таблиці Автомобіль





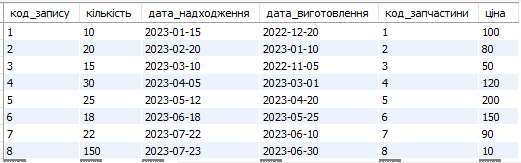
**Включення кортежів**





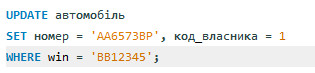
**Виключення кортежів**

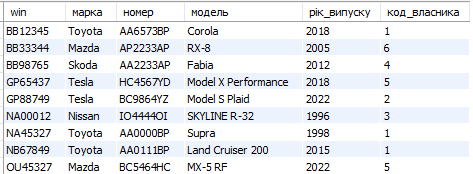




**Зміна значень атрибута**

В таблиці автомобіль змінимо власника та номер





**Висновок**

Створення бази даних для автоматизованої системи управління автомайстернею є ключовим кроком у поліпшенні ефективності, організації та точності управління бізнесом. Розробка бази даних для цієї сфери включає в себе комплексне моделювання процесів обслуговування та управління даними про клієнтів, транспортні засоби, запчастини, робочий персонал та інші аспекти, що охоплюють роботу автомайстерні.

Одним із ключових джерел при розробці цієї бази даних є актуальна інформація від власників та менеджерів автомайстерень. Також, під час розробки використовуються відомі стандарти управління та обліку запчастин, сервісних робіт, а також специфічні особливості цієї сфери бізнесу.

Щоб забезпечити оптимальну ефективність та коректність бази даних, важливо врахувати специфічні потреби автомайстерень, уникати зайвого дублювання даних та забезпечити достатню захищеність інформації, що зберігається в базі.

Крім того, при розробці бази даних для автомайстерень, важливо врахувати можливість масштабування та подальшого розвитку системи, оскільки вона повинна відповідати зростаючим потребам та можливостям бізнесу.

**Список використаних джерел**

Smith, J. "Effective Database Management in Automotive Repair Shops." Journal of Business Technology, 2020.

Johnson, K. "Optimizing Repair Shop Operations Through Database Solutions." International Conference on Business Information Systems, 2019..

Date, C.J. (2003). An Introduction to Database Systems (8th Edition). Addison Wesley.

Elmasri, Ramez and Navathe, Shamkant B. (2016). Fundamentals of Database Systems (7th Edition). Pearson.

<https://uk.wikipedia.org/wiki/%D0%9C%D0%BE%D0%B4%D0%B5%D0%BB%D1%8C_%C2%AB%D1%81%D1%83%D1%82%D0%BD%D1%96%D1%81%D1%82%D1%8C_%E2%80%94_%D0%B7%D0%B2%27%D1%8F%D0%B7%D0%BE%D0%BA%C2%BB>

