

Лабораторна робота №3

Робота із базами даних в .NET. Використання бібліотеки Entity Framework.

Виконав Зубченко А.С

Передумова

Виконайте лабораторну роботу №2. Створіть нову гілку у раніше створеному гіт репозиторії *NUPP_NET_2025_{Номер Групи}_TK_{Прізвище}_Lab* із назвою **lab3**, яка міститиме код із другої лабораторної роботи.

Для здачі лабораторної роботи, необхідно буде створити пул реквест із гілки **lab3** у **master**, або якщо на момент здачі пул реквест із гілки lab2 у master не закритий - із гілки lab3 у lab2.

Завдання

1. Створити проєкт {Назва тематики}.Infrastructure. У новоствореному проєкті створити папку Models. У створеній папці додати модель(набір класів), яка відповідає створеній у першій лабораторній роботі моделі, та містить всередині себе властивості. Наприклад, якщо існував у першій лабораторній клас Bus, у папці Models необхідно створити клас BusModel.
Отримана модель має обов'язково мати зв'язки один-до-одного та один-до-багатьох. За додавання зв'язків багато-до-багатьох можна отримати додаткові бали. Для наслідування більш пріоритетно застосовувати підхід Таблиця на тип (Table-per-Type).
2. Створити клас {Назва тематики}Context, який наслідуватиметься від класу DbContext та опише схему бази даних для обраної тематики застосовуючи Entity Framework та анотації або Fluent API на вибір, Fluent API більш пріоритетно.
3. Створити міграцію використовуючи Entity Framework. Створену міграцію застосувати до реляційної бази даних. Можна використовувати будь-яку реляційну СУБД, як MS SQL, PostgreSQL, MySQL, тощо. Також слід використати SQLite, коли немає можливості розробляти та здавати Лабораторну роботу на одній системі. При використанні SQLite файл бази даних необхідно додати до гіт репозиторія.
4. Реалізувати шаблон проєктування Репозиторій, {Назва тематики}Context який буде використовувати клас {Назва тематики}Context для досьупу до даних із БД:

```
public interface IRepository<T> where T : class
{
    Task<T> GetByIdAsync(int id);
    Task<IEnumerable<T>> GetAllAsync();
    Task AddAsync(T entity);
    Task Update(T entity);
    Task Delete(T entity);
}
```

5. Оновити асинхронну версію дженерік CRUD сервісу, щоб він використовував репозиторій для доступу до даних:

