

КУРСОВА РОБОТА
КР.ІПЗ-16.00.000 ПЗ
Група ІПЗ-2018
Піхоцький Назарій

2020

Івано-Франківський університет
Короля Данила

Кафедра програмної інженерії та інформаційних технологій

КУРСОВА РОБОТА

з дисципліни ОРГАНІЗАЦІЯ БАЗ ДАНИХ

Тема – Розробка демонстраційного програмного забезпечення
Моделі та базові характеристики накопичувачів даних

ПОЯСНУВАЛЬНА ЗАПИСКА

КР.ІІЗ – 16.00.000. ІІЗ

Студент гр.ІІЗ-2018 _____ (Піхоцький Н.Т.)
(підпис)

Допускається до захисту
Керівник курсового проекту

(посада) (підпис) (дата) (розшифровка підпису)

м.Івано-Франківськ
2020

Івано-Франківський університет Короля Данила
Кафедра програмної інженерії та інформаційних технологій
Дисципліна – Організація баз даних
Спеціальність – Інженерія програмного забезпечення
Курс - 3 Група - ІІІЗ-2018 Семестр - 5

ЗАВДАННЯ НА КУРСОВИЙ ПРОЕКТ

Студенту Піхоцькому Назарію Тарасовичу

1. Тема проекту - Розробка демонстраційного програмного забезпечення, моделі та базові характеристики накопичувачів даних.
2. Термін здачі студентом закінченого проекту – 10 грудня 2020 року.
3. Зміст пояснювальної записки:
 - ознайомлення з вимогами до розробки структури бази даних;
 - розробка структури бази даних, створення таблиць ключів і зав'язків між ними;
 - розробка програмного забезпечення на мові C# і SQL Server, реалізація функціоналу бази даних, демонстрація пошуку по запиту.
 - огляд роботи програми, перевірка функціоналу бази даних (Insert, Update, Delete)
4. Дата видачі завдання - 15 вересня 2020 року

КАЛЕНДАРНИЙ ПЛАН

Номер і назва етапів курсової роботи	Термін виконання етапів роботи	Примітка
1. Аналіз теми курсової роботи	15.09.2020-17.09.2020	
2. Розробка вступу і пояснюючої записки	20.09.2020-25.09.2020	
3. Розробка Баз Даних	20.10.2020-23.10.2020	
4. Розробка ПЗ	26.10.2020-29.10.2020	
5. Створення інтерфейсу програми	30.10.2020-30.10-2020	
6. Опис програми в пояснювальній записці	15.11.2020-20.11.2020	
7. Висновки	21.11.2020-21.11.2020	
8. Здача курсової роботи	14.12.2020	

Студент _____

(підпис)

(розшифровка підпису)

Керівник проекту _____

(підпис)

(розшифровка підпису)

Анотація

У курсовому проєкті було розроблено базу даних моделей та характеристик накопичувачів даних, та допоміжне програмне забезпечення для відображення інформації. Для розробки бази даних було використано реляційну базу даних MS SQL Server. Допоміжне програмне забезпечення розроблене за допомогою C#.

У першому розділі описано вимоги до проектування та характерні особливості інформаційного процесу та способи представлення даних, які беруться за основу при проектуванні структури бази даних.

У другому розділі розроблено таблиці бази даних, проведений аналіз всіх полів і їх ключів, створені зв'язки, описаний етап створення таблиць через запити.

У третьому розділі коротко описано підключення демонстраційного програмного забезпечення до бази даних, розробку програмного забезпечення, розглянуто вивід всіх даних з трьох таблиць, продемонстровано пошук по таблиці через запит.

У четвертому розділі продемонстровано роботу програмного забезпечення і його функціоналу такого як Insert, Update, Delete.

Summary

The course project developed a database of models and characteristics of data drives, and auxiliary software for displaying information. The MS SQL Server relational database was used to develop the database. Auxiliary software is developed using C#.

The first section describes the design requirements and the specific features of the information process and the ways in which data is presented, which are taken as a basis for designing the database structure.

The second section develops database tables, analysis all fields and their keys, connections created, the stage of creating tables through queries is described.

The third section briefly describes the connection of demo software to the database, software development, the output of all data from the three tables is considered, and a table search is demonstrated through the query.

The fourth section demonstrates the operation of the software and its functionality such as Insert, Update, Delete.

ЗМІСТ

ВСТУП.....	7 ст
1. Вимоги до проектування бази даних.....	9 ст
1.1 Вимоги до проектування бази даних.....	9 ст
1.2 Огляд існуючих рішень.....	11 ст
2. Розробка структури бази даних.....	13 ст
2.1 Проектування бази даних.....	13 ст
2.2 Створення таблиць.....	15 ст
3. Розробка програми	17 ст
3.1 Огляд виводу даних	17 ст
3.2 Розробка демонстраційної програми	18 ст
4 Огляд програми	22 ст
ВИСНОВКИ	24 ст
ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ	25 ст

ВСТУП

Основні ідеї сучасної інформаційної технології базуються на концепції, згідно якої дані мають бути організовані в бази даних з метою адекватного відображення реального світу, що змінюється, і задоволення інформаційних потреб користувачів. Ці бази даних створюються і функціонують під управлінням спеціальних програмних комплексів, званих системами управління базами даних (СУБД).

В загальному випадку база даних містить схеми, таблиці, подання, збережені процедури та інші об'єкти. Дані у базі організовують відповідно до моделі організації даних. Таким чином, сучасна база даних, крім саме даних, містить їх опис та може містити засоби для їх обробки.

Реляційна база даних — база даних, заснована на реляційній моделі даних. Інакше кажучи, реляційна база даних — це база даних, яка сприймається користувачем як набір нормалізованих відношень різного ступеня.

Проекти на основі безкоштовного ПЗ, які вимагають повнофункціональної системи керування базами даних часто використовують MS SQL Server.

Можливості сервера MS SQL Server:

- простота у використанні;
- підтримується необмежена кількість користувачів, що одночасно працюють із базою даних;
- кількість рядків у таблицях може досягати 50 млн.;
- висока швидкість виконання команд;
- наявність простої і ефективної системи безпеки

1 ВИМОГИ ДО ПРОЕКТУВАННЯ БАЗИ ДАНИХ

1.1 Вимоги до проектування бази даних

Головні ідеї сучасних інформаційних технологій базуються на концепції баз даних. Відповідно до цієї концепції, основою інформаційних технологій є дані, які повинні бути організовані в бази даних в цілях адекватного відображення мінливого реального світу і задоволення інформаційних потреб користувачів.

Одним з найважливіших і основоположних понять в теорії баз даних є поняття інформації. Під інформацією розуміються будь-які відомості про будь-яку подію, процеси, об'єкти. У свою чергу, дані - це інформація, представлена в певному вигляді, що дозволяє автоматизувати її збір, зберігання і подальшу обробку людиною або інформаційним засобом. Для комп'ютерних технологій дані - це інформація в дискретному, фіксованому вигляді, зручна для зберігання, обробки на ЕОМ, а також для передачі по каналах зв'язку.

Для коректного визначення вимог, необхідних для проектування бази даних моделей та базових характеристики накопичувачів даних, визначимо такі поняття. База даних (БД) - це сукупність взаємозв'язаних даних при такій мінімальній надмірності, яка допускає їх використання оптимальним чином для одного або декількох додатків в певній предметній області.

Система управління базами даних (СУБД) - сукупність мовних та програмних засобів, призначених для створення, ведення і сумісного використання БД багатьма користувачами.

У базі даних інформація повинна бути організована так, щоб забезпечити мінімальну частку її надмірності, оскільки надмірність даних тягне за собою ряд негативних наслідків: збільшення обсягу інформації, а отже, потреба в додаткових ресурсах для зберігання і обробки, а так само поява помилок при введенні дублюючої інформації, що порушують цілісність бази даних.

В БД повинні зберігатися дані, логічно пов'язані між собою. Для того щоб дані можна було пов'язати між собою, і зв'язати так, щоб ці зв'язки відповідали реально існуючим в даній предметній області, останню піддають детальному аналізу, виділяючи сутності чи об'єкти. Сутність або об'єкт - це те, про що необхідно зберігати інформацію. Сутності мають деякі характеристики, звані атрибутами, які теж необхідно зберігати в БД.

Визначивши сутності і їхні атрибути, необхідно перейти до виявлення зв'язків, які можуть існувати між деякими сутностями. Зв'язок - це те, що об'єднує дві або більше сутностей. Зв'язки між сутностями також є частиною даних, і вони також повинні зберігатися в базі даних.

Проектована база даних повинна володіти певними властивостями.

- Цілісність бази даних досягається внаслідок введення обмежень, пов'язані з нормалізацією БД. У кожен момент часу існування БД відомості, що містяться в ній, повинні бути несуперечливі.
- Відновлюваність - можливість відновлення БД після збою системи або окремих видів полумок системи.
- Безпека - припускає захист даних від навмисного і ненавмисного доступу, модифікації або руйнування.
- Ефективність. Властивість ефективності зазвичай оцінюється двома параметрами: мінімальний час реакції на запит користувача і мінімальні потреби в пам'яті, а так же поєднанням цих параметрів.

Для програмної реалізації бази даних були обрані мови SQL і C#, і відповідно, програмне забезпечення.

Розглянута не процедурна мова SQL (Structured Query Language - структурована мова запитів) орієнтована на операції з даними, представленими у вигляді логічно взаємопов'язаних сукупностей таблиць.

Особливість пропозицій цієї мови полягає в тому, що вони орієнтовані більшою мірою на кінцевий результат обробки даних, ніж на процедуру цієї обробки. SQL сам визначає, де знаходяться дані, які індекси і навіть які

найбільш ефективні послідовності операцій слід використовувати для їх отримання.

Згідно індивідуальним завданням, мета даної роботи полягає у проектуванні та розробці бази даних для відображення інформації про поточні накопичувачі даних.

Для реалізації поставленої мети ми скористаємося двома методами для проектування БД. Першим підходом до проектування є метод сутність - зв'язок, другим же є метод нормалізації відносин.

Модель сутність - зв'язок або по іншому ER-модель - це модель даних, що дозволяє описувати концептуальні схеми предметної області. Дана модель використовується при високорівневому проектуванні баз даних і дозволяє з її допомогою виділити ключові сутності і позначити зв'язки, які встановлюються між цими сутностями.

Нормалізація відносин - це формальний апарат обмежень на формування відносин, який дозволяє усунути дублювання і потенційну суперечливість збережених даних, таким чином зменшує трудовитрати на ведення БД. Процес нормалізації полягає в декомпозиції вихідних відносин на більш прості відносини, до тих пір, поки не вийде такий проект БД, в якому «кожен факт з'являється лише в одному місці».

1.2 Огляд існуючих рішень

Яскравим прикладом існуючих рішень з використанням MS SQL Server є сайт Microsoft який використовує їхню ж базу даних.

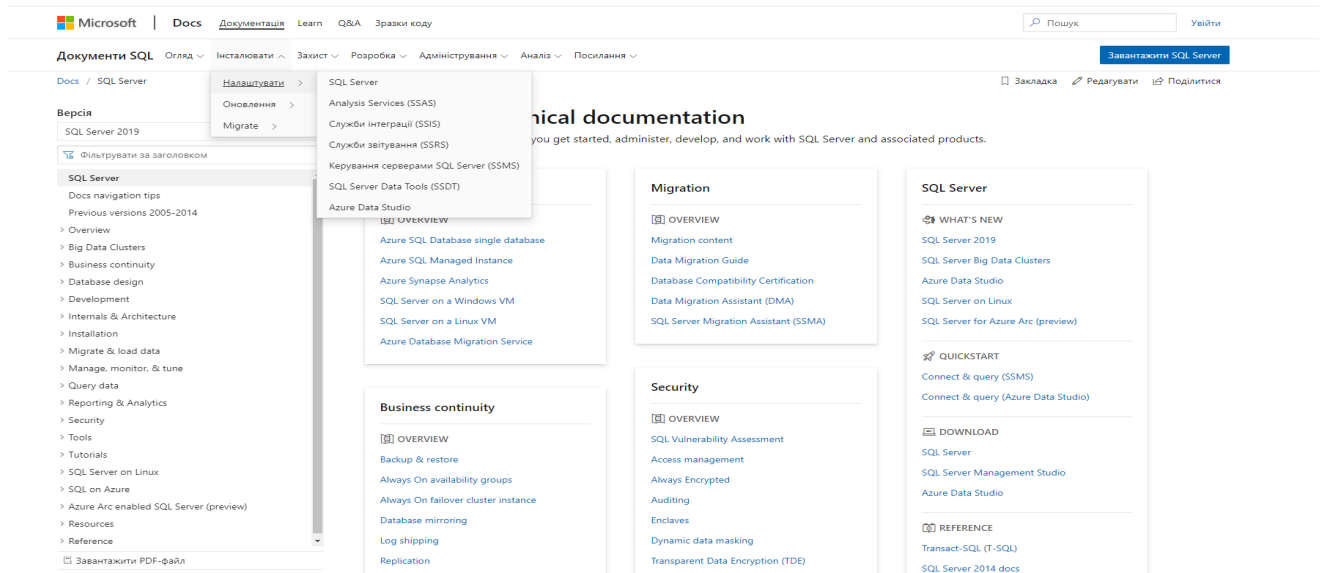


Рисунок 1.1 – Меню сайту Microsoft

На «Рисунку 1.1» в меню Документи SQL відображається розділ Огляд, Інсталювати, Захист, Розробка, Адміністрування, Аналіз, Посилання, в яких є випадające меню з під розділами. В розділі Інсталювати вибираємо Налаштувати і переходимо на сторінку SQL Server.

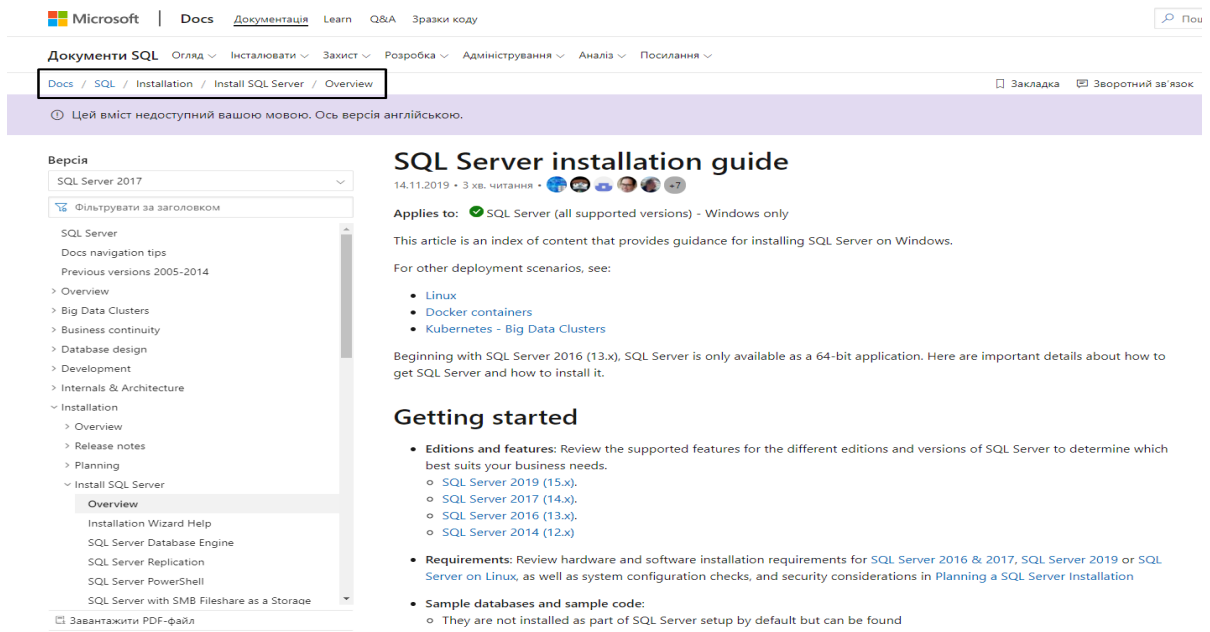


Рисунок 1.2 – Сторінка Налаштування MS SQL Server

На «Рисунку 1.2» відображається інформація про запити Налаштування – SQL Server, показано список налаштувань і установки SQL Server крок за кроком.

Кожний розділ Документи SQL містить під розділи по яких можна переходити і поступово просуватись по сторінці і шукати потрібну інформацію про SQL.

Огляд даного сайту допоміг мені з визначенням основного функціоналу для демонстраційного програмного забезпечення а також допоміг з формуванням Бази даних.

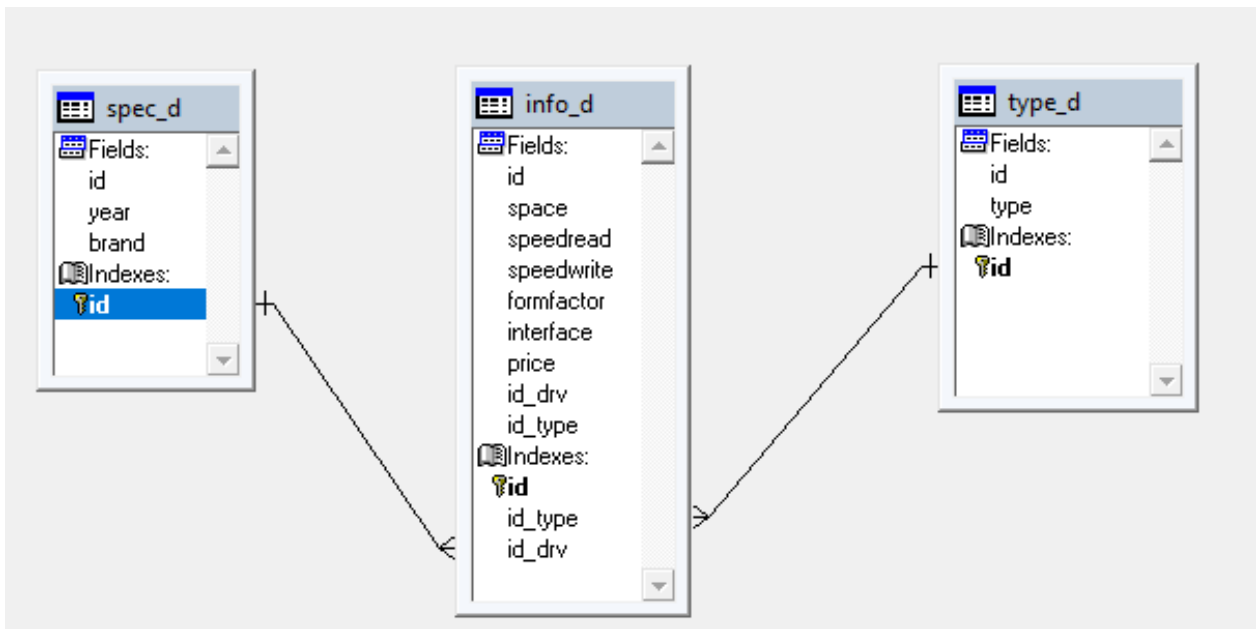
2 РОЗРОБКА СТРУКТУРИ БАЗИ ДАНИХ

2.1 Проектування бази даних

На першому етапі проектування необхідно визначити сутності. Сутність - це якийсь об'єкт, що представляє інтерес для користувача.

З точки зору заданої предметної області типовим користувачем розроблюваної бази даних буде проста людина, яка хоче вибрати накопичувач даних. Основне завдання даної бази даних саме забезпечити всією необхідною інформацією. Користувачем даної бази даних в тому числі може стати проста

людина,
так як в
базі буде



міститися інформація про види накопичувачів даних.

Рисунок 2.1 – Загальна структура бази даних

Для досягнення мети і вирішення поставленого завдання виділимо на основі аналізу предметної області наступні сутності: Spec_d(Специфікації), Info_d(Інформація), Type_d (Тип).

Другий етап - визначення зв'язків між виділеними сутностями.

Info_d(Інформація) містить посилання на Type_d (Тип) та посилання на Spec_d(Специфікації).

Далі визначимо атрибути або властивості для кожної з сутностей, а так само ключі для кожної сутності. Ключем сутності називається атрибут або набір атрибутів, що дозволяють однозначно ідентифікувати примірник сутності:

Type_d(Тип):

- Id(Ідентифікатор) – внутрішній ключ
- Type(Тип накопичувача) – типу nvarchar(50)

Info_d(Інформація):

- Id(Ідентифікатор) – внутрішній ключ
- Space(Простір) – типу nvarchar(50)
- SpeedRead(Швидкість читання) – типу int
- SpeedWrite(Швидкість запису) – типу int
- FormFactor(Форм фактор) – типу nvarchar(50)
- Interface(Інтерфейс підключення) – типу nvarchar(50)
- Price(Ціна) – типу int
- id_drv(Ідентифікатор пристрою) – типу int
- id_type(Ідентифікатор типу пристрою) – типу int

Spec_d(Специфікація):

- Id(Ідентифікатор) - внутрішній ключ
- Year(Рік випуску) - типу date
- Brand(Бренд накопичувача) – типу nvarchar(50)

П'ятий і шостий етапи проектування бази даних за методом сутність - зв'язок вимагають визначити ступінь зв'язку і клас приналежності. Ступінь

зв'язку показує скільки безпосередніх зв'язків може мати кожен екземпляр сутності з іншим екземпляром сутності, з яким він пов'язаний. Клас приналежності може бути обов'язковим і необов'язковим, якщо екземпляри даної суті мають брати участь у зв'язку, то участь називається обов'язковою, а якщо екземпляри можуть не брати участь у зв'язку, то необов'язковою.

Info_d(Інформація) містить посилання на Type_d(Тип).

Для вказання власності потрібно додати до Info_d(Інформація) атрибут ідентифікатор власності (id_type).

Info_d(Інформація) містить посилання на Spec_d(Специфікація).

Для вказання власності потрібно додати до Info_d(Інформація) атрибут ідентифікатор типу (id_drv).

2.2 Створення таблиць

Таблиці складають основу бази даних - саме в них зберігаються всі дані. Перед усім, повинна бути спланована структура кожної таблиці. Структура таблиць обумовлюється вмістом тих вихідних форм, запитів та звітів, які повинні бути отримані при роботі з базою даних.

При плануванні таблиць необхідно уникати повторення колонок в різних таблицях, тільки якщо вони не слугують для визначення зв'язків між ними.

Таблиця складається з записів (рядків), кожний з яких описує одну сутність. Кожна колонка таблиці - це поле. Поле містить однотипну інформацію, яка визначає тип даних.

Тип даних визначає вид і межі допустимих значень, які можуть бути введені в поле, а також об'єм пам'яті, який виділяється для цього поля, що важливо при проектуванні великих БД.

В створеній базі даних, створюємо таблиці за допомогою SQL запитів.

Створення таблиці Info_d

```
CREATE TABLE [dbo].[Info_d] (  
    [Id] INT IDENTITY (1, 1) NOT NULL,  
    [Space] NVARCHAR (MAX) NULL,  
    [SpeedRead] INT NULL,  
    [SpeedWrite] INT NULL,  
    [FormFactor] NVARCHAR (50) NULL,  
    [Interface] NVARCHAR (50) NULL,  
    [Price] INT NULL,  
    [id_drv] INT NULL,  
    [id_type] INT NULL,  
    CONSTRAINT [PK_Info_d] PRIMARY KEY CLUSTERED ([Id] ASC)  
);
```

Створення таблиці Spec_d

```
CREATE TABLE [dbo].[Spec_d] (  
    [Id] INT IDENTITY (1, 1) NOT NULL,  
    [Year] DATE NULL,  
    [Brand] NVARCHAR (50) NULL,  
    CONSTRAINT [PK_Spec_d] PRIMARY KEY CLUSTERED ([Id] ASC)  
);
```

Створення таблиці Type_d

```
CREATE TABLE [dbo].[Type_d] (  
    [Id] INT IDENTITY (1, 1) NOT NULL,  
    [Type] NVARCHAR (50) NULL,  
    PRIMARY KEY CLUSTERED ([Id] ASC)  
);
```

Запити створені в середовищі розробки Visual Studio. Створені первинні ключі для поля id, також реалізовано авто заповнення поля id. При видаленні поля з індексом 'x' ми більше не зустрінемо в цій таблиці це id і створити елемент в цій таблиці з таким id більше не можливо. Це зроблено для уникнення конфліктів між даними і створення унікальності кожного елемента таблиці.

3 РОЗРОБКА ПРОГРАМИ

3.1 Огляд виводу даних

У процесі проектування бази даних був розроблений ряд запитів, що забезпечує пошук, додавання, оновлення та видалення інформації, що зберігається в базі даних. Ці запити надають можливість в зручній формі переглянути інформацію про пристрої накопичування даних.

Запит для виводу всієї інформації з трьох таблиць, результат якого можна переглянути на Рисунку 3.1

```
SELECT Info_d.Id, Info_d.id_drv, Info_d.id_type, Space, SpeedRead,
SpeedWrite, FormFactor, Interface, Price, Spec_d.Id, Year, Brand,
type_d.Id, Type_d.Type
FROM ((Info_d INNER JOIN Spec_d ON Info_d.id_drv = Spec_d.id)
INNER JOIN Type_d ON Info_d.id_type = Type_d.Id)
```

	Id	id_drv	id_type	Space	SpeedRead	SpeedWrite	FormFactor	Interface	Price	Id	Year	Brand	Id	Type
1	5	3	1	480	500	450	2.5	SATA III	1749	3	2013-07-29	Kingston	1	SDD
2	6	10	3	500	3400	2300	M.2	PCI Express 3.0	2399	10	2015-06-09	Samsung	3	M2
3	7	8	1	250	550	520	2.5	SATA III	1399	8	2017-03-04	Samsung	1	SDD
4	8	7	1	1000	540	540	Portable	USB Type-A	4355	7	2017-03-01	Samsung	1	SDD
5	9	11	3	256	3200	3000	M.2	PCI Express 3.0	1469	11	2014-01-12	Silicon Power	3	M2
6	10	12	3	512	3200	3000	M.2	PCI Express 3.0	2309	12	2016-03-14	Silicon Power	3	M2
7	11	13	1	256	550	490	2.5	SATA III	949	13	2010-01-01	GoodRAM	1	SDD
8	12	14	1	512	550	500	2.5	SATA III	1679	14	2013-08-05	GoodRAM	1	SDD
9	13	15	2	1000	5000	5000	2.5	USB 3.1	1699	15	2010-06-14	Apacer	2	HDD
10	14	16	2	2000	194	194	3.5	SATA III	3378	16	2017-05-01	Seagate	2	HDD
11	15	17	2	1000	184	184	3.5	SATA III	2595	17	2013-08-03	Western Digital	2	HDD
12	16	18	2	12000	1024	1024	3.5	SAS	10...	18	2019-06-16	Western Digital	2	HDD
13	17	19	2	4000	226	226	3.5	SATA III	4635	19	2015-05-05	Seagate Exos	2	HDD
14	18	20	4	64	70	18	Portable	USB 3.0	340	20	2010-08-05	Kingston	4	Flash
15	19	21	4	8	30	15	Portable	USB 2.0	120	21	2003-02-01	Silicon Power	4	Flash
16	20	22	4	24	5000	3200	Portable	USB 3.0	340	22	2003-06-05	Transcend	4	Flash
17	21	23	4	124	340	200	Portable	USB 3.0	280	23	2014-03-06	Transcend	4	Flash
18	22	24	6	0.7	52	52	CD	CD Drive	17	24	2000-07-07	LG	6	CD
19	23	25	6	3.7	54	54	CD	CD Drive	30	25	2002-02-01	LG	6	CD
20	24	26	6	4.2	52	52	CD	CD Drive	49	26	2007-08-05	Philips	6	CD

Рисунок 3.1 – Вивід даних

3.2 Розробка демонстраційної програми

Для створення демонстраційної програми було використано мову програмування C# і Базу даних Microsoft SQL Server.

Візуальну частину було розроблено за допомогою WinForms і DataGridView.

Першим кроком було створення бази даних (див. додаток 1). Після чого створення таблиць і заповнення її даними (див. додаток 2).

Дальше було створено WinForms для кожної таблиці і одне головне вікно в якому можемо рухатись між таблицями.

Для кожної форми підключаємо простір імен для роботи з базою даних
`using System.Data.SqlClient.`

Головним є підключення нашої бази до WinForm, що можна зробити за допомогою `SqlConnection`.

```
SqlConnection=  
new SqlConnection(@"DataSource=(LocalDB)\MSSQLLocalDB;  
AttachDbFilename=C:\Users\Nazar.s\Desktop\сру\App\App\Database1.mdf;  
Integrated Security=True");
```

Наступним кроком буде створення методу для загрузки даних в DataGridView (див. додаток 3). Дальше добавляємо кнопку оновлення бази даних для того, щоб після її заповнення або внесення змін ці зміни збереглись в базу даних (див. додаток 4).

Тепер добавляємо функціонал (insert, delete, update) за допомогою `dataGridView_CellContentClick`. Спочатку перевіряємо кількість стовпців і в останній стовбець добавляємо функцію Delete, якщо користувач вніс зміни з рядок, параметр Delete міняється на Update, і якщо користувач вводить нові дані в пустий рядок параметр Delete міняється на Insert (див. додаток 5).

Останнім потрібно реалізувати пошук, його було реалізовано за допомогою SqlDataAdapter і запитом:

```
SELECT * FROM Info_d WHERE id_type LIKE '%' + SearchBox.Text + '%'  
OR Interface LIKE '%' + SearchBox.Text + '%'
```

Де ми шукаємо інформацію про пристрій який буде містити в собі значення поля SearchBox.Text із поля id_type таблиці Info_d зображено на Рисунок 3.2 і Рисунок 3.3.

Для таблиці Type_d реалізовано такий запит:

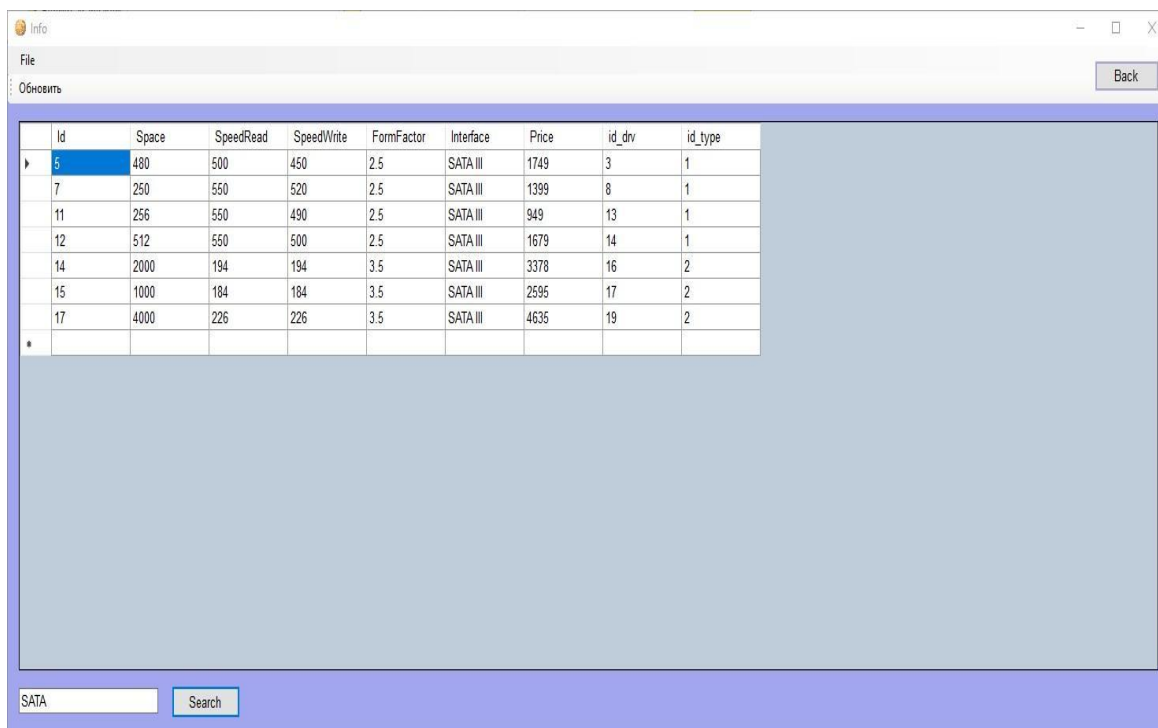
```
SELECT * FROM Type_d WHERE id LIKE '%' + SearchBox2.Text + '%'
```

Вибірка по полю id з таблиці Type_d зображено на Рисунок 3.4

Для таблиці Spec_d реалізовано такий запит:

```
SELECT * FROM Spec_d WHERE id LIKE '%' + Search_box1.Text + '%' OR  
Brand LIKE '%' + Search_box1.Text + '%'
```

Вибірка по полю id з таблиці Spec_d зображено на Рисунок 3.5 і Рисунок 3.6



Id	Space	SpeedRead	SpeedWrite	FormFactor	Interface	Price	id_drv	id_type
6	480	500	450	2.5	SATA III	1749	3	1
7	250	550	520	2.5	SATA III	1399	8	1
11	256	550	490	2.5	SATA III	949	13	1
12	512	550	500	2.5	SATA III	1679	14	1
14	2000	194	194	3.5	SATA III	3378	16	2
15	1000	184	184	3.5	SATA III	2595	17	2
17	4000	226	226	3.5	SATA III	4635	19	2

Рисунок 3.2 – Вибірка по полю Interface

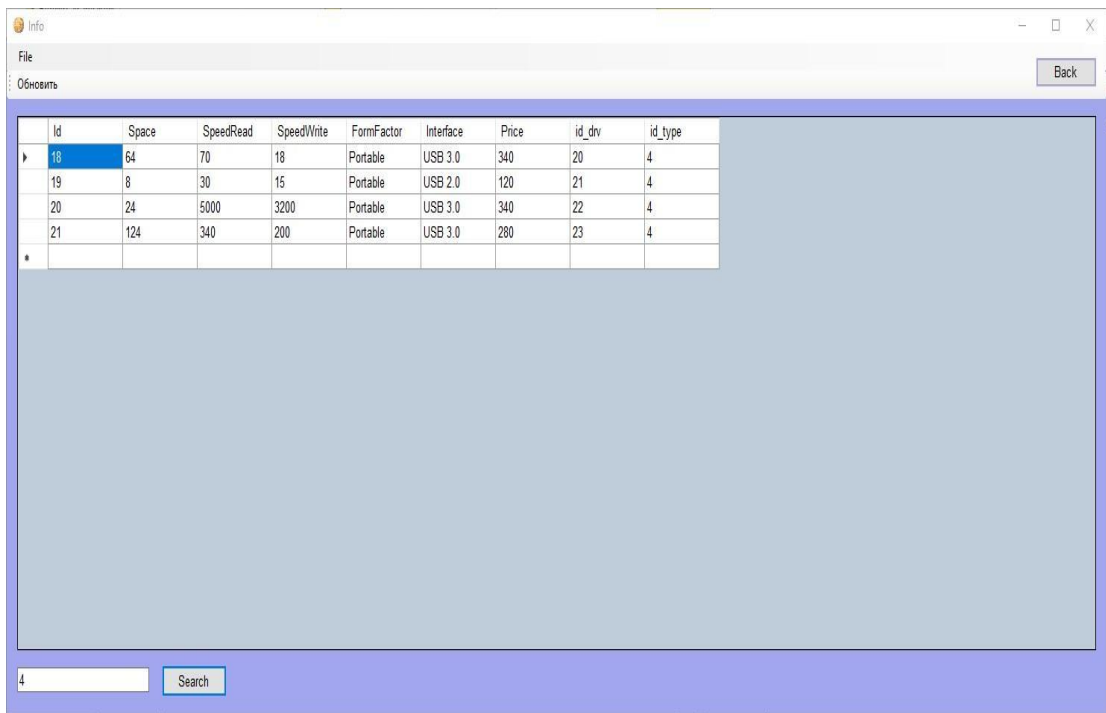


Рисунок 3.3 – Вибірка по полю id_type

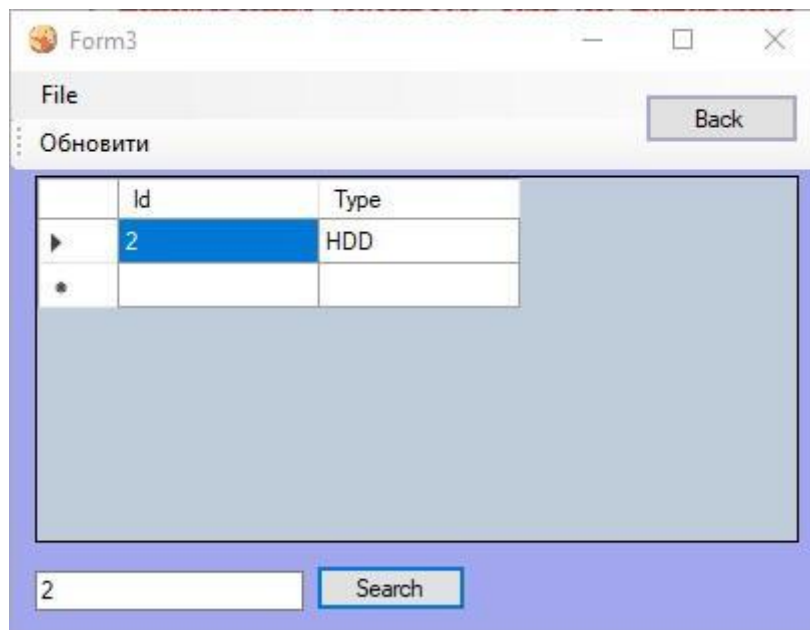


Рисунок 3.4 – Вибірка по полю id

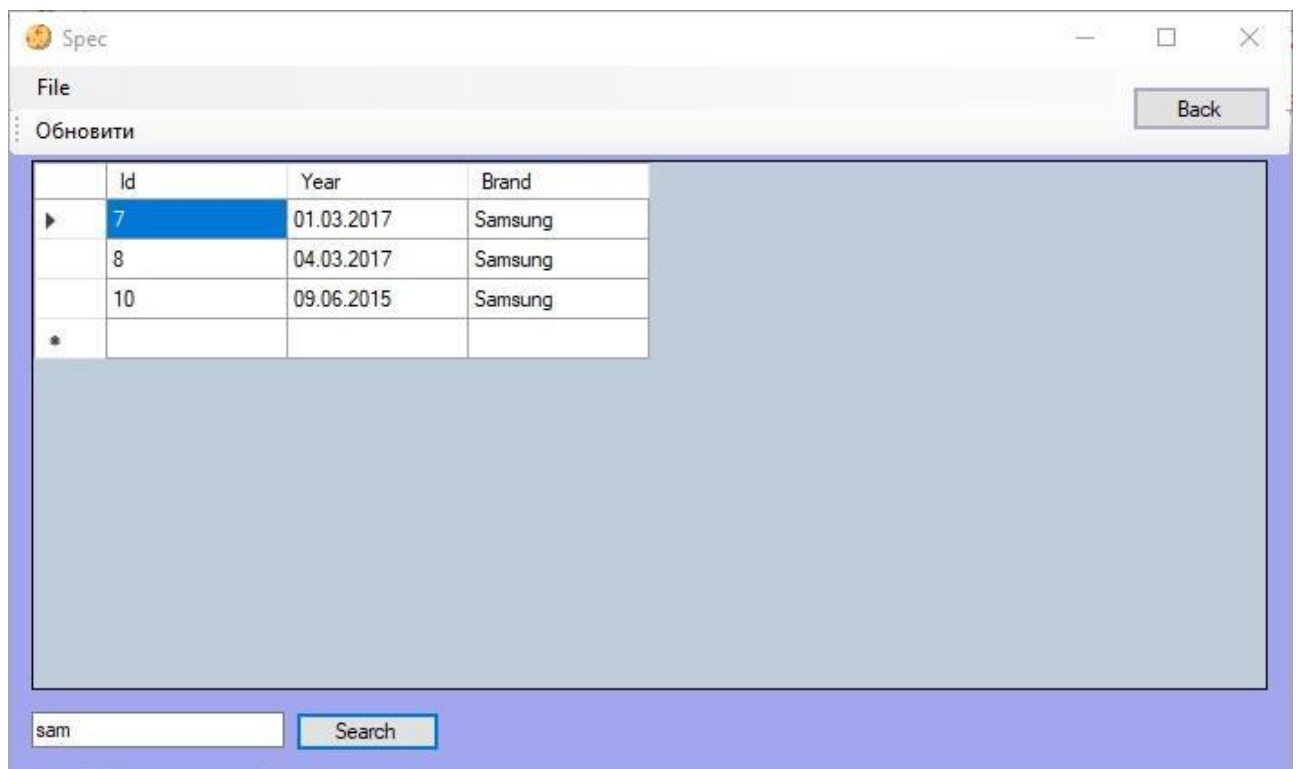


Рисунок 3.5 – Вибірка по полю Brand

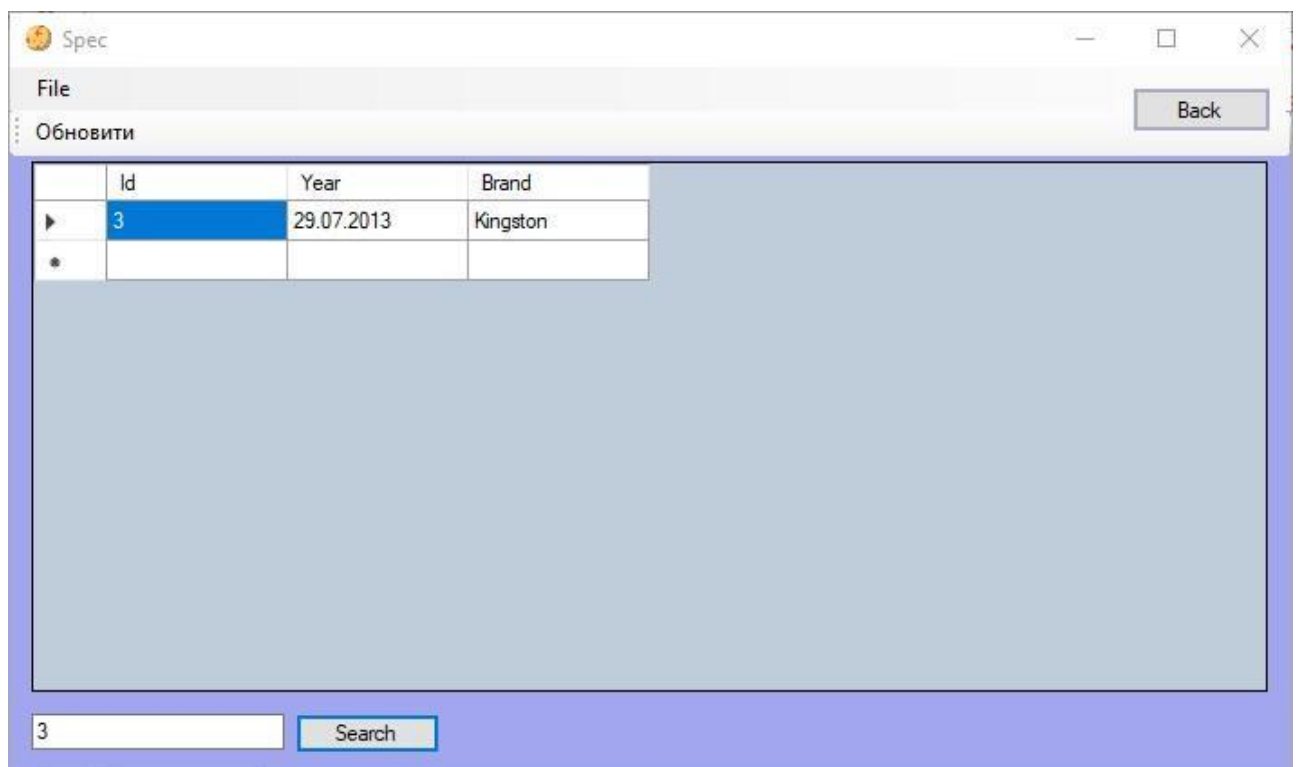
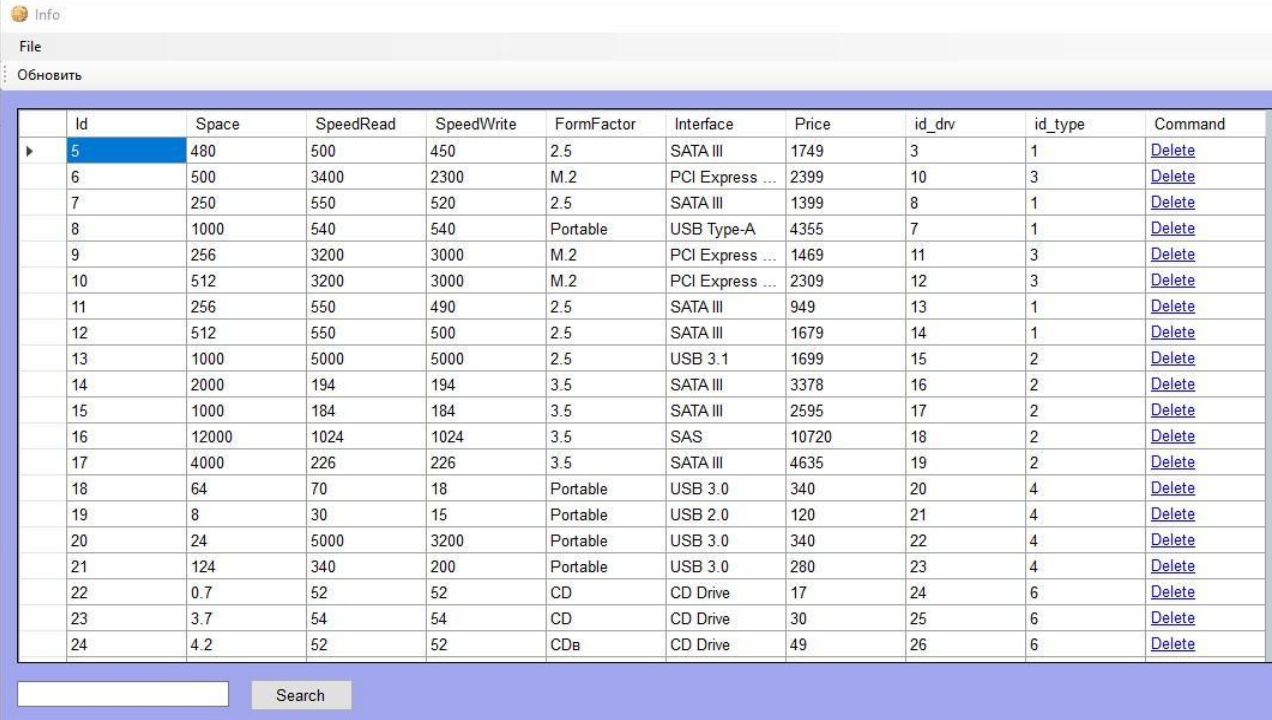


Рисунок 3.6 – Вибірка по полю id

4 ОГЛЯД ПРОГРАМИ

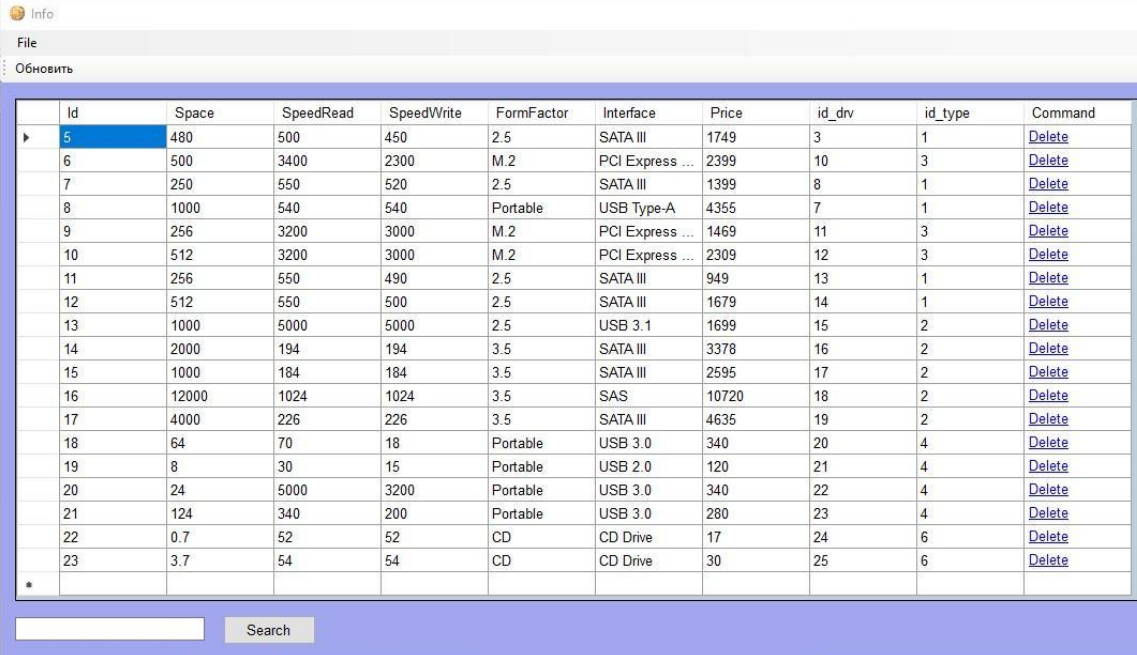
Демонстрацію роботи програми розглянемо на прикладі таблиці Info_d
Ось такий вигляд має таблиця Info_d зображено на рисунку 4.1



Id	Space	SpeedRead	SpeedWrite	FormFactor	Interface	Price	id_drv	id_type	Command
5	480	500	450	2.5	SATA III	1749	3	1	Delete
6	500	3400	2300	M.2	PCI Express ...	2399	10	3	Delete
7	250	550	520	2.5	SATA III	1399	8	1	Delete
8	1000	540	540	Portable	USB Type-A	4355	7	1	Delete
9	256	3200	3000	M.2	PCI Express ...	1469	11	3	Delete
10	512	3200	3000	M.2	PCI Express ...	2309	12	3	Delete
11	256	550	490	2.5	SATA III	949	13	1	Delete
12	512	550	500	2.5	SATA III	1679	14	1	Delete
13	1000	5000	5000	2.5	USB 3.1	1699	15	2	Delete
14	2000	194	194	3.5	SATA III	3378	16	2	Delete
15	1000	184	184	3.5	SATA III	2595	17	2	Delete
16	12000	1024	1024	3.5	SAS	10720	18	2	Delete
17	4000	226	226	3.5	SATA III	4635	19	2	Delete
18	64	70	18	Portable	USB 3.0	340	20	4	Delete
19	8	30	15	Portable	USB 2.0	120	21	4	Delete
20	24	5000	3200	Portable	USB 3.0	340	22	4	Delete
21	124	340	200	Portable	USB 3.0	280	23	4	Delete
22	0.7	52	52	CD	CD Drive	17	24	6	Delete
23	3.7	54	54	CD	CD Drive	30	25	6	Delete
24	4.2	52	52	CD	CD Drive	49	26	6	Delete

Рисунок 4.1 – Дані таблиці Info_d

Перевіримо оператор видалення рядка з бази даних, візьмемо пристрій під індексом 24 і видалимо його, рисунок 4.2



Id	Space	SpeedRead	SpeedWrite	FormFactor	Interface	Price	id_drv	id_type	Command
5	480	500	450	2.5	SATA III	1749	3	1	Delete
6	500	3400	2300	M.2	PCI Express ...	2399	10	3	Delete
7	250	550	520	2.5	SATA III	1399	8	1	Delete
8	1000	540	540	Portable	USB Type-A	4355	7	1	Delete
9	256	3200	3000	M.2	PCI Express ...	1469	11	3	Delete
10	512	3200	3000	M.2	PCI Express ...	2309	12	3	Delete
11	256	550	490	2.5	SATA III	949	13	1	Delete
12	512	550	500	2.5	SATA III	1679	14	1	Delete
13	1000	5000	5000	2.5	USB 3.1	1699	15	2	Delete
14	2000	194	194	3.5	SATA III	3378	16	2	Delete
15	1000	184	184	3.5	SATA III	2595	17	2	Delete
16	12000	1024	1024	3.5	SAS	10720	18	2	Delete
17	4000	226	226	3.5	SATA III	4635	19	2	Delete
18	64	70	18	Portable	USB 3.0	340	20	4	Delete
19	8	30	15	Portable	USB 2.0	120	21	4	Delete
20	24	5000	3200	Portable	USB 3.0	340	22	4	Delete
21	124	340	200	Portable	USB 3.0	280	23	4	Delete
22	0.7	52	52	CD	CD Drive	17	24	6	Delete
23	3.7	54	54	CD	CD Drive	30	25	6	Delete
*									

Рисунок 4.2 – Дані таблиці Info_d після Delete

Як ми бачимо такого запису більше немає в базі. Тепер поспробуємо оновити дані по id 20, рисунок 4.3.

Id	Space	SpeedRead	SpeedWrite	FormFactor	Interface	Price	id_drv	id_type	Command
5	480	500	450	2.5	SATA III	1749	3	1	Delete
6	500	3400	2300	M.2	PCI Express ...	2399	10	3	Delete
7	250	550	520	2.5	SATA III	1399	8	1	Delete
8	1000	540	540	Portable	USB Type-A	4355	7	1	Delete
9	256	3200	3000	M.2	PCI Express ...	1469	11	3	Delete
10	512	3200	3000	M.2	PCI Express ...	2309	12	3	Delete
11	256	550	490	2.5	SATA III	949	13	1	Delete
12	512	550	500	2.5	SATA III	1679	14	1	Delete
13	1000	5000	5000	2.5	USB 3.1	1699	15	2	Delete
14	2000	194	194	3.5	SATA III	3378	16	2	Delete
15	1000	184	184	3.5	SATA III	2595	17	2	Delete
16	12000	1024	1024	3.5	SAS	10720	18	2	Delete
17	4000	226	226	3.5	SATA III	4635	19	2	Delete
18	64	70	18	Portable	USB 3.0	340	20	4	Delete
19	8	30	15	Portable	USB 2.0	120	21	4	Delete
20	24	5000	3200	Portable	USB 3.0	3400	22	4	Update
21	124	340	200	Portable	USB 3.0	280	23	4	Delete
22	0.7	52	52	CD	CD Drive	17	24	6	Delete
23	3.7	54	54	CD	CD Drive	30	25	6	Delete

Рисунок 4.3 – Дані таблиці Info_d після Update

Як бачимо в таблиці Info_d поле DELETE змінилось на Update от і результат зберігся в базу.

Тепер можна переглянути оператор Insert, рисунок 4.4

Id	Space	SpeedRead	SpeedWrite	FormFactor	Interface	Price	id_drv	id_type	Command
6	500	3400	2300	M.2	PCI Express ...	2399	10	3	Delete
7	250	550	520	2.5	SATA III	1399	8	1	Delete
8	1000	540	540	Portable	USB Type-A	4355	7	1	Delete
9	256	3200	3000	M.2	PCI Express ...	1469	11	3	Delete
10	512	3200	3000	M.2	PCI Express ...	2309	12	3	Delete
11	256	550	490	2.5	SATA III	949	13	1	Delete
12	512	550	500	2.5	SATA III	1679	14	1	Delete
13	1000	5000	5000	2.5	USB 3.1	1699	15	2	Delete
14	2000	194	194	3.5	SATA III	3378	16	2	Delete
15	1000	184	184	3.5	SATA III	2595	17	2	Delete
16	12000	1024	1024	3.5	SAS	10720	18	2	Delete
17	4000	226	226	3.5	SATA III	4635	19	2	Delete
18	64	70	18	Portable	USB 3.0	340	20	4	Delete
19	8	30	15	Portable	USB 2.0	120	21	4	Delete
20	24	5000	3200	Portable	USB 3.0	3400	22	4	Update
21	124	340	200	Portable	USB 3.0	280	23	4	Delete
22	0.7	52	52	CD	CD Drive	17	24	6	Delete
23	3.7	54	54	CD	CD Drive	30	25	6	Delete
	3	44	5	CD	cd drive	3			Insert

Рисунок 4.4 – Дані таблиці Info_d після Insert

ВИСНОВКИ

Виконання курсової роботи практично закріпило знання з основ сучасної технології створення складних програмних продуктів з використанням баз даних. Такі знання можуть бути використані у розробках як системного, так і прикладного програмного забезпечення з урахуванням сучасних вимог у відношенні до надійності, якості інтерфейсу та ефективності програмних продуктів, які створюються.

Отримані знання та практичні навички мають служити базою для опанування у подальшому нових майбутніх систем програмування. Для виконання курсової роботи використались знання та практичні навички, що отримані на лекціях та лабораторних заняттях.

Використовуючи набуті знання створено базу даних, яка відображає інформацію про пристрої накопичення даних, та демонстраційне програмне забезпечення, яке полегшує роботу з базою даних.

Курсова робота була виконана з застосуванням мови програмування C#.

ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ

Основна:

1. Пасічник В.В., Резніченко В.А. Організація баз даних та знань. – К.: Видавнича група ВНУ, 2006.
2. Смирнов С.Н., Задворев И.С. Работаем с Oracle 9i – Х.: Изд-во при Харьк. унив-те, Фолио 2003.
3. Юрчишин В.М., Клим Б.В., Кропивницька В.Б. Організація баз даних. / Навчальний посібник – Івано-Франківськ: Факел, 2009.
4. Боуман, Эмерсон, Дарновски. Практическое руководство по SQL, 4-е издание.: Пер. с англ. – М.: Издательский дом «Вильямс», 2001.-352с.
5. Дейт, К.Дж. Введение в системы баз данных, 7-е издание.: Пер. с англ. – М.: Издательский дом «Вильямс», 2002.-1072с.
6. Документація MS SQL Server
<https://docs.microsoft.com/ru-ru/sql/sql-server/tutorials-for-sql-server-2016?view=sql-server-ver15>
7. Документація C#
<https://docs.microsoft.com/ru-ru/dotnet/csharp/>