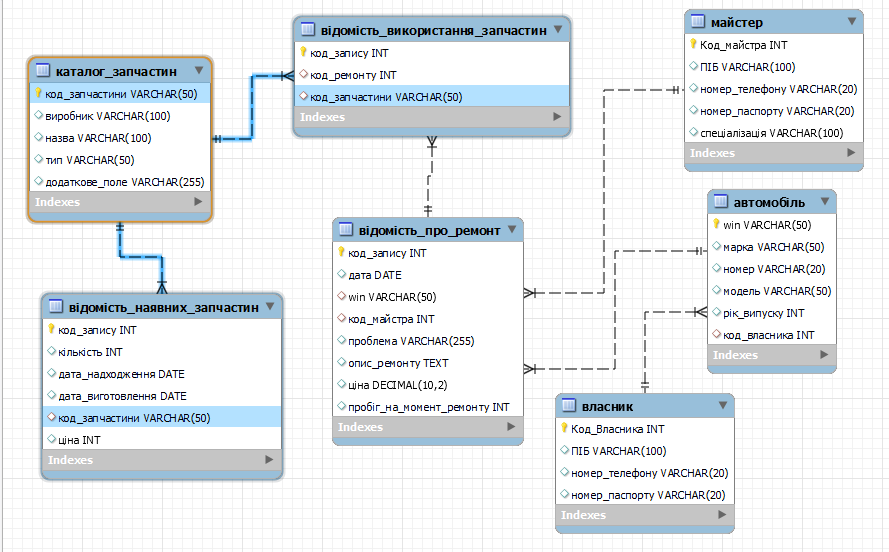
<https://chat.openai.com/>

<https://bard.google.com/chat?hl=uk>

**Додатки**

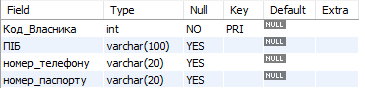
Додаток А

Схема Бази даних

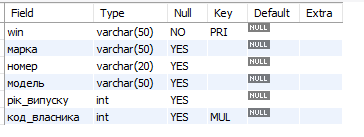


Додаток Б

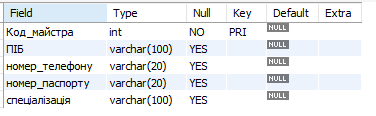
Опис таблиці Власник



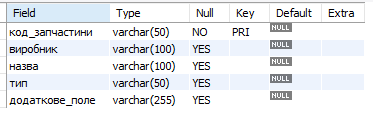
Опис таблиці Автомобіль



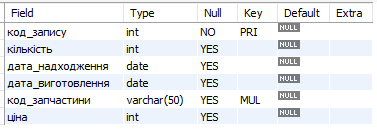
Опис таблиці Майстер



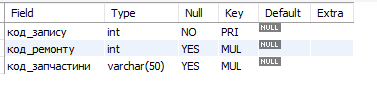
Опис таблиці Каталог\_запчастин



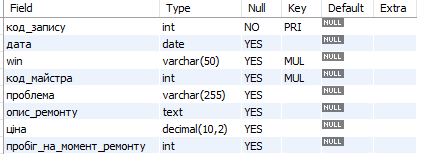
Опис таблиці Відомість\_наявних\_запчастин



Опис таблиці Відомість\_використання\_запчастин



Опис таблицы Відомість\_про\_ремонт



Додаток В

use kursova;

CREATE TABLE Власник (

Код\_Власника INT PRIMARY KEY,

ПІБ VARCHAR(100),

номер\_телефону VARCHAR(20),

номер\_паспорту VARCHAR(20)

);

describe Власник;

CREATE TABLE Автомобіль (

win VARCHAR(50) PRIMARY KEY,

марка VARCHAR(50),

номер VARCHAR(20),

модель VARCHAR(50),

рік\_випуску INT,

код\_власника INT,

FOREIGN KEY (код\_власника) REFERENCES Власник(Код\_Власника)

);

describe Автомобіль;

CREATE TABLE Майстер (

Код\_майстра INT PRIMARY KEY,

ПІБ VARCHAR(100),

номер\_телефону VARCHAR(20),

номер\_паспорту VARCHAR(20),

спеціалізація VARCHAR(100)

);

describe Майстер;

CREATE TABLE Каталог\_запчастин (

код\_запчастини VARCHAR(50) PRIMARY KEY,

виробник VARCHAR(100),

назва VARCHAR(100),

тип VARCHAR(50),

додаткове\_поле VARCHAR(255)

);

describe Каталог\_запчастин;

CREATE TABLE Відомість\_наявних\_запчастин (

код\_запису INT PRIMARY KEY,

кількість INT,

дата\_надходження DATE,

дата\_виготовлення DATE,

код\_запчастини VARCHAR(50),

ціна INT,

FOREIGN KEY (код\_запчастини) REFERENCES Каталог\_запчастин(код\_запчастини)

);

describe Відомість\_наявних\_запчастин;

CREATE TABLE Відомість\_про\_ремонт (

код\_запису INT PRIMARY KEY,

дата DATE,

win VARCHAR(50),

код\_майстра INT,

проблема VARCHAR(255),

опис\_ремонту TEXT,

ціна DECIMAL(10, 2),

пробіг\_на\_момент\_ремонту INT,

FOREIGN KEY (win) REFERENCES Автомобіль(win),

FOREIGN KEY (код\_майстра) REFERENCES Майстер(Код\_майстра)

);

describe Відомість\_про\_ремонт;

CREATE TABLE Відомість\_використання\_запчастин (

код\_запису INT PRIMARY KEY,

код\_ремонту INT,

код\_запчастини VARCHAR(50),

FOREIGN KEY (код\_ремонту) REFERENCES Відомість\_про\_ремонт(код\_запису),

FOREIGN KEY (код\_запчастини) REFERENCES Каталог\_запчастин(код\_запчастини)

);

describe Відомість\_використання\_запчастин;

INSERT INTO Власник(Код\_власника, ПІБ, номер\_телефону, номер\_паспорту)

VALUES (1, 'Іванов Іван', 930000000, 'AA0000'),

(2, 'Іванова Іванна', 931111111, 'AI1111'),

(3, 'Петров Петро', 731111000, 'BB0000'),

(4, 'Сидоренко Олег', 961234567, 'NN05343'),

(5, 'Боднарчук Віктор', 639876543, 'AA1111'),

(6, 'Петрова Іванна', 631234567, 'AI8753');

INSERT INTO Автомобіль(win, марка, номер, модель, рік\_випуску, код\_власника)

VALUES ('NA45327', 'Toyota', 'AA0000BP', 'Supra', 1998, 1),

('NB67849', 'Toyota', 'AA0111BP', 'Land Cruiser 200', 2015, 1),

('GP88749', 'Tesla', 'BC9864YZ', 'Model S Plaid', 2022, 2),

('NA00012', 'Nissan', 'IO4444OI', 'SKYLINE R-32', 1996, 3),

('BB12345', 'Toyota', 'II1234PH', 'Corola', 2018, 4),

('BB98765', 'Skoda', 'AA2233AP', 'Fabia', 2012, 4),

('GP65437', 'Tesla', 'HC4567YD', 'Model X Performance', 2018, 5),

('OU45327', 'Mazda', 'BC5464HC', 'MX-5 RF', 2022, 5),

('BB33344', 'Mazda', 'AP2233AP', 'RX-8', 2005, 6);

INSERT INTO Майстер (Код\_майстра, ПІБ, номер\_телефону, номер\_паспорту, спеціалізація)

VALUES

(1, 'Іванов Іван Іванович', '+1234567890', 'AB123456', 'Ремонт двигунів'),

(2, 'Петров Петро Петрович', '+9876543210', 'CD654321', 'Ремонт підвіски'),

(3, 'Сидоров Сидір Сидорович', '+1112223333', 'EF987654', 'Електрик'),

(4, 'Коваленко Олександр Олександрович', '+4445556666', 'GH123456', 'Ремонт двигунів'),

(5, 'Михайленко Михайло Михайлович', '+7778889999', 'IJ987654', 'Діагностування ЕБК');

INSERT INTO Каталог\_запчастин (код\_запчастини, виробник, назва, тип, додаткове\_поле)

VALUES

('1', 'Bosch', 'Свічка запалювання', 'Двигун', 'Для бензинових двигунів'),

('2', 'ContiTech', 'Ремінь ГРМ', 'Двигун', 'Довжина - 1000мм'),

('3', 'Mann-Filter', 'Фільтр повітряний', 'Фільтрація', 'Для дизельних двигунів'),

('4', 'Brembo', 'Гальмівні колодки', 'Гальма', 'Керамічні, для передніх коліс'),

('5', 'Sachs', 'Амортизатори', 'Підвіска', 'Для задньої підвіски'),

('6', 'Castrol', 'Моторне масло', 'Мастило', 'Синтетичне, в"язкість 5W-30'),

('7', 'NGK', 'Керамічні свічки', 'Двигун', 'Для сучасних автомобілів'),

('8', 'Bosch', 'Мідний кабель багатожильний', 'Електрика', 'Струм до 30А'),

('9', 'Revo', 'Електронний блок керування STAGE-1', 'Електрика', 'Для WAG');

INSERT INTO Відомість\_наявних\_запчастин (код\_запису, кількість, дата\_надходження, дата\_виготовлення, код\_запчастини, ціна)

VALUES

(1, 10, '2023-01-15', '2022-12-20', '1', 100),

(2, 20, '2023-02-20', '2023-01-10', '2', 80),

(3, 15, '2023-03-10', '2022-11-05', '3', 50),

(4, 30, '2023-04-05', '2023-03-01', '4', 120),

(5, 25, '2023-05-12', '2023-04-20', '5', 200),

(6, 18, '2023-06-18', '2023-05-25', '6', 150),

(7, 22, '2023-07-22', '2023-06-10', '7', 90),

(8, 150, '2023-07-23', '2023-06-30','8', 10);

INSERT INTO Відомість\_про\_ремонт (код\_запису, дата, win, код\_майстра, проблема, опис\_ремонту, ціна, пробіг\_на\_момент\_ремонту)

VALUES

(1, '2023-01-05', 'NA45327', 1, 'Проблеми з запалюванням', 'Заміна свічок запалювання та налаштування системи запалювання', 150.00, 125000),

(2, '2023-02-10', 'NB67849', 2, 'Шум у ГРМ', 'Заміна ременя ГРМ та регулювання його натягу', 200.00, 95000),

(3, '2023-03-15', 'GP88749', 3, 'Проблеми з фільтрацією', 'Заміна повітряного фільтра та проведення комплексної діагностики', 80.00, 75000),

(4, '2023-04-20', 'NA00012', 4, 'Проблеми з гальмами', 'Заміна гальмівних колодок та обстеження гальмівної системи', 300.00, 105000),

(5, '2023-05-25', 'BB12345', 5, 'Проблема з підвіскою', 'Заміна амортизаторів та перевірка стану підвіски', 400.00, 85000),

(6, '2023-06-30', 'BB98765', 1, 'Проблеми з двигуном', 'Проведення комплексної діагностики двигуна та заміна масла', 250.00, 115000),

(7, '2023-07-05', 'GP65437', 2, 'Проблеми з електрикою', 'Пошук та виправлення короткого замикання в електричній системі', 180.00, 90000),

(8, '2023-08-10', 'OU45327', 3, 'Проблема з двигуном', 'Заміна свічок запалювання та проведення комп"ютерної діагностики', 160.00, 98000),

(9, '2023-09-15', 'BB33344', 4, 'Проблеми з ГРМ', 'Регулювання ременя ГРМ та перевірка роботи', 220.00, 75000),

(10, '2023-10-20', 'BB12345', 5, 'Проблеми з трансмісією', 'Заміна моторного масла та перевірка коробки передач', 280.00, 110000),

(11, '2023-11-25', 'NA00012', 1, 'Проблема з електрикою', 'Заміна запалювальних свічок та перевірка датчиків', 200.00, 99000),

(12, '2023-12-30', 'GP88749', 2, 'Проблеми з ГРМ', 'Заміна ременя ГРМ та регулювання натягу', 180.00, 88000),

(13, '2024-01-05', 'OU45327', 3, 'Проблема з підвіскою', 'Заміна амортизаторів та перевірка стану підвіски', 300.00, 95000),

(14, '2024-02-10', 'BB33344', 4, 'Проблеми з гальмами', 'Заміна гальмівних колодок та регулювання гальмівної системи', 280.00, 105000),

(15, '2024-03-15', 'BB98765', 5, 'Проблема з двигуном', 'Проведення комп"ютерної діагностики та заміна фільтра повітряного', 150.00, 115000);

INSERT INTO Відомість\_використання\_запчастин (код\_запису, код\_ремонту, код\_запчастини)

VALUES

(1, 1, '1'),

(2, 2, '2'),

(3, 3, '3'),

(4, 4, '4'),

(5, 5, '5'),

(6, 6, '1'),

(7, 6, '1'),

(8, 7, '8'),

(9, 8, '1'),

(10, 9, '2'),

(11, 10, '6'),

(12, 11, '7'),

(13, 12, '2'),

(14, 13, '5'),

(15, 14, '4'),

(16, 15, '3');

SELECT \* FROM Власник;

SELECT \* FROM Автомобіль;

SELECT \* FROM Майстер;

SELECT \* FROM Каталог\_запчастин;

SELECT \* FROM Відомість\_наявних\_запчастин;

SELECT \* FROM Відомість\_про\_ремонт;

SELECT \* FROM Відомість\_використання\_запчастин;

CREATE TABLE Відомість\_наявних\_запчастин1

AS(SELECT\* from Відомість\_наявних\_запчастин WHERE ціна> 80);

SELECT\* from Відомість\_наявних\_запчастин1;

CREATE TABLE Відомість\_наявних\_запчастин2

AS(SELECT\* from Відомість\_наявних\_запчастин WHERE ціна < 120);

SELECT\* from Відомість\_наявних\_запчастин2;

SELECT\* from Відомість\_наявних\_запчастин1

INTERSECT

SELECT\* from Відомість\_наявних\_запчастин2;

SELECT\* from Відомість\_наявних\_запчастин1

UNION

SELECT\* from Відомість\_наявних\_запчастин2;

SELECT\* from Відомість\_наявних\_запчастин1

EXCEPT

SELECT\* from Відомість\_наявних\_запчастин2;

SELECT\* from Відомість\_наявних\_запчастин1 CROSS JOIN каталог\_запчастин;

SELECT марка, модель, рік\_випуску from Автомобіль;

SELECT \* from каталог\_запчастин WHERE (тип = 'двигун');

SELECT власник.ПІБ, марка, модель, номер

from власник, автомобіль

WHERE власник.код\_власника = автомобіль.код\_власника;

SELECT \* from відомість\_наявних\_запчастин1

JOIN відомість\_наявних\_запчастин2

ON відомість\_наявних\_запчастин1.ціна = відомість\_наявних\_запчастин2.ціна;

ALTER TABLE Автомобіль

ADD COLUMN модель\_двигуна VARCHAR(25) NOT NULL;

ALTER TABLE Автомобіль

DROP COLUMN модель\_двигуна;

UPDATE каталог\_запчастин

SET додаткове\_поле = 'ДЛЯ VAG. Лише під замовлення'

WHERE код\_запчастини = '9';

INSERT INTO Відомість\_наявних\_запчастин (код\_запису, кількість, дата\_надходження, дата\_виготовлення, код\_запчастини, ціна)

VALUES

(9, 0, '2023-05-20', '2022-01-23', '9', 50);

DELETE from Відомість\_наявних\_запчастин

WHERE код\_запису = 9;

UPDATE автомобіль

SET номер = 'AA6573BP', код\_власника = 1

WHERE win = 'BB12345';

DROP TABLE Відомість\_наявних\_запчастин1;

DROP TABLE Відомість\_наявних\_запчастин2;