

2. ТЕХНОЛОГІЇ ДОСТУПУ ДО ДАНИХ. З'ЄДНАННЯ З БД РІЗНИХ ФОРМАТІВ

2.1. Мета роботи

Закріпити знання з використання технологій доступу до даних в базах даних різних форматів [4-6]: MS Access, MS SQL Server та MySQL. Навчитися створювати базу даних в форматі MS SQL Server, використовуючи MS SQL Server Management Studio (SSMS), та в інших СУБД.

2.2 Методичні вказівки з організації самостійної роботи студентів

Необхідно повторити наступний теоретичний матеріал: створення нового Windows Application проекту в Microsoft Visual Studio, технологій доступу до даних, технологія доступу до даних ADO.NET [6].

2.3 Порядок виконання роботи та методичні вказівки до її виконання

Виконувати лабораторні роботи № 2 та 3 необхідно на базі власної БД. Можна в іншому середовищі, але знання з технології ADO.NET у Вас повинні бути. Питання за цією технологією будуть на екзамені. Отже, необхідно ретельно ознайомитися з прикладом виконання роботи для системи «Гуртожиток» (див. розділ 2.3.1), а потім виконати лабораторну роботу у відповідності до розділу 2.3.2 «Порядок виконання роботи».

2.3.1 Приклад виконання лабораторної роботи для ІС «Гуртожиток»

2.3.1.1 Створення проекту Windows Forms Application

Для створення нового проекту в Microsoft Visual Studio необхідно виконати наступну послідовність дій: на панелі вгорі вибрати **File** → **New** → **Project**. Далі в діалоговому вікні **New Project** вибрати **Templates** → **Visual C #** → **Windows** → **Windows Forms Application**. В поле **Name** задайте ім'я проекту

«Hostel» (див. рис. 2.1.1, рис. 2.1.2), в полі **Location** вкажіть шлях, куди ви хочете зберегти свій проект і натисніть кнопку ОК.

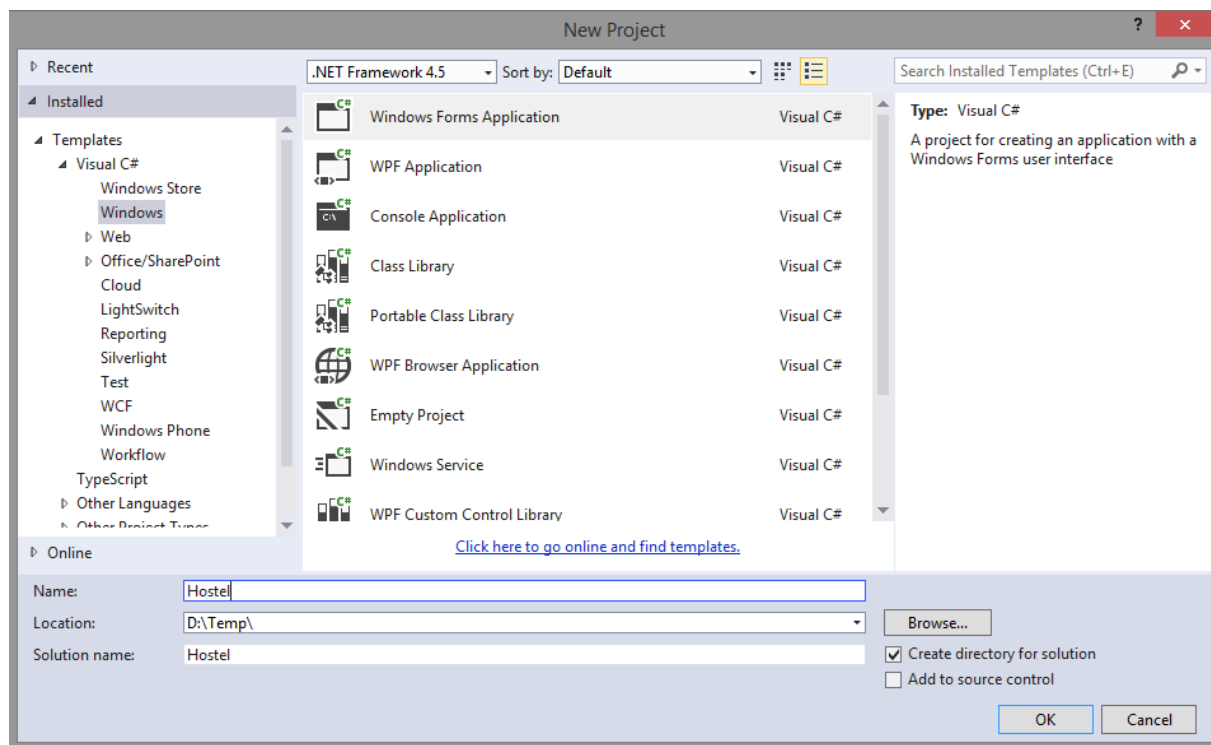


Рисунок 2.1.1 – Вибір типу проекту для створення

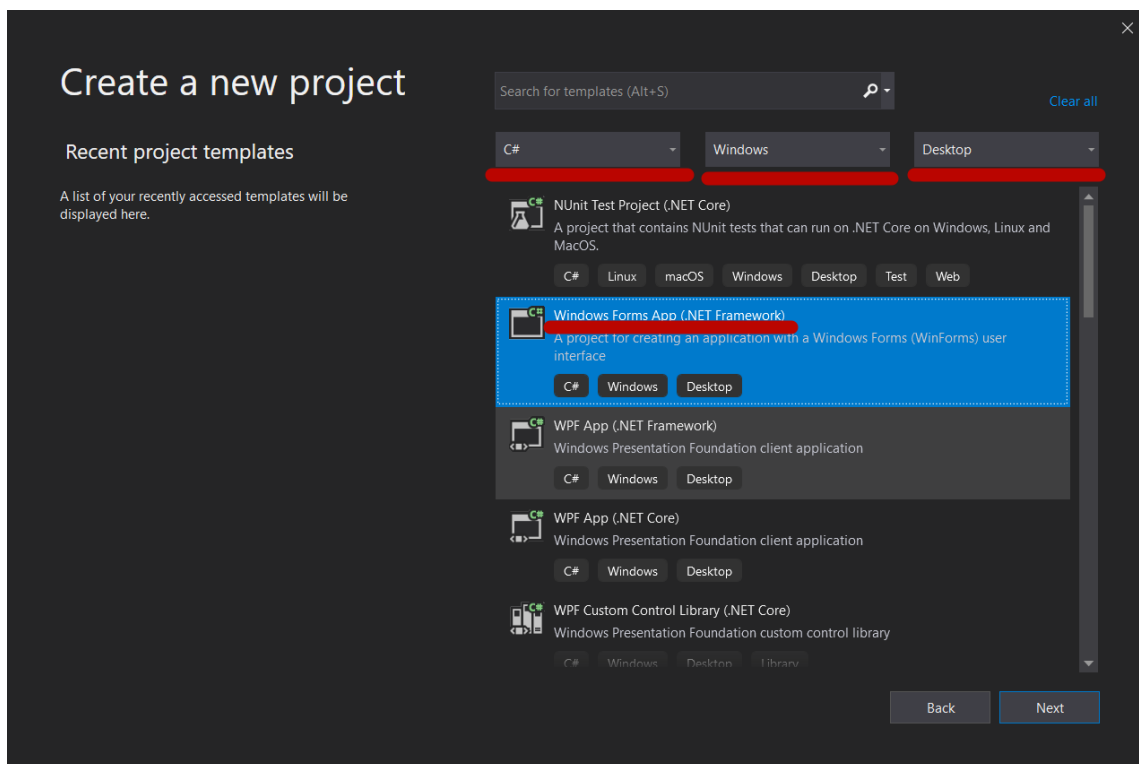


Рисунок 2.1.2 – Вибір типу проекту для створення (VS 2019)

2.3.1.2 Створення підключення до бази даних MS Access

Спочатку роботи ми підключаємося до бази даних MS Access за технологією ADO.NET. Пропонуємо Вам підключитися до бази даних MS Access, яку Ви створили під час першої лабораторної роботи. Ми ж продемонструємо можливість з'єднання на прикладі бази даних MS Access «*Общежитие*» (бажаючи взяти файл бази даних на dl.nure.ua з англomовними назвами таблиць та атрибутів).

Після створення проекту клацніть правою кнопкою миші по формі і натисніть пункт **Properties**. Задайте в поля **Name** і **Text** значення «**MainForm**». Для підключення бази даних за технологією ADO.NET на панелі елементів **Toolbox** виберіть елемент **bindingSource** і перетягніть його на форму. Відкрийте властивості **bindingSource1** і клацніть по полю **DataSource**. У списку, що впливає, виберіть **Add Project Data Source ...**. У новому діалоговому вікні (див. рис. 2.2) виберіть тип джерела даних **Database** і натисніть кнопку **Next**. Виберіть тип моделі **Dataset** і знову натисніть Next. На наступному етапі натисніть кнопку «**New Connection**» і в новому діалоговому вікні (див. рис. 2.3) виберіть шлях до вашої бази даних і натисніть кнопку **OK**.

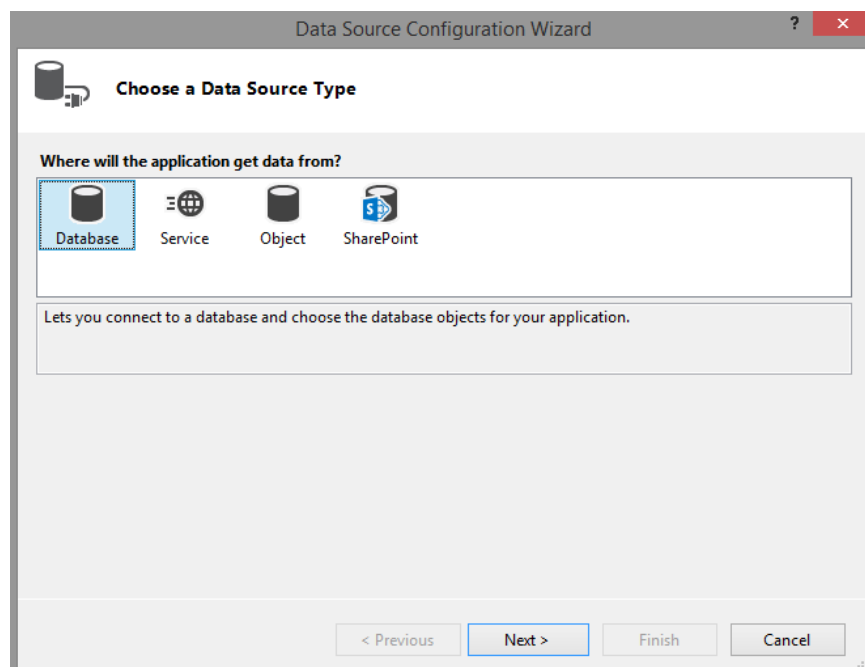


Рисунок 2.2 – Вибір джерела даних

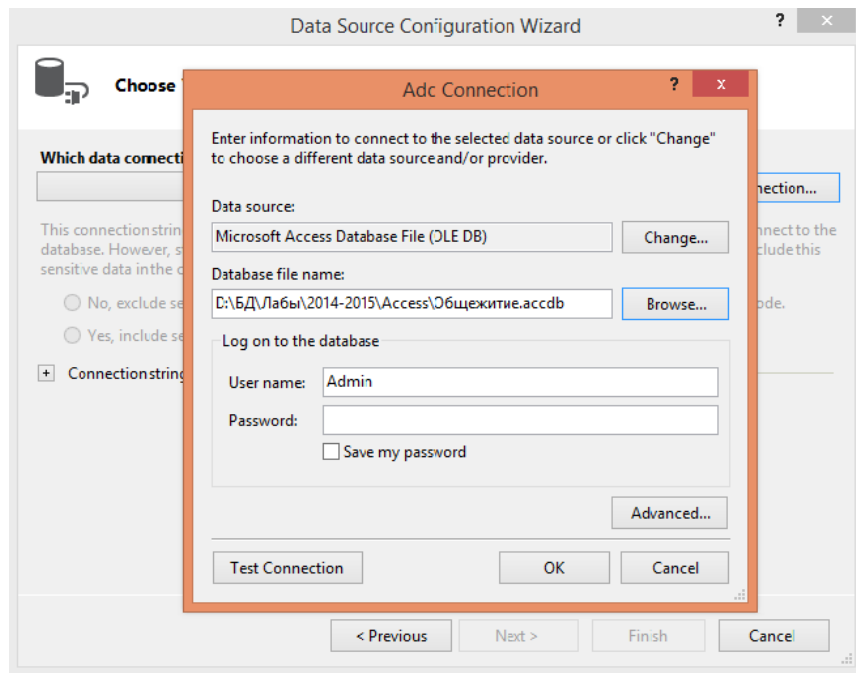


Рисунок 2.3 – Створення з'єднання

Після цього натисніть кнопку «*Next*». Після натискання відобразиться діалогове вікно, в якому буде питатися, чи хочете скопіювати файл (вашу базу даних) в ваш проект і змінити з'єднання. Слід вибрати кнопку «Так» (див. рис. 2.4).

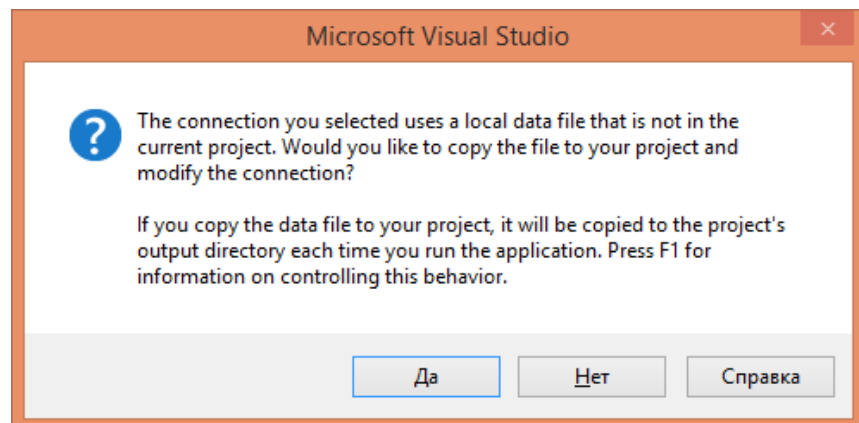


Рисунок 2.4 – Підтвердження створення локальної бази даних

Після цього у вікні створення підключення до бази даних необхідно вибрати елементи бази даних, які нам будуть потрібні в проекті. Отже, необхідно встановити прапорець на полі **Tables** і натиснути кнопку **Finish** (див. рис.2.5). Зверніть увагу, що після цього в Вашому проекті з'явиться елемент **ОбщезнаниеDataSet** (головний об'єкт за технологією ADO.NET),

який буди містити копію Вашої бази даних. Ви можете через меню, що впливає при натисканні правої кнопки миші, обрати **Конструктор** для DataSet і побачити схему Вашої БД.

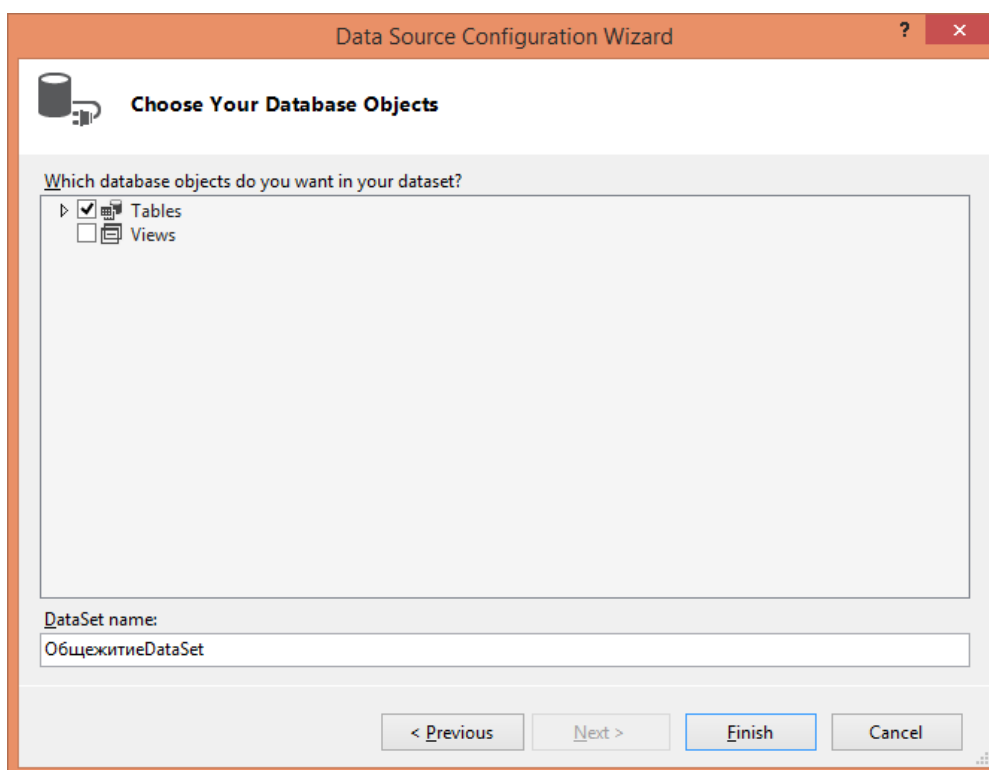


Рисунок 2.5- Завершення створення з'єднання

2.3.1.3 Відображення таблиці

Додайте на форму елемент **DataGridView**. У правому верхньому кутку елемента клацніть по білому квадрату з іконкою трикутника всередині і виберіть джерело даних для таблиці: **bindingSource1** → **Кімнати**. В результаті на формі повинна з'явитися таблиця, як на рисунку нижче (див. рис. 2.6).

2.3.1.4 Додавання навігації по таблиці

Виберіть елемент **bindingNavigator** та перетягніть його на форму. Перейдіть в властивості елемента **bindingNavigator1**. Для цього елемента виберіть значення **None** для властивості **Dock**. У властивості **BindingSource** виберіть джерелом даних **комнатыBindingSource**. Тепер таблиця має свою навігацію.

MainForm

	№ комнаты	Стоимость	Количество	Инвентарь	Этаж
*					

Рисунок 2.6 – Таблица Кімнати

Аналогічно додайте і налаштуйте елементи *DataGridView* і *bindingNavigator* для таблиць *Студенти* та *Пільги* (див рис. 2.7).

MainForm

File

	№ комнаты	Стоимость	Количество	Инвентарь	Этаж
*					

	Код студента	ФИО	Год рождения	Пол	Адрес	Группа
*						

	Код льготы	Тип льготы	Скидка
*			

Рисунок 2.7- Головна форма

2.3.1.5 Додання меню форми

Виберіть в панелі **Toolbox** елемент **MenuStrip** і перетягніть його на форму. Додайте в меню пункт **File** → **Exit**.

Після цього клацніть двічі по елементу **File** → **Exit**. Додайте наступний код:

```
private void exitToolStripMenuItem_Click(object sender, EventArgs e)
{
    Application.Exit();
}
```

Тепер під час роботи з додатком при натисканні на пункт меню **Exit** буде здійснюватися вихід з програми.

Клацніть по формі і в вікні **Properties** перейдіть до події **FormClosing**. Клацніть двічі по цій події і додайте наступний код:

```
private void MainForm_FormClosing(object sender, FormClosingEventArgs e)
{
    комнатыTableAdapter.Update(общезитиеDataSet1);
    льготыTableAdapter.Update(общезитиеDataSet1);
    студентыTableAdapter.Update(общезитиеDataSet1);
}
```

Клацніть лівою кнопкою миші по базі даних **Общезимие.accdb** (або з розширенням **mdb** для більш ранніх версій Access) в **Solution Explorer** і вікні **Properties** перейдіть до властивості **Copy_To_Output_Directory**. Встановіть в цій якості значення **Copy if newer**.

Кінцевий вигляд форми показаний на рисунку 2.7. Відкрийте програму, протестуйте. Покажіть його роботу викладачеві.

2.3.1.6 Створення бази даних для MS SQL Server

2.3.1.6.1 Створення БД та імпорт існуючих даних з MS Access файлу

Зверніть увагу, що працювати з російськомовними назвами таблиць (а в подальшому і полів) при написанні програмних кодів незручно. Крім того, в подальшому можуть виникнути і більш складні проблеми з російськомовними назвами. Тому рекомендується під час написання серйозних додатків використовувати бази даних з англomовними назвами таблиць і полів. Пізніше в нас буде можливість їх перейменувати.

Для створення нової бази даних для СУБД MS SQL, необхідно мати встановлені MS SQL Server Management Studio (SSMS) та MS SQL Server. Необхідно відкрити SSMS. Далі у вікні підключення до серверу, необхідно вказати ім'я серверу, до якого буде здійснюватися підключення (значення за замовчуванням – ім'я комп'ютеру, можна переглянути: натиснути **Win+X**, обрати **Система**, у полі **Ім'я пристрою** буде ім'я комп'ютеру, яке нам необхідне). Приклад вікна з інформацією наведено на рис 2.8.

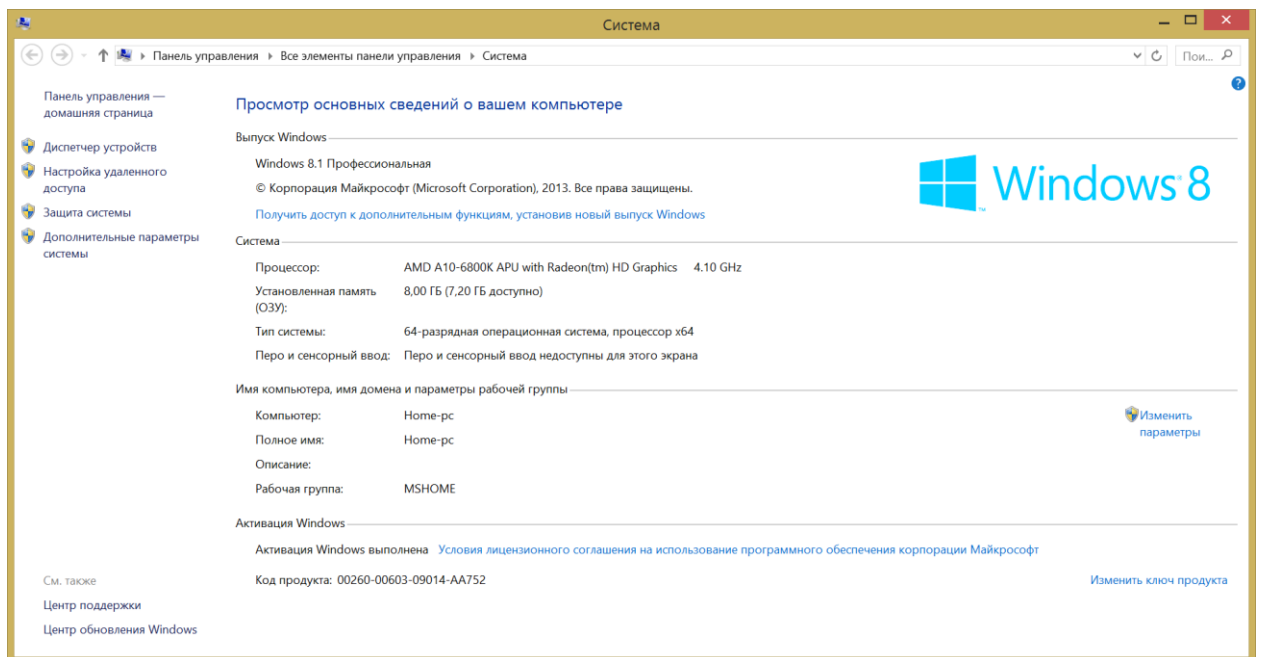


Рисунок 2.8 – Інформація про комп'ютер

Далі у вікні **Object Explorer** натиснути правою кнопкою на папці **Databases** та обрати **New Database**.

У вікні створення БД, дати ім'я базі даних та натиснути **ОК**. Вікно створення БД наведено на рис. 2.9.

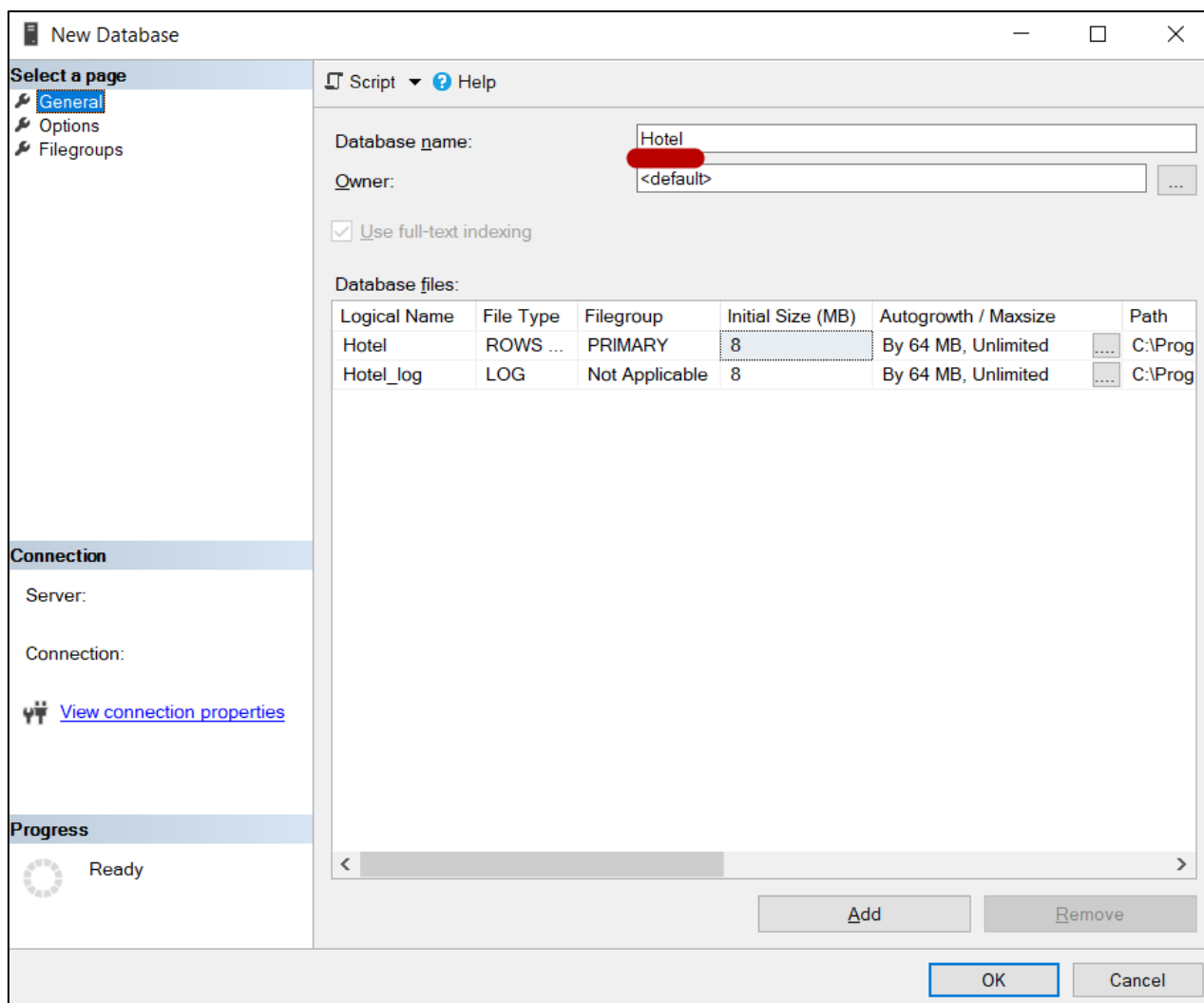


Рисунок 2.9 – Вікно створення БД

Далі нам необхідно імпортувати дані з файлу MS Access до нашої БД. Для цього необхідно натиснути правою клавішею миші по створеній БД та обрати **Tasks -> Import Data**. Візуальні інструкції наведено на рис. 2.10.

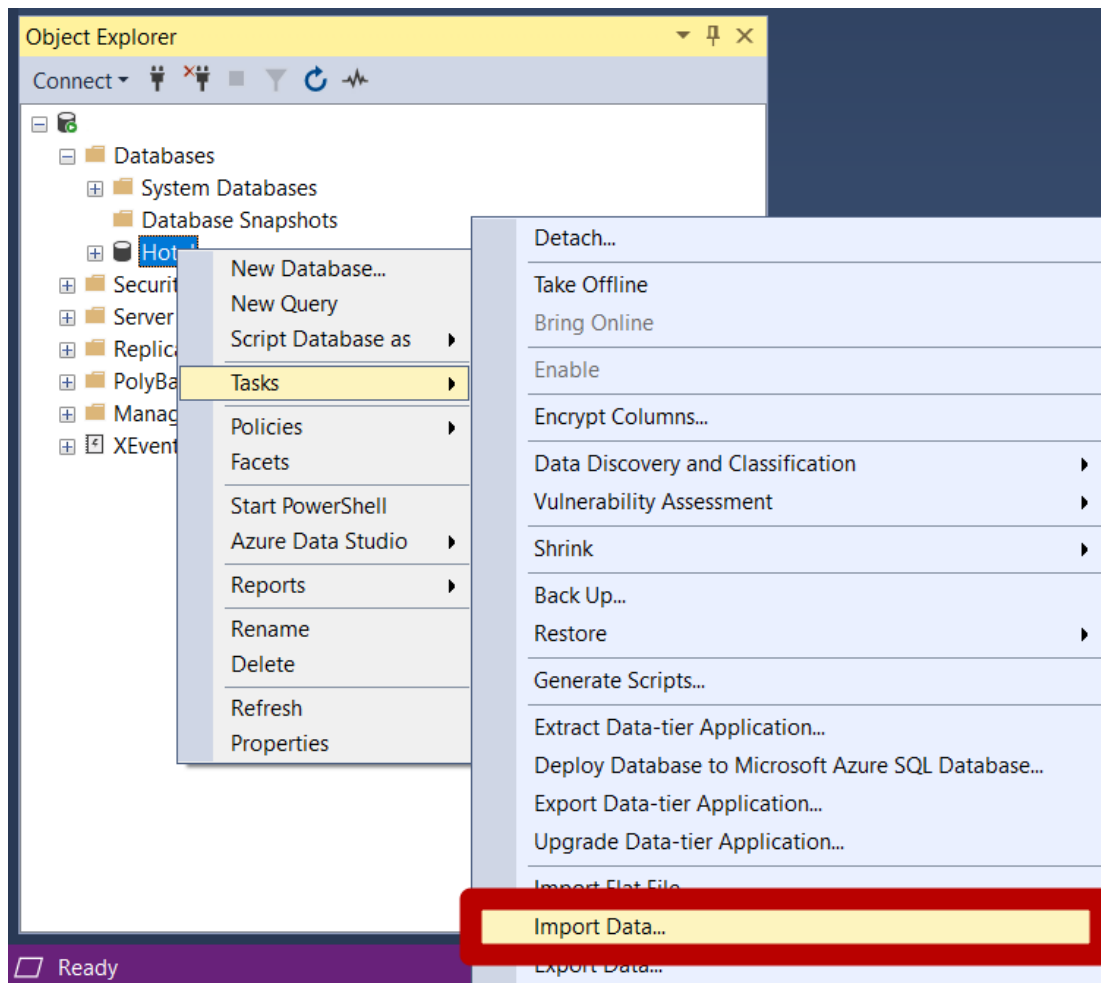


Рисунок 2.10 – Імпорт даних у БД

Далі у вікні імпорту необхідно вказати **Microsoft Access** у полі **Data source** та вказати **User name** як **Admin**. Натиснути **Next**. Потім у вікні вказати створену БД як показано на рис. 2.11. Натиснути **Next**.

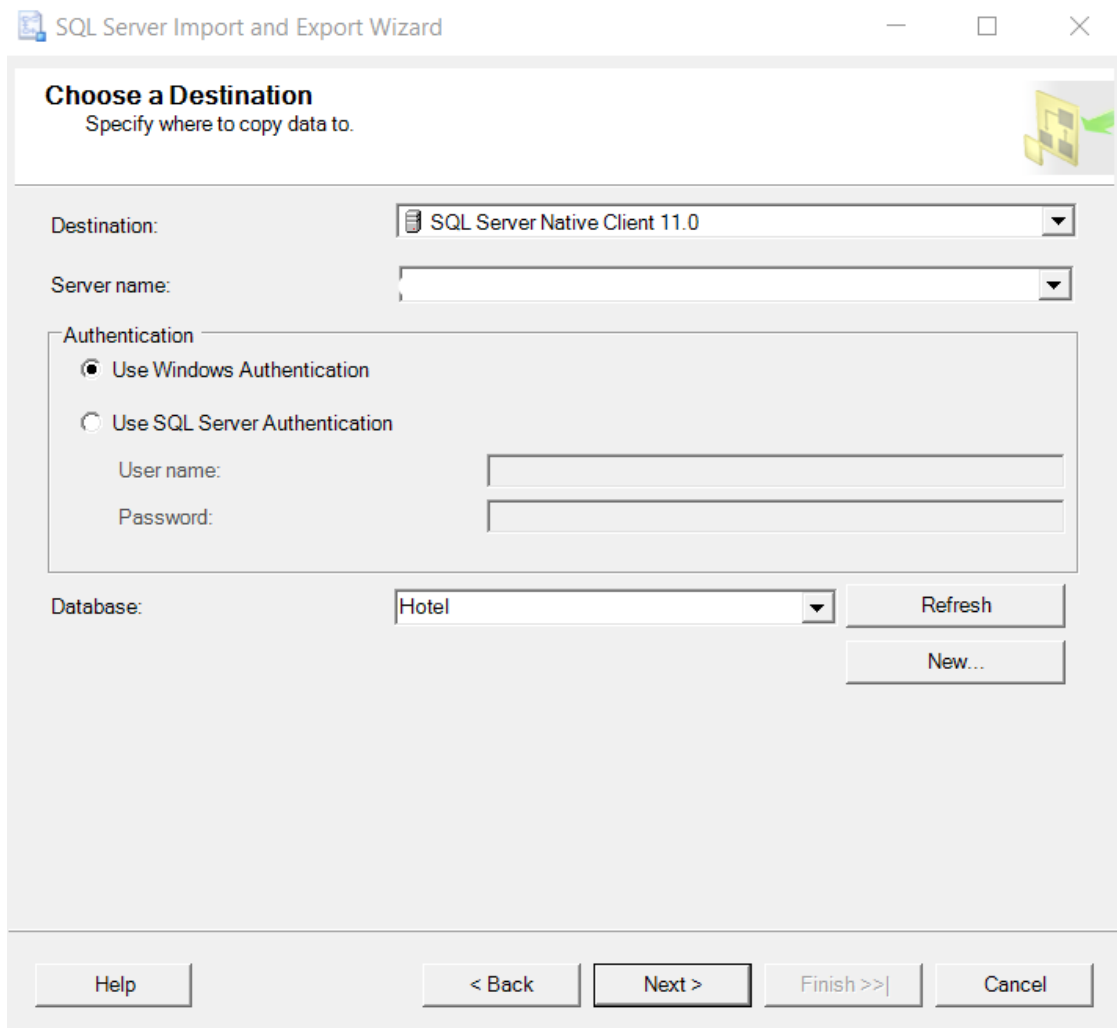


Рисунок 2.11 – Вікно призначення імпортованих даних

Вибрати опцію **копіювати усі дані** та натиснути **Next**. Далі у вікні вибору таблиць, обрати необхідні для імпорту (у нашому випадку – усі таблиці). Далі натискати **Next** до моменту, коли кнопка **Finish** стане доступною, натиснути її. Після цього усі дані буде імпортовано до новоствореної БД.

Далі необхідно додатково заповнити таблиці даними. Для цього в **Server Explorer** виберіть **Data Connections** → **Назва вашого Data Source для SQL Server** → **Tables**, виберіть таблицю **Rooms**, натисніть правою кнопкою миші по ній, виберіть **Show Table Data** (див. рис. 2.12) і додайте три-чотири кімнати (див. рис. 2.13). Аналогічним чином заповніть таблицю **Benefits** (див. рис. 2.14) і таблицю **Students** (див. рис. 2.15). Зверніть увагу, що поля, які є зовнішніми ключами, можуть вказувати тільки на існуючі поля у відповідних таблицях (обмеження посилальної цілісності).

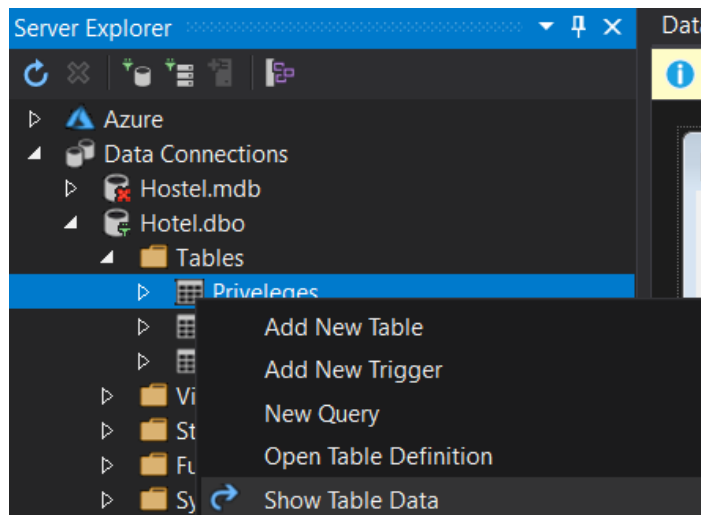


Рисунок 2.12 – Місцезнаходження кнопки Show Table Data

	Id	Price	Amount	Floor	Tools
▶	1	1200,0000	4	1	телевизор
	2	1000,0000	3	2	холодильник
	3	1500,0000	4	2	тел+хол

Рисунок 2.13 – Кімнати

	Id	Type	Discount
▶	1	Отсутствует	0
	2	Сирота	100
	3	Инвалидность	60

Рисунок 2.14 – Пільги

	Id	Name	Gender	Address	Group	Passport	BenefitCode	RoomNumber	ColonizeDate
▶	1	Иванов И.И.	мужской	ул. Бакулина, 1	ПИ-14-1	AA123454	1	1	30.08.2013
	2	Петров П.П.	мужской	ул. Бакулина, 1	ПИ-14-2	BB223333	2	1	28.08.2014
	3	Андреева О.О.	женский	ул.Целиноград...	ПИ-14-3	PP464545	3	2	23.08.2013
	4	Сергеева С.С.	женской	ул. Целиногра...	ПИ-14-5	LL345345	1	2	24.08.2013
	5	Сидоров В.В.	мужской	пр. Ленина, 14	ПИ-14-6	KK332433	1	3	26.09.2014

Рисунок 2.15 – Студенти

2.3.1.6.2 Створення таблиць власноруч

Додатково, існує можливість створення таблиць власноруч. Для цього необхідно створити БД у SSMS (див. попередній пункт). Потім, необхідно створити таблицю. Для цього необхідно створити скрипт. Для того, щоб відкрити пустий скрипт, необхідно натиснути правою кнопкою миші на створеній БД та вибрати пункт *New script*. (див. рис. 2.16).

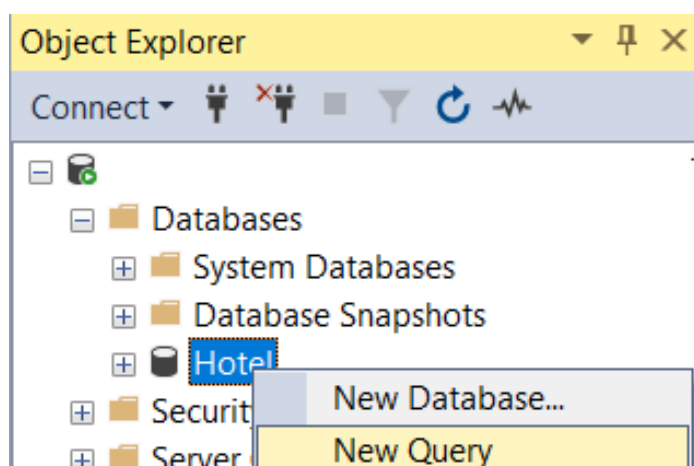


Рисунок 2.16 – Створення скрипту до БД

У відкритому вікні необхідно ввести скрипт для створення БД на мові T-SQL. На рисунку 2.17 зображено зразок скрипту для створення необхідних таблиць. Для того, щоб виконати скрипт, необхідно натиснути клавішу **F5** або натиснути правою кнопкою мишу по скрипту та натиснути *Execute (F5)*.

Розберемо ключові моменти скрипту, наведеного на рисунку 2.17. Інструкція **USE** вказує на те, яку БД використовувати для подальших команд. Потім використовується команда **CREATE TABLE** для творення таблиці із схемою прав доступу **dbo**. Потім у дужках через кому вказуються параметри полів таблиці, а саме ім'я поля, його тип та спеціальні модифікатори. Після дужок вказується ім'я файлової групи, у якій буде створена таблиця, якщо немає спеціальних вимог до місця збереження таблиці – вказувати **ON [PRIMARY]**.

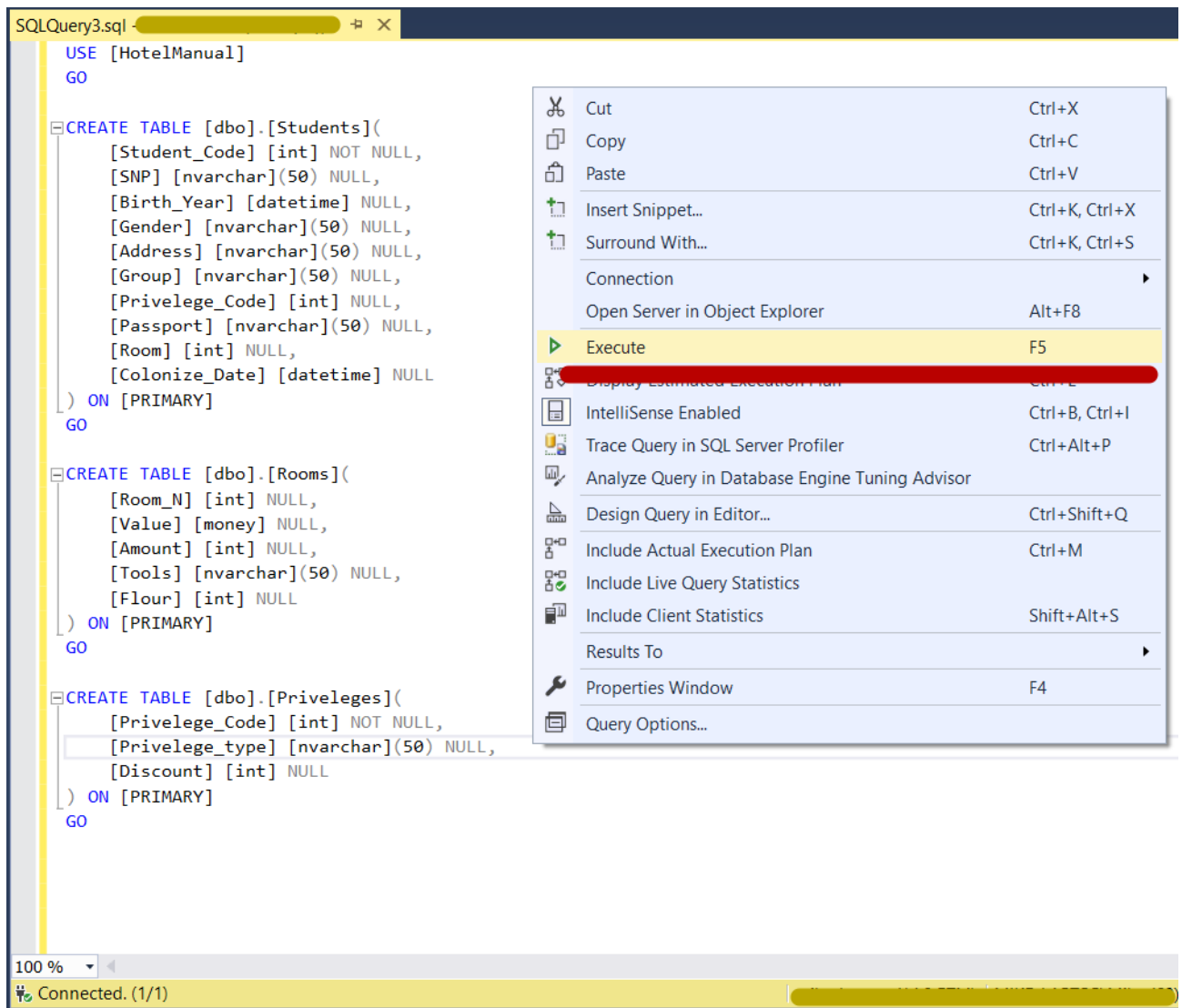


Рисунок 2.17 – Вікно редактору скрипту. Інструкція до того, як запустити скрипт

Приклади команд, за допомогою яких Ви можете створити таблиці:

```
CREATE TABLE priveleges (
    privelege_id int NOT NULL,
    privelege_type varchar(50) NULL,
    discount int NULL,
    PRIMARY KEY (privelege_id) );
```

```
CREATE TABLE rooms (
    room_num int NULL,
    value decimal NULL,
    amount int NULL,
    tools varchar(50) NULL,
    flour int NULL,
    PRIMARY KEY (room_num) );
```

```
CREATE TABLE students (
    student_id int NOT NULL,
    SNP varchar(50) NULL,
    birth_year int NULL,
    gender varchar(50) NULL,
    address varchar(50) NULL,
    group_name varchar(50) NULL,
    passport varchar(50) NULL,
    colonize_date datetime NULL,
    room_num int NULL REFERENCES Rooms(Room_N),
    privelege_id int NULL REFERENCES Priveleges(privelege_id),
    PRIMARY KEY (student_id) );
```

2.3.1.6.3 Копіювання та відновлення БД

Для того, щоб БД можна було перенести з одного комп'ютеру на інший, необхідно натиснути ***Generate Scripts*** (див рис. 2.18):

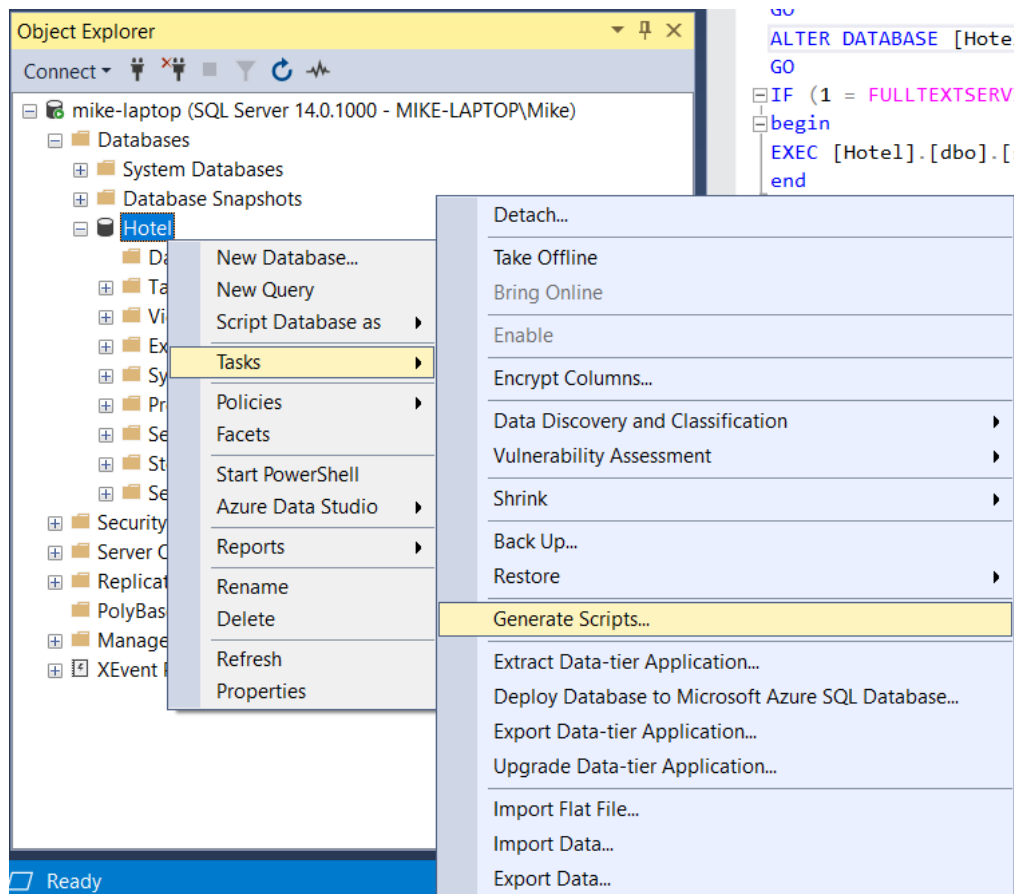


Рисунок 2.18 – Місцезнаходження кнопки Generate Scripts

Далі необхідно дотримуватися усіх вимог у відкритому вікні, за виключенням особливого налаштування: на екрані *Set scripting options* натиснути кнопку *Advanced* та обрати у опції *Types of data to script* значення *Schema and data*. Вікно із налаштуваннями наведено на рис. 2.19.

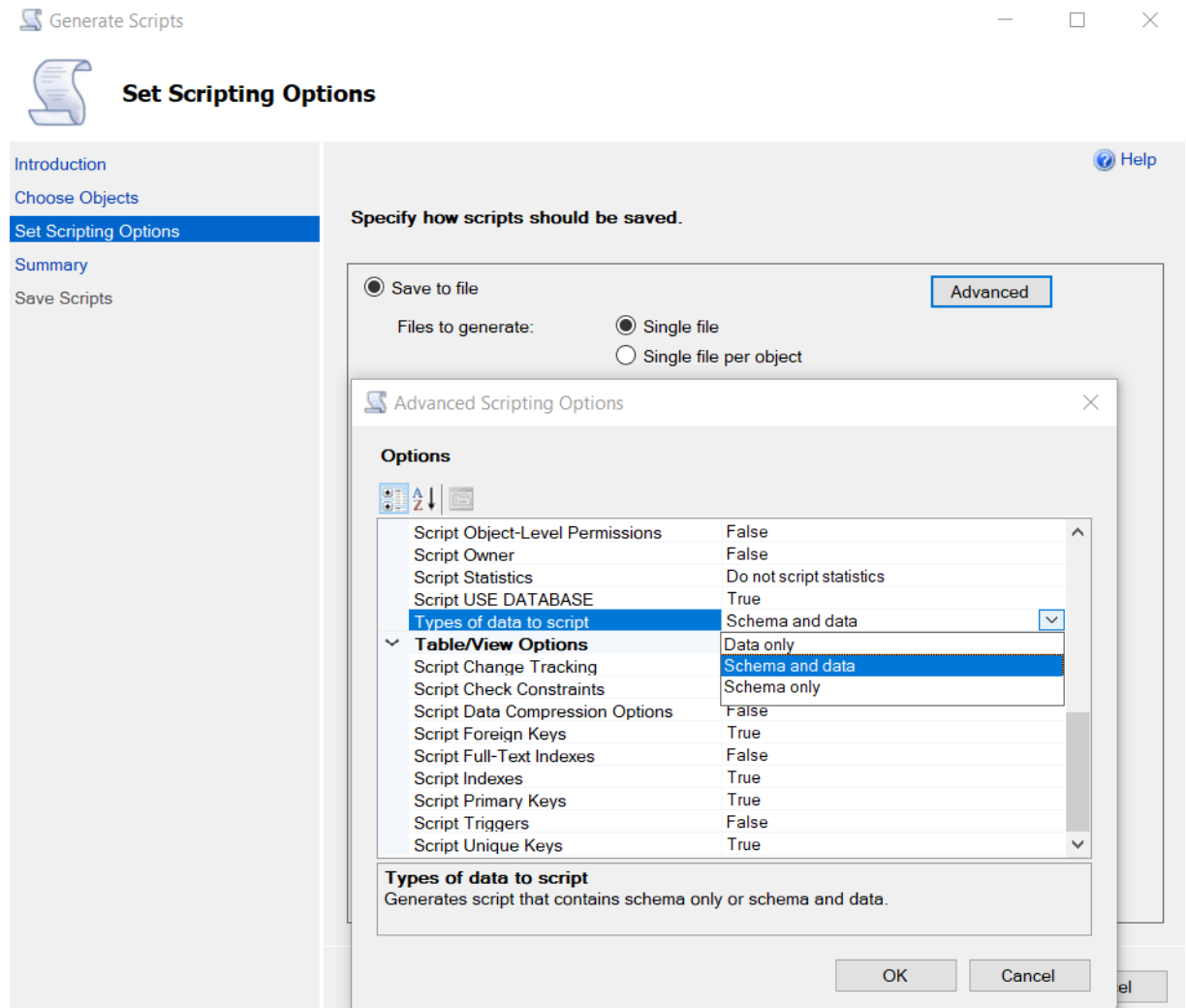


Рисунок 2.19 – Налаштування *Generate script*

Тепер для того, щоб розгорнути цю базу даних на іншому комп'ютері, достатньо відкрити створений *script.sql* у SSMS, вибрати *master* (див. рис. 2.20) та натиснути *Execute*. Після цього на іншому комп'ютері створиться повна точна копія бази даних.

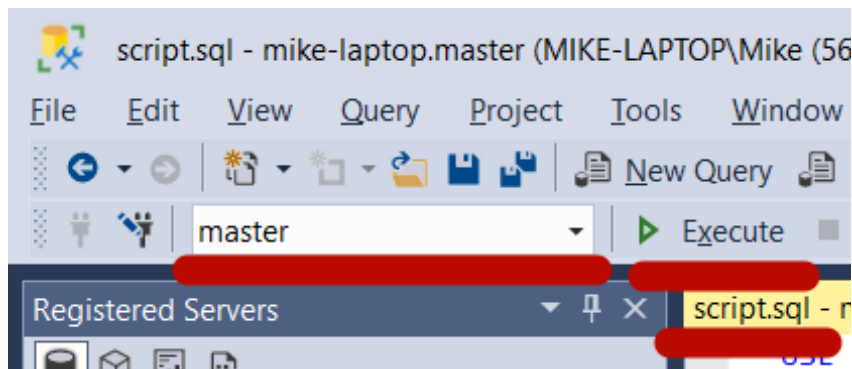


Рисунок 2.20 – Виконання скрипту для відновлення БД

2.3.1.6.4 Перейменування таблиць у SSMS

Далі необхідно перейменувати таблиці, щоб їх ім'я були написані англійською мовою. Рекомендовані імена: ***Rooms, Benefits, Students.***

2.3.1.6.3 Створення підключення до бази даних

Для підключення створеної бази даних до існуючого додатку, необхідно проробити кроки, вказані у пункті 2.3.2, за виключенням того, що тепер нам потрібно створити зв'язок з провайдером даних ***MS SQL Server*** та вказати назву новоствореної БД. Налаштування наведено на рис 2.19.

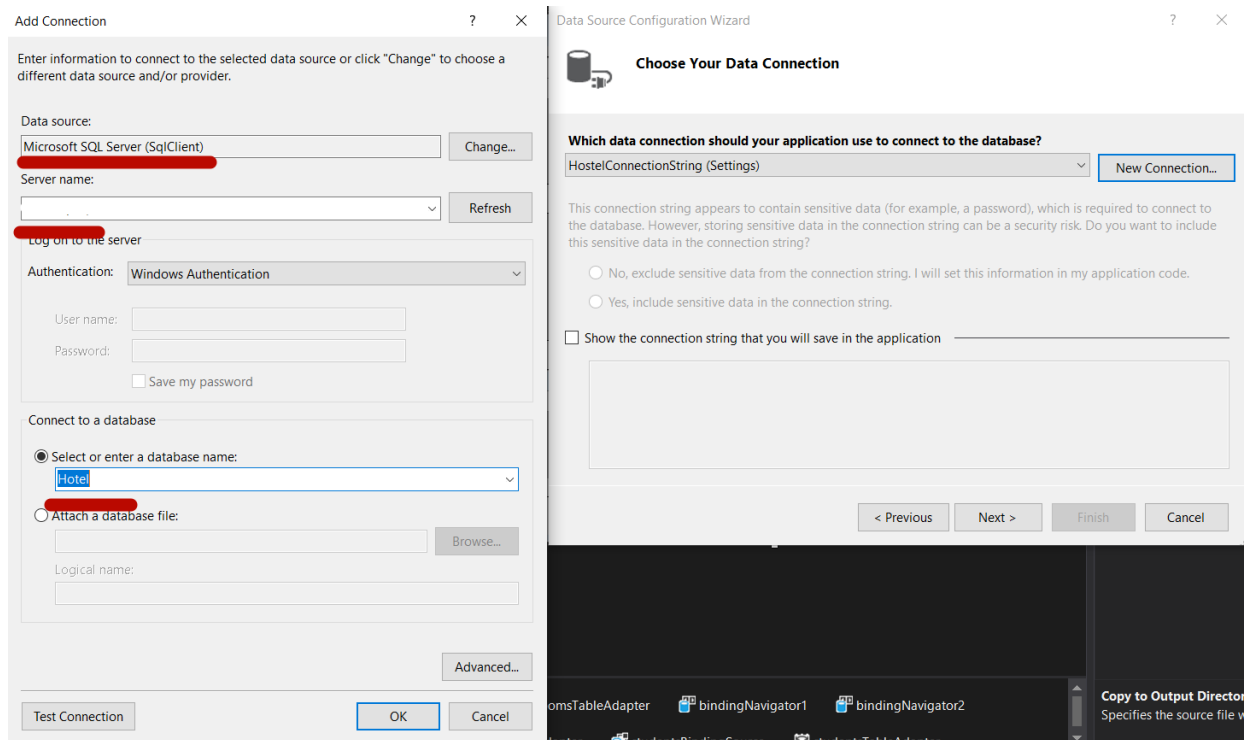


Рисунок 2.19 – Налаштування зв'язку з БД MS SQL Server

Далі, для того, щоб зв'язати новий Data Source з елементами інтерфейсу, необхідно у *bindingSource1* змінити властивість *Data Source* і обрати новостворений *Data Source*.

Тепер можете налаштувати через компонент DataSet зовнішні ключі в його дизайнері.

Потім самостійно налаштуйте візуальні компоненти DataGridView і bindingNavigator для всіх трьох таблиць і доведіть програму до працездатності. Тобто необхідно виконати нехитрі дії з налаштування цих компонентів заново, використовуючи компоненти DataGridView. Не забудьте також:

- правильно налаштувати компоненти bindingNavigator;
- переписати обробник події FormClosing:

```
private void MainForm_FormClosing(object sender, FormClosingEventArgs e) {
    roomsTableAdapter.Update(hostelDataSet);
    benefitsTableAdapter.Update(hostelDataSet);
    studentsTableAdapter.Update(hostelDataSet);
}
```

- Відкрийте програму і покажіть її роботу викладачеві.

2.3.1.7 Створення бази даних для MySQL

Для початку, необхідно встановити коннектор, який можна знайти за посиланням <https://downloads.mysql.com/archives/c-odbc/>.

Після встановлення коннектора, необхідно відкрити у системі Windows **Панель управління => Система та безпека => Адміністрування => Джерела даних (ODBC)**. Якщо опцій декілька, обрати версію з (x32). Необхідно натиснута кнопку **Додати** на вкладці **Користувацькі DSN**. У списку, що з'явився вибрати **My SQL ODBC X.X Unicode Driver** та натиснути **Закінчити**.

Повинно з'явитися вікно конфігурації My SQL Connector. Необхідно вказати **Data Source Name => My MySQL Data Source, TCP/IP Server => 127.0.0.1, User => root**. Натиснути **OK**.

Потім необхідно запустити OpenServer. У системному треї (справа знизу на екрані) з'явиться позначка прапорця. Необхідно натиснути правою кнопкою миші на прапорець, натиснути **Run Server**.

Коли прапорець стане зеленим, необхідно знову натиснути правою кнопкою миші на прапорець та вибрати **Advanced => MySQL Manager**.

У відкритому вікні натиснути кнопку **New** у лівому нижньому кутку вікна, потім натиснути **Open**.

Після цього, повинно відкритися вікно, зовнішній вигляд якого зображено на рис. 2.20.

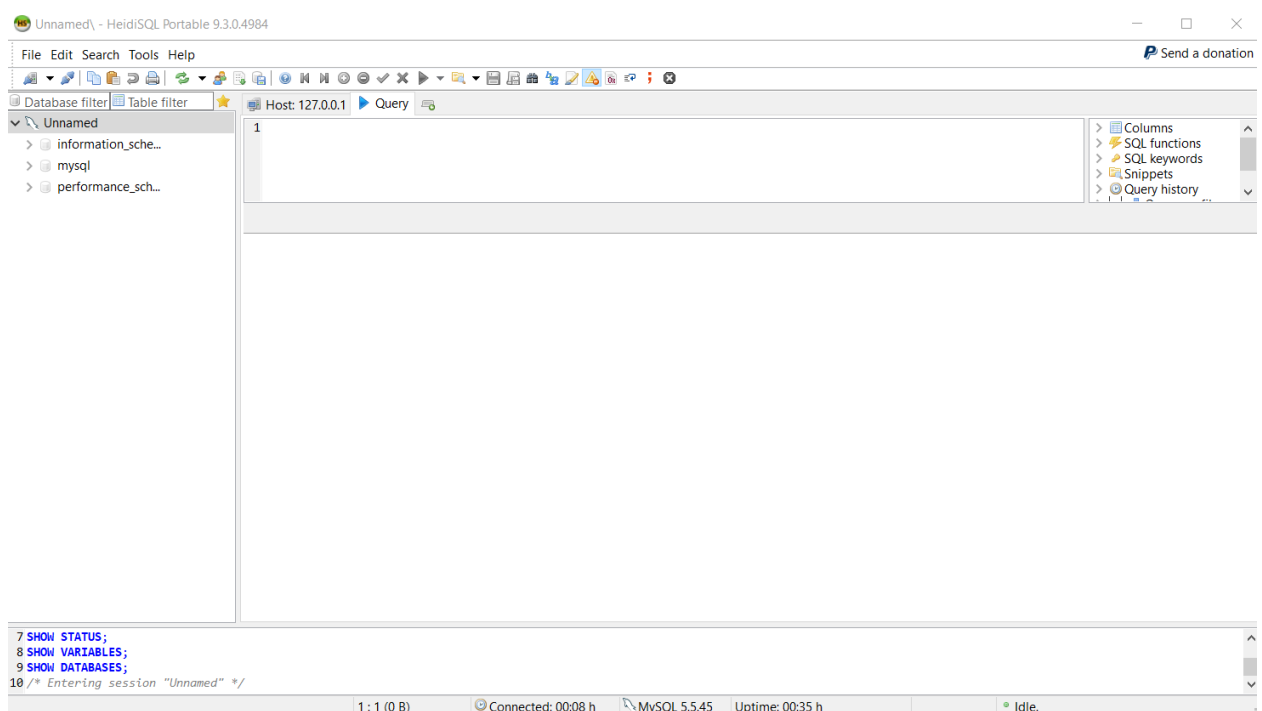


Рисунок 2.20 – Вікно HeidiSQL

Після цього, необхідно створити базу даних та створити таблиці з тими ж самими типами, що і у MS SQL Server. Приклад створеної БД та таблиць наведено на рис. 2.21 (виділено створені таблиці, вкладка **Data** для заповнення даними вибрану таблицю та створені поля таблиці)

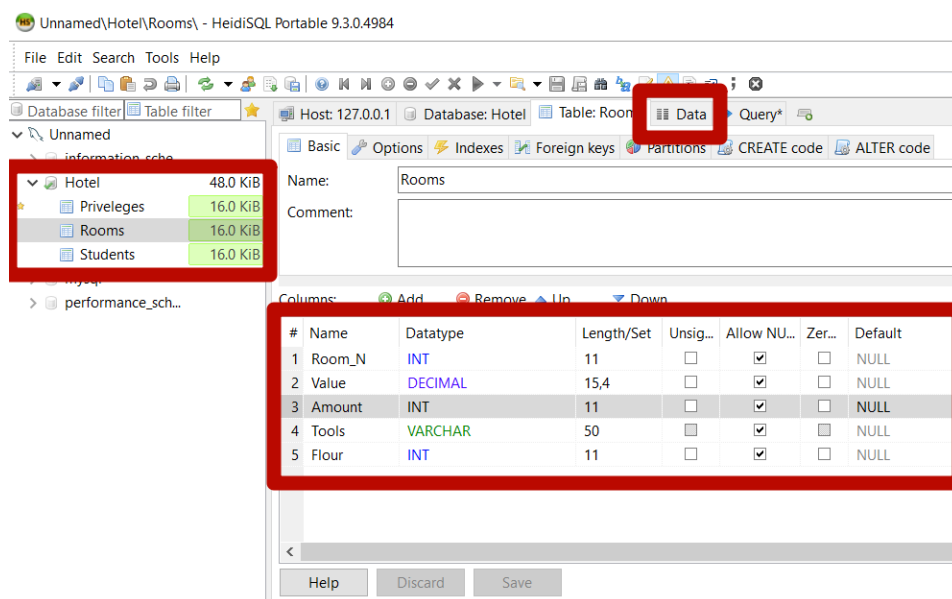
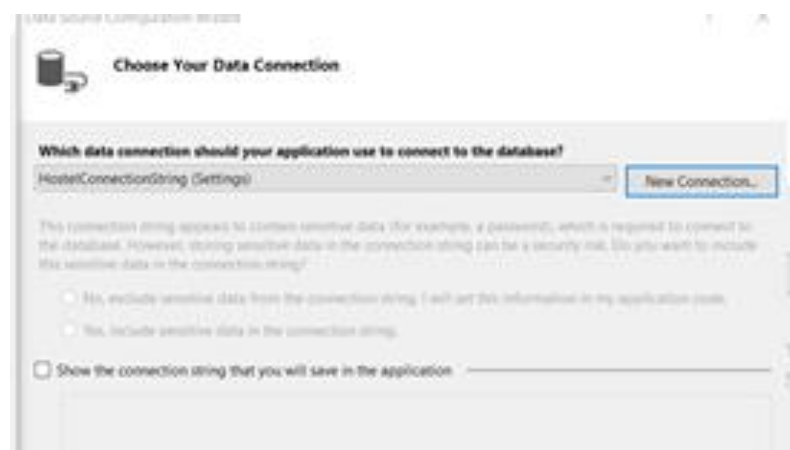
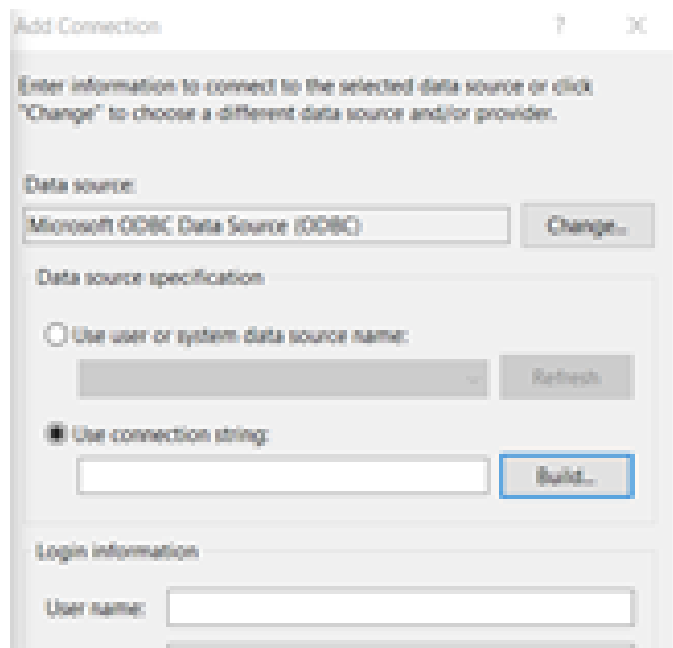
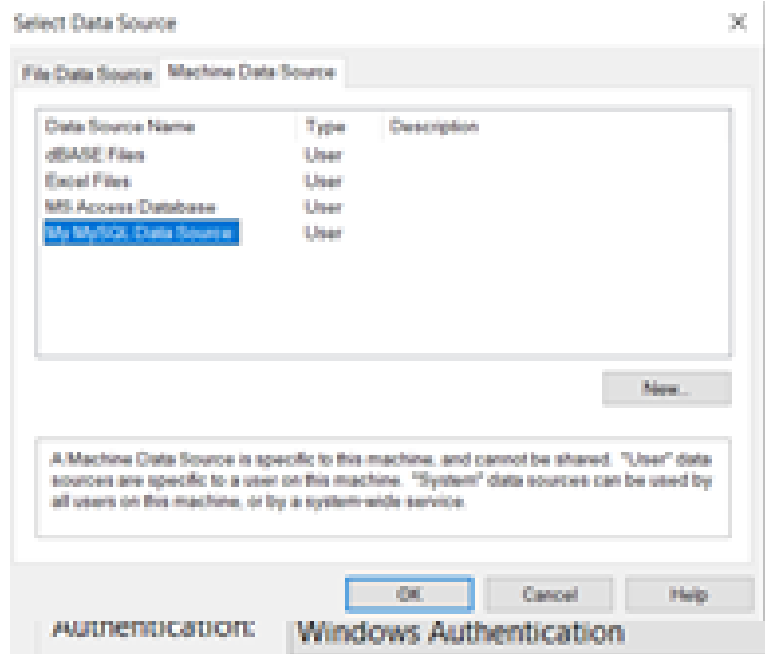


Рисунок 2.21 – Створені БД та таблиці у вікні HeidiSQL

2.3.1.8 Створення підключення до бази даних MySQL

Для підключення створеної бази даних до існуючого додатку, необхідно проробити кроки, вказані у пункті 2.3.2, за виключенням того, що необхідно обрати **Data Source** => **Microsoft ODBC Data Source (ODBC)**, потім необхідно змінити тип **Data Source specification** на **Use connection string** та натиснути кнопку **Build**, вибрати створений у Панелі Управління **Data Source** (див рис. 2.24) та натиснути **OK**.



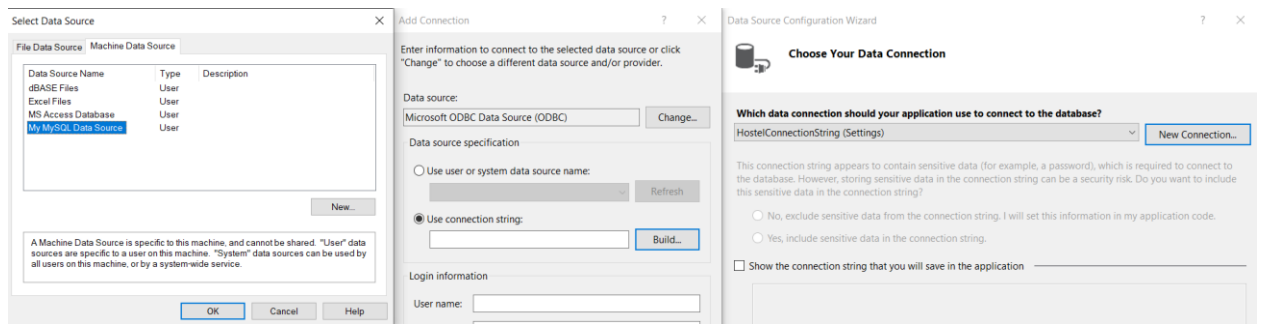


Рисунок 2.24 – Додання підключення до БД через ODBC

Після цього з'явиться вікно MySQL Connector, у якому вже будуть задані нами параметри, необхідно вибрати створену БД та натиснути **OK** (див рис. 2.25).

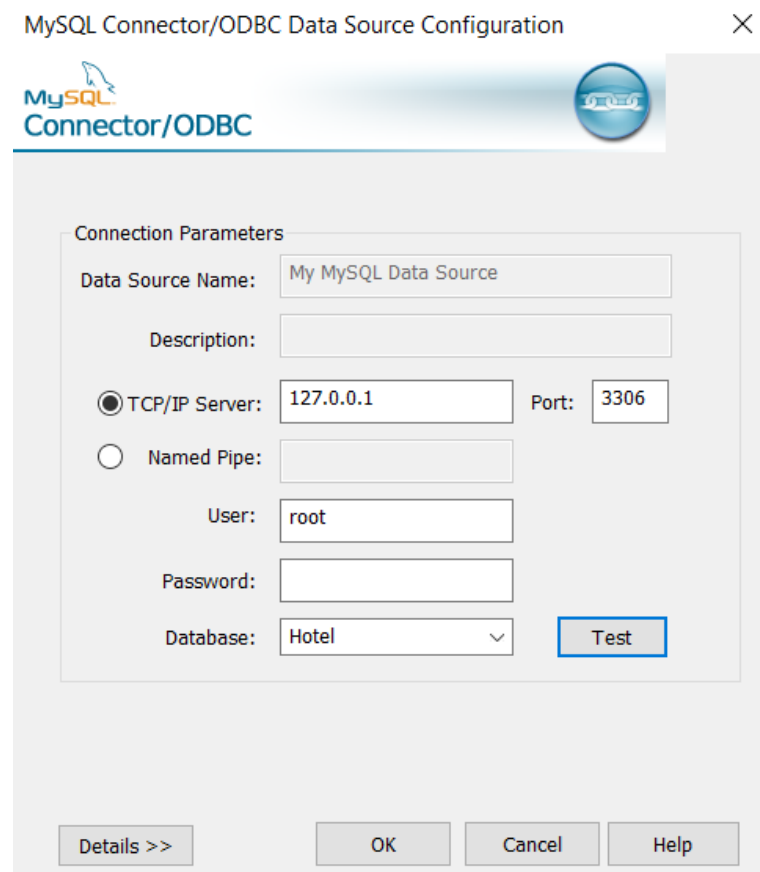


Рисунок 2.25 – Вікно коннектора

Потім необхідно вибрати створені таблиці та завершити налаштування нового з'єднання.

Потім необхідно повторити кроки по зміні **Data Source**. Далі, для того, щоб зв'язати новий Data Source з елементами інтерфейсу, необхідно у

bindingSource1 змінити властивість ***Data Source*** і обрати новостворений ***Data Source***.

Тепер можете налаштувати через компонент **DataSet** зовнішні ключі в його дизайнері.

Потім самостійно налаштуйте візуальні компоненти **DataGridView** і **bindingNavigator** для всіх трьох таблиць і доведіть програму до працездатності. Тобто необхідно виконати нехитрі дії з налаштування цих компонентів заново, використовуючи компоненти **DataGridView**. Не забудьте також:

- правильно налаштувати компоненти **bindingNavigator**;
- переписати обробник події **FormClosing**:

```
private void MainForm_FormClosing(object sender, FormClosingEventArgs e) {  
    roomsTableAdapter.Update(hostelDataSet);  
    benefitsTableAdapter.Update(hostelDataSet);  
    studentsTableAdapter.Update(hostelDataSet);  
}
```

- Відкрийте програму і покажіть її роботу викладачеві.

2.3.1.9 Додавання меню форми

Продовжимо роботу над додатком.

Додайте в меню поля **Database → Rooms**, **Database → Students** і **Database → Benefits**.

Додайте на форму новий елемент **Label** та видаліть два компонента **DataGridView** і два **BindingNavigator**. Задайте в властивості **Text** елемента **label1** значення «**Rooms**». Клацніть двічі по пункту меню **Database → Rooms**, після чого ви перейдіть до коду даної функції.

Додайте наступний код:

```
private void roomsToolStripMenuItem_Click(object sender, EventArgs e)  
{  
    bindingNavigator1.BindingSource = roomsBindingSource;  
    dataGridView1.DataSource = roomsBindingSource;  
    label1.Text = "Rooms";  
}
```

Аналогічно додайте наступний код для пункту **Database → Students**:

```
private void studentsToolStripMenuItem_Click(object sender, EventArgs e)
{
    bindingNavigator1.BindingSource = studentsBindingSource;
    dataGridView1.DataSource = studentsBindingSource;
    label1.Text = "Students";
}
```

І для пункту меню **Database** → **Benefits**:

```
private void benefitsToolStripMenuItem_Click(object sender, EventArgs e)
{
    bindingNavigator1.BindingSource = benefitsBindingSource;
    dataGridView1.DataSource = benefitsBindingSource;
    label1.Text = "Benefits";
}
```

Клацніть правою кнопкою миші по формі і перейдіть до події **Load**. Клацніть двічі по цій події і додайте до вже написаних рядків коду в цій функції наступний рядок:

```
dataGridView1.AutoGenerateColumns = true;
```

Тепер ми можемо перемикатися між таблицями під час роботи додатку за допомогою пунктів меню. Кінцевий вигляд головної форми показаний на рисунку 2.24. Запустіть, протестуйте додаток.

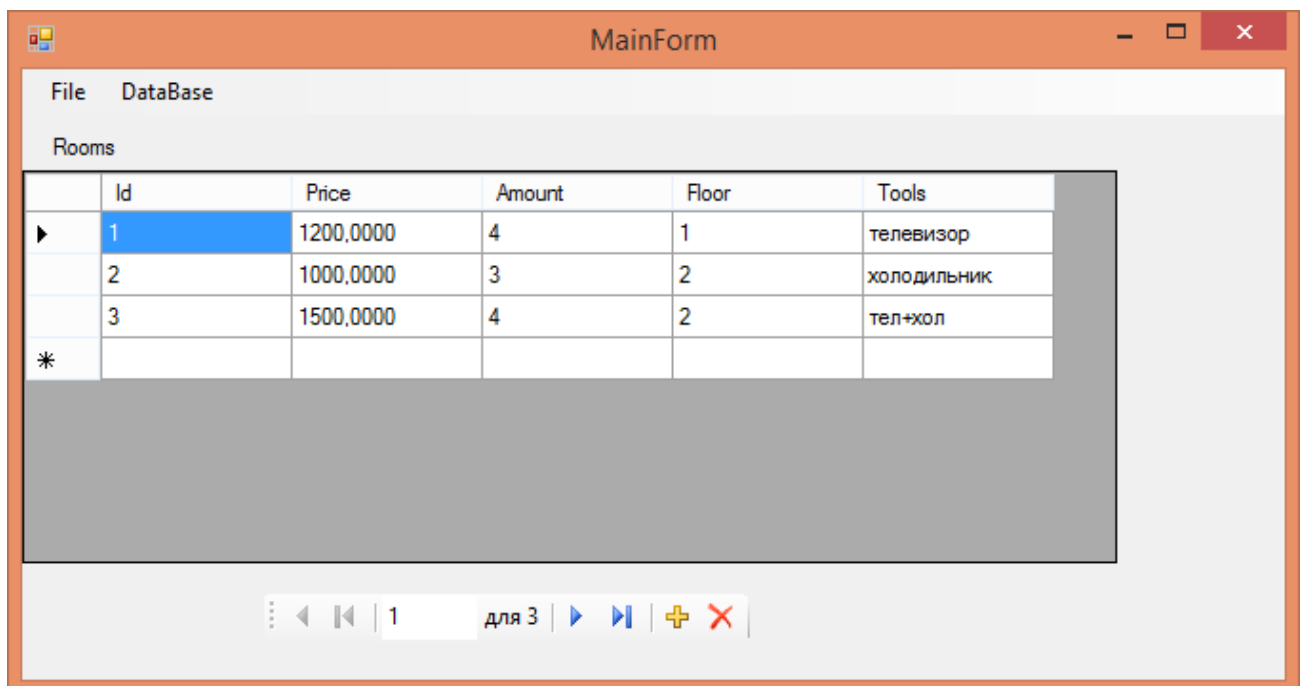


Рисунок 2.24 – Кінцевий вигляд форми

2.3.1.10 Додавання відображення пов'язаних таблиць

Додамо в проект нову форму, на якій покажемо розселення студентів по кімнатах. Через пункт **Project** → **Add New Item** → **Windows Form** додайте до проекту нову форму **RSForm**. Змініть значення властивості форми **Text** на **Розселення**. Виконайте команду **View** → **Other Windows** → **Data Sources**. У вікні **Data Sources** відображено джерело даних **HostelDataSet**. Клацніть на стрілці біля таблиці **Rooms** і виберіть у списку, що випадає, вид відображення **Details** (див. рис.2.25).

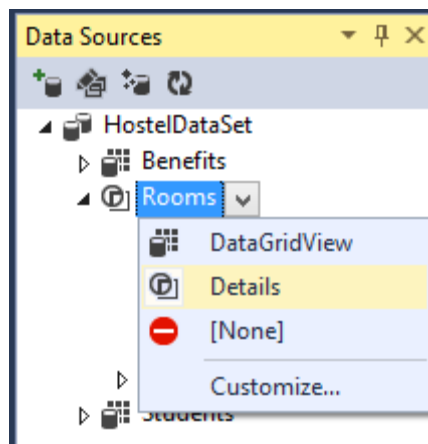


Рисунок 2.25 – Джерело даних HostelDataSet

Після цього, натисніть на назву таблиці **Rooms** лівою кнопкою миші, і не відпускаючи її, перетягніть на форму.

В результаті на формі з'являться засоби для відображення полів таблиці (5 компонентів **Label** і **TextBox**), засіб навігації по таблиці (**BindingNavigator**), а в нижній частині робочої області середовища розробки з'явилися п'ять невидимих об'єктів призначені для зв'язку форми з таблицею (див. рис 2.26) . Зверніть увагу, у вікні **Data Sources** під полями таблиці **Rooms** у вигляді підтаблиці розташована таблиця **Students**. Підтаблиця показує, що таблиця **Students** є вторинною / підпорядкованої / дочірньої по відношенню до таблиці **Rooms**.

Виведемо дані з таблиці студентів в табличному вигляді під полями таблиці кімнат. Для цього слід клацнути по підтаблиці **Students** під списком полів таблиці **Rooms**, вибрати в списку, що випадає, вид відображення

DataGridView та перенести підтаблицю на форму, помістивши внизу форми. На формі з'явиться елемент **studentsDataGridView** для відображення пов'язаних записів з таблиці **Students** і невидимі компоненти для зв'язку.

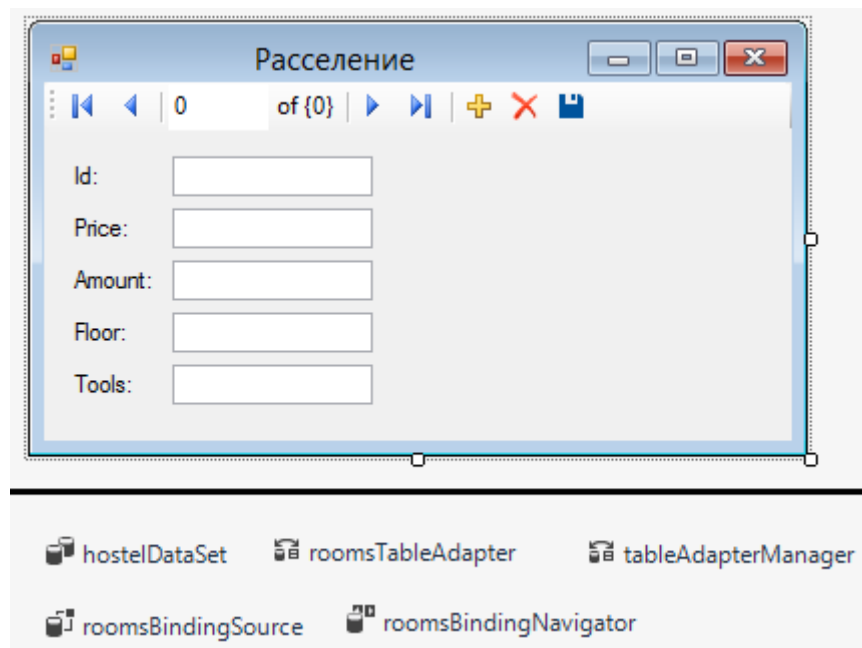


Рисунок 2.26 – Форма для відображення таблиці Rooms

Додайте кнопку (**Button**) на форму і змініть значення її властивості **Text** на «**Підтвердити зміни**». Клацніть на кнопці і шаблоні функції обробки її натискання, що з'явився, напишіть код:

```
private void button1_Click(object sender, EventArgs e)
{
    if (MessageBox.Show("Вы действительно хотите подтвердить изменения?", "Изменение данных", MessageBoxButtons.YesNo) == DialogResult.Yes)
    {
        roomsBindingSource.EndEdit();
        roomsTableAdapter.Update(hostelDataSet);
        studentsTableAdapter.Update(hostelDataSet);
    }
}
```

На головній формі слід додати пункт меню **View** і в нього підпункт **Resettlement**. Додайте код в обробник події натискання на пункт меню **Resettlement**:

```
private void resettlementToolStripMenuItem_Click(object sender, EventArgs e)
{
    var rs = new RSForm();
    rs.ShowDialog();
}
```

```

benefitsTableAdapter.Fill(hostelDataSet.Benefits);
studentsTableAdapter.Fill(hostelDataSet.Students);
roomsTableAdapter.Fill(hostelDataSet.Rooms);
}

```

Відкрийте програму. Тепер, якщо з головної форми перейти на форму **Розселення**, то можна побачити відображення кімнат і студентів, що саме живуть в них (див. рис. 2.27). Протестуйте роботу цієї форми і зміни даних на ній.

	Id	Name	Gender	Address	Group
▶	1	Иванов И.И.	мужской	ул. Бакулина, 1	ПИ-14-1
	2	Петров П.П.	мужской	ул. Бакулина, 1	ПИ-14-2
*					

Рисунок 2.27 – Форма відображення пов'язаних даних

2.3.2 Порядок виконання роботи

Після того, як Ви вивчили принципи роботи з такими СУБД, як MS Access, MS SQL Server та MySQL, а також технологією доступу до даних ADO.Net, можна переходити до виконання індивідуального завдання за

темою курсової роботи. Отже, Ваша реалізація на цю лабу повинна містити наступний функціонал:

- Наявність БД (достатньо 3-х таблиць) в тій СУБД, яку Ви обрали. БД повинна бути заповнена тестовою інформацією;
- Ви повинні продемонструвати можливість огляду БД засобами СУБД (наприклад, за допомогою Менеджмент студії для MS SQL Server);
- Ваш додаток повинен бути підключений за обраною Вами технологією до БД (готуйтеся під час захисту розповідати докладно про свою технології доступу до даних);
- Ваш додаток повинен підтримувати наступний функціонал:
 - можливість переглядання та навігації за трьома таблицями БД;
 - можливість додавання, видалення та редагування інформацію в таблицях за допомогою стандартного навігатора;
 - можливість переглядання інформації з двох підпорядкованих таблиць, що пов'язані відношенням «один-до-багатьох».

2.4 Зміст звіту

Звіт повинен містити: мету, завдання, отримані результати у вигляді форми з переглядання БД в засобах СУБД, скрин-шотів головних форм або сторінок додатку, та тексти обробників подій з додатку. Їх виконання (вид форми і код), висновки з роботи. Звіт своєчасно необхідно завантажити до дистанційного курсу.

2.5 Контрольні завдання і питання

- 1 Які основні об'єкти для роботи з даними за технологією ADO.NET?
- 2 Яке призначення об'єкту DataSet?
- 3 Що таке провайдер даних, які об'єкти він містить за технологією ADO.NET?
- 4 Які провайдери існують для підтримки технології ADO.NET?
- 5 Які елементи використовуються для підключення даних?
- 6 Які елементи використовуються для відображення даних?
- 7 У чому відмінність роботи з даними при використанні різних СУБД?

8 Як підтвердити зміни в таблиці?