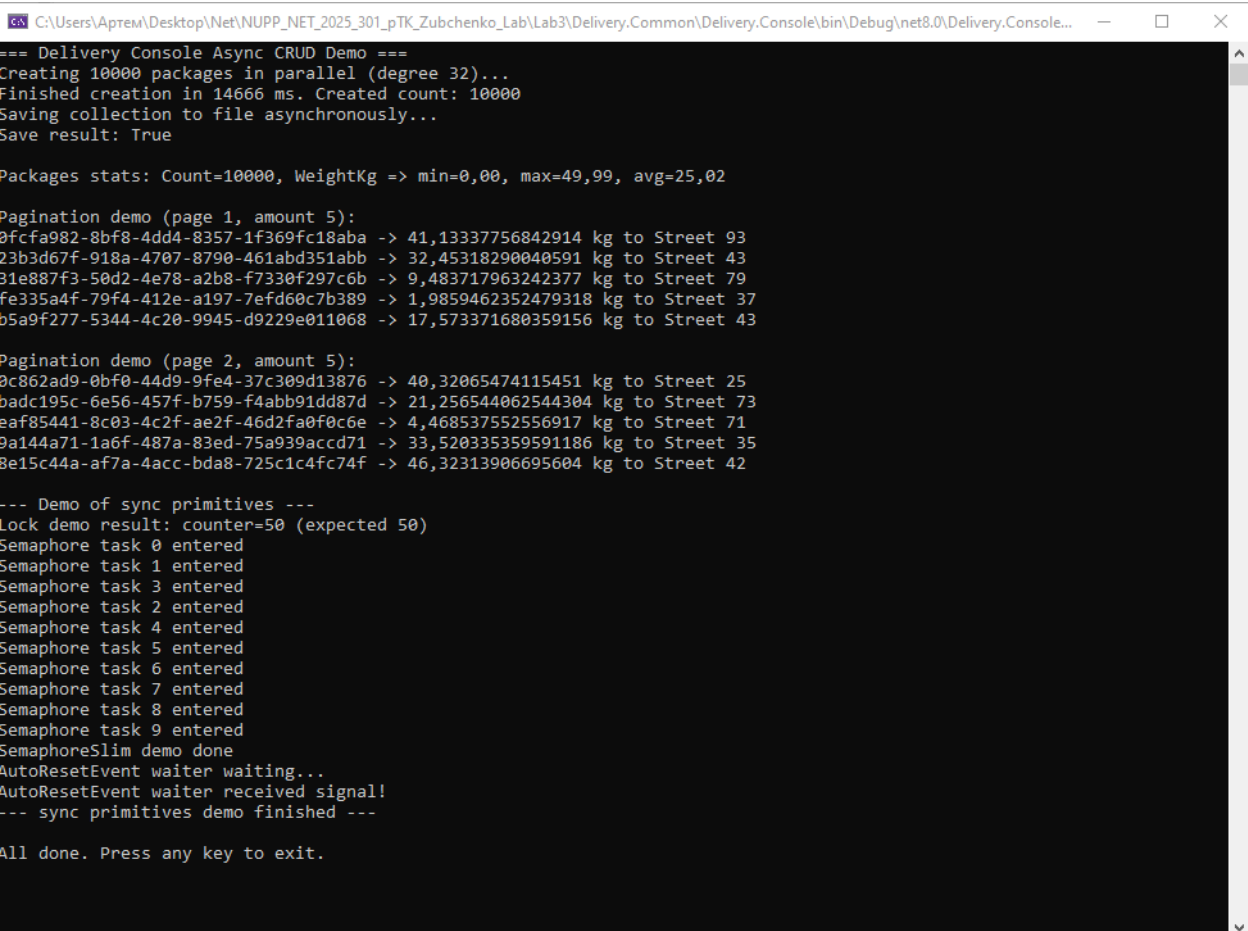
```
public interface ICrudServiceAsync<T>
{
    public Task<bool> CreateAsync(T element);
    public Task<T> ReadAsync(Guid id);
    public Task<IEnumerable<T>> ReadAllAsync();
    public Task<IEnumerable<T>> ReadAllAsync(int page, int amount);
    public Task<bool> UpdateAsync(T element);
    public Task<bool> RemoveAsync(T element);
    public Task<bool> SaveAsync();
}
```

6. Модифікуйте консольний застосунок, щоб він використовував оновлену версію CRUD сервісу та дані із бази даних.

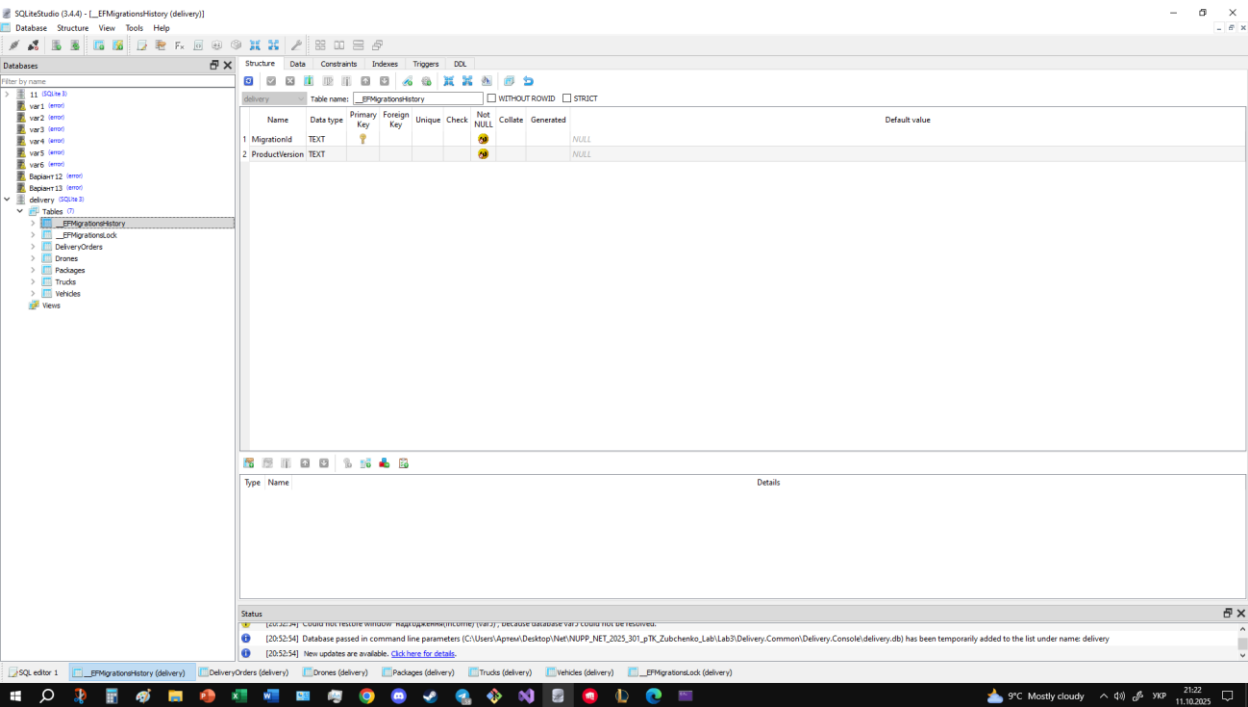
До PR готової лабораторної роботи додайте PDF файл у якому будуть результати виконання консольного застосунку, скріншоти створеної бази даних та ERD-діаграма БД.

ФАЙЛИ ПРОЕКТУ: NUPP_NET_2025_301_pTK_Zubchenko_Lab\Lab3\Delivery.Common

Результат виконання консольного застосунку



Скріншоти створеної БД



SQLiteStudio (3.4.4) - [_EFMigrationsLock (delivery)]

Database Structure View Tools Help

Databases

Filter by name

11 SQLite 3.39.1

- var1 (mem)
- var2 (mem)
- var3 (mem)
- var4 (mem)
- var5 (mem)
- var6 (mem)
- SQLite12 (mem)
- SQLite13 (mem)
- delivery (SQLite 3.39.1)
 - Tables (7)
 - __EFMigrationsHistory
 - __EFMigrationsLock**
 - DeliveryOrders
 - Drones
 - Packages
 - Trucks
 - Vehicles

Table name: __EFMigrationsLock

WITHOUT ROWID ☐ STRUCT ☐

Name	Data type	Primary Key	Foreign Key	Unique	Check	Not NULL	Collate	Generated	Default value
1 Id	INTEGER								NULL
2 Timestamp	TEXT								NULL

Type Name Details

Status

[2023-04-24] Could not restore window: migrationName(column: var1); because database var1 could not be restored.

[2023-54] Database passed in command line parameters (C:\Users\Apreu\Desktop\Net\NUPP_NET_2023_301_gTK_Zubchenko_Lab3\Delivery.Common\Delivery.Console\delivery.db) has been temporarily added to the list under name: delivery

[2023-54] New updates are available. [Click here for details](#)

SQL editor 1: __EFMigrationsHistory (delivery) | DeliveryOrders (delivery) | Drones (delivery) | Packages (delivery) | Trucks (delivery) | Vehicles (delivery) | __EFMigrationsLock (delivery)

9°C Mostly cloudy 21:22 11.10.2025

SQLiteStudio (3.4.4) - [DeliveryOrders (delivery)]

Database Structure View Tools Help

Databases

Filter by name

11 SQLite 3.39.1

- var1 (mem)
- var2 (mem)
- var3 (mem)
- var4 (mem)
- var5 (mem)
- var6 (mem)
- SQLite12 (mem)
- SQLite13 (mem)
- delivery (SQLite 3.39.1)
 - Tables (7)
 - __EFMigrationsHistory
 - __EFMigrationsLock
 - DeliveryOrders**
 - Drones
 - Packages
 - Trucks
 - Vehicles

Table name: DeliveryOrders

WITHOUT ROWID ☐ STRUCT ☐

Name	Data type	Primary Key	Foreign Key	Unique	Check	Not NULL	Collate	Generated	Default value
1 Id	TEXT								NULL
2 AssignedVehicleId	TEXT								NULL
3 PackageId	TEXT								NULL
4 ScheduledDate	TEXT								NULL
5 Status	TEXT								NULL

Type Name Details

1	FOREIGN KEY	FK_DeliveryOrders_Packages_PackageId	(PackageId) REFERENCES Packages (Id) ON DELETE CASCADE
2	FOREIGN KEY	FK_DeliveryOrders_Vehicles_AssignedVehicleId	(AssignedVehicleId) REFERENCES Vehicles (Id) ON DELETE CASCADE

Status

[2023-04-24] Could not restore window: migrationName(column: var1); because database var1 could not be restored.

[2023-54] Database passed in command line parameters (C:\Users\Apreu\Desktop\Net\NUPP_NET_2023_301_gTK_Zubchenko_Lab3\Delivery.Common\Delivery.Console\delivery.db) has been temporarily added to the list under name: delivery

[2023-54] New updates are available. [Click here for details](#)

SQL editor 1: __EFMigrationsHistory (delivery) | **DeliveryOrders (delivery)** | Drones (delivery) | Packages (delivery) | Trucks (delivery) | Vehicles (delivery) | __EFMigrationsLock (delivery)

9°C Mostly cloudy 21:22 11.10.2025

SQLiteStudio (3.4.4) - [Drones (delivery)]

Database Structure View Tools Help

Databases

Filter by name

11 SQLite 3

- var1 (text)
- var2 (text)
- var3 (text)
- var4 (text)
- var5 (text)
- var6 (text)
- Equipment12 (text)
- Equipment13 (text)
- delivery (SQLite 3)
 - Tables (7)
 - __EPNigrationsHistory
 - __EPNigrationsLock
 - DeliveryOrders
 - Drones
 - Packages
 - Trucks
 - Vehicles
 - Views

Structure Data Constraints Indexes Triggers DDL

Table name: Drones

WITHOUT ROWID ☐ STRICT ☐

Name	Data type	Primary Key	Foreign Key	Unique	Check	Not NULL	Collate	Generated	Default value
1 Id	TEXT								NULL
2 BatteryLevel	REAL								NULL
3 MaxPayloadKg	REAL								NULL
4 MaxFlightTimeMinutes	REAL								NULL
5 MaxAltitudeMeters	REAL								NULL

Type Name Details

1 FOREIGN KEY FK_Drones_Vehicles_Id (Id) REFERENCES Vehicles (Id) ON DELETE CASCADE

Status

[2023-07-24] "sqlite3" FROM "TESSARTE" WINDOW "MAGAZIN" (C:\Users\Aprew\Desktop\Net\NUPP_NET_2025_301_gTK_Zubchenko_Lab7\Delivery.Common\Delivery.Console\delivery.db) has been temporarily added to the list under name: delivery

[2023-07-24] Database passed in command line parameters (C:\Users\Aprew\Desktop\Net\NUPP_NET_2025_301_gTK_Zubchenko_Lab7\Delivery.Common\Delivery.Console\delivery.db) has been temporarily added to the list under name: delivery

[2023-07-24] New updates are available. [Click here for details.](#)

SQL editor 1: __EPNigrationsHistory (delivery) DeliveryOrders (delivery) Drones (delivery) Packages (delivery) Trucks (delivery) Vehicles (delivery) __EPNigrationsLock (delivery)

9°C Mostly cloudy 20:23 11.10.2025

SQLiteStudio (3.4.4) - [Packages (delivery)]

Database Structure View Tools Help

Databases

Filter by name

11 SQLite 3

- var1 (text)
- var2 (text)
- var3 (text)
- var4 (text)
- var5 (text)
- var6 (text)
- Equipment12 (text)
- Equipment13 (text)
- delivery (SQLite 3)
 - Tables (7)
 - __EPNigrationsHistory
 - __EPNigrationsLock
 - DeliveryOrders
 - Drones
 - Packages
 - Trucks
 - Vehicles
 - Views

Structure Data Constraints Indexes Triggers DDL

Table name: Packages

WITHOUT ROWID ☐ STRICT ☐

Name	Data type	Primary Key	Foreign Key	Unique	Check	Not NULL	Collate	Generated	Default value
1 Id	TEXT								NULL
2 DestinationAddress	TEXT								NULL
3 WeightKg	REAL								NULL
4 IsFragile	INTEGER								NULL

Type Name Details

Status

[2023-07-24] "sqlite3" FROM "TESSARTE" WINDOW "MAGAZIN" (C:\Users\Aprew\Desktop\Net\NUPP_NET_2025_301_gTK_Zubchenko_Lab7\Delivery.Common\Delivery.Console\delivery.db) has been temporarily added to the list under name: delivery

[2023-07-24] Database passed in command line parameters (C:\Users\Aprew\Desktop\Net\NUPP_NET_2025_301_gTK_Zubchenko_Lab7\Delivery.Common\Delivery.Console\delivery.db) has been temporarily added to the list under name: delivery

[2023-07-24] New updates are available. [Click here for details.](#)

SQL editor 1: __EPNigrationsHistory (delivery) DeliveryOrders (delivery) Drones (delivery) Packages (delivery) Trucks (delivery) Vehicles (delivery) __EPNigrationsLock (delivery)

9°C Mostly cloudy 20:23 11.10.2025

SQLStudio (3.4.4) - [Trucks (delivery)]

Database Structure View Tools Help

Databases

Filter by name

11 (SQL) 0

var1 (smol)

var2 (smol)

var3 (smol)

var4 (smol)

var5 (smol)

var6 (smol)

Equipment 12 (smol)

Equipment 13 (smol)

delivery (SQL) 0

Tables (7)

__EFMigrationsHistory

__EFMigrationsLock

DeliveryOrders

Drones

Packages

Trucks

Vehicles

Views

Structure Data Constraints Indexes Triggers DDL

delivery Table name: Trucks WITHOUT ROWID STRUCT

Name	Data type	Primary Key	Foreign Key	Unique	Check	Not NULL	Collate	Generated	Default value
1 Id	TEXT								NULL
2 HalfRefrigeration	INTEGER								NULL
3 NumberOfAxes	INTEGER								NULL

Type Name Details

1 FOREIGN KEY FK_Trucks_Vehicles_Id (Id) REFERENCES Vehicles (Id) ON DELETE CASCADE

Status

[20:52:34] C:\Users\Aprex\Desktop\Net\NUPP_NET_2025_301_gTK_Zubchenko_Lab\Lab7\Delivery.Common\Delivery.Console\delivery.db has been temporarily added to the list under name: delivery

[20:52:54] Database passed in command line parameters (C:\Users\Aprex\Desktop\Net\NUPP_NET_2025_301_gTK_Zubchenko_Lab\Lab7\Delivery.Common\Delivery.Console\delivery.db) has been temporarily added to the list under name: delivery

[20:52:54] New updates are available. [Click here for details.](#)

SQL editor 1: __EFMigrationsHistory (delivery) DeliveryOrders (delivery) Drones (delivery) Packages (delivery) Trucks (delivery) Vehicles (delivery) __EFMigrationsLock (delivery)

9°C Mostly cloudy 20:24 11.10.2025

SQLStudio (3.4.4) - [Vehicles (delivery)]

Database Structure View Tools Help

Databases

Filter by name

11 (SQL) 0

var1 (smol)

var2 (smol)

var3 (smol)

var4 (smol)

var5 (smol)

var6 (smol)

Equipment 12 (smol)

Equipment 13 (smol)

delivery (SQL) 0

Tables (7)

__EFMigrationsHistory

__EFMigrationsLock

DeliveryOrders

Drones

Packages

Trucks

Vehicles

Views

Structure Data Constraints Indexes Triggers DDL

delivery Table name: Vehicles WITHOUT ROWID STRUCT

Name	Data type	Primary Key	Foreign Key	Unique	Check	Not NULL	Collate	Generated	Default value
1 Id	TEXT								NULL
2 LicensePlate	TEXT								NULL
3 Model	TEXT								NULL
4 MaxLoadKg	REAL								NULL
5 FuelConsumption	REAL								NULL
6 IsOperational	INTEGER								NULL

Type Name Details

Status

[20:52:34] C:\Users\Aprex\Desktop\Net\NUPP_NET_2025_301_gTK_Zubchenko_Lab\Lab7\Delivery.Common\Delivery.Console\delivery.db has been temporarily added to the list under name: delivery

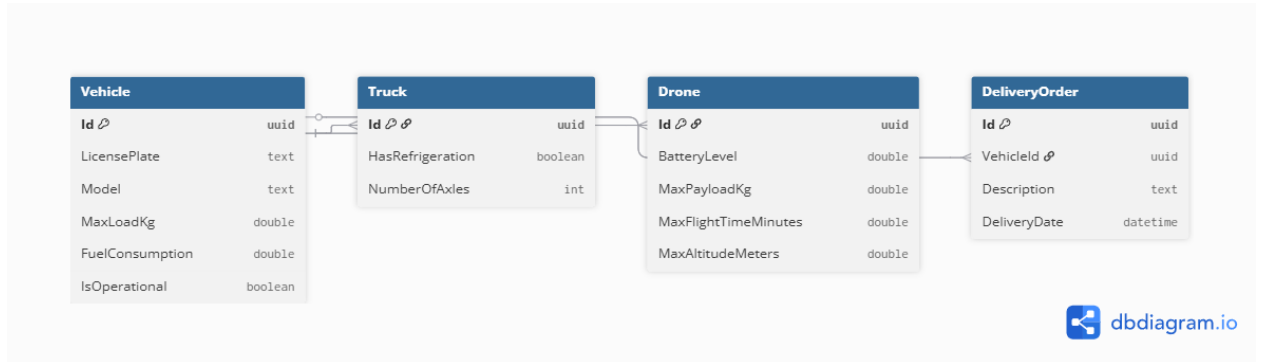
[20:52:54] Database passed in command line parameters (C:\Users\Aprex\Desktop\Net\NUPP_NET_2025_301_gTK_Zubchenko_Lab\Lab7\Delivery.Common\Delivery.Console\delivery.db) has been temporarily added to the list under name: delivery

[20:52:54] New updates are available. [Click here for details.](#)

SQL editor 1: __EFMigrationsHistory (delivery) DeliveryOrders (delivery) Drones (delivery) Packages (delivery) Trucks (delivery) Vehicles (delivery) __EFMigrationsLock (delivery)

9°C Mostly cloudy 20:24 11.10.2025

ERD-діаграма БД



Контрольні питання

1. Що таке реляційні бази даних? Що таке СУБД? Які СУБД ви знаєте?

Реляційні бази даних (РБД) — це бази даних, у яких дані зберігаються у вигляді **таблиць** (relations), що мають **рядки** (записи) та **стовпці** (поля). Кожен рядок унікально ідентифікується первинним ключем (Primary Key).

СУБД (система управління базами даних) — це програмне забезпечення для створення, зберігання, обробки та управління базами даних.

Приклади СУБД:

- Реляційні: Microsoft SQL Server, MySQL, PostgreSQL, SQLite, Oracle DB
- Нереляційні: MongoDB, CouchDB, Redis, Cassandra

2. Що позначає термін таблиця у реляційній БД? Які існують зв'язки у реляційній БД?

Таблиця — це структура для зберігання даних у РБД, яка складається з **рядків** (record, tuple) та **стовпців** (attributes, fields).

Основні типи зв'язків:

- **Один-до-одного (1:1)** — кожен запис однієї таблиці пов'язаний з одним записом іншої таблиці.
- **Один-до-багатьох (1:N)** — один запис однієї таблиці пов'язаний з багатьма записами іншої таблиці.
- **Багато-до-багатьох (M:N)** — записи обох таблиць можуть бути пов'язані з багатьма записами іншої таблиці, зазвичай реалізується через проміжну таблицю (join table).

3. Що таке ERD-діаграма? Як її створити та для чого вона потрібна?

ERD (Entity-Relationship Diagram) — це графічне представлення структури бази даних, яке показує сутності (Entities), атрибути та зв'язки між ними.

Призначення:

- Візуалізувати структуру даних

- Планувати схему бази перед її створенням
- Покращити розуміння взаємозв'язків у системі

Як створити:

- Ручне малювання в draw.io, Lucidchart, PowerPoint
- Автоматично за допомогою інструментів: MySQL Workbench, pgAdmin, Visual Studio (Database Diagram)

4. Що таке DbContext і яку роль він відіграє в роботі з базою даних?

DbContext — це основний клас у **Entity Framework Core**, який виконує роль мосту між кодом і базою даних.

Він забезпечує:

- Збереження та отримання об'єктів із БД
- Визначення DbSet<T> для кожної таблиці
- Конфігурацію моделі через Fluent API або Data Annotations
- Транзакції, відстеження змін та роботу з міграціями

5. Що таке зв'язки один-до-одного, один-до-багатьох і багато-до-багатьох у контексті EF Core?

- **Один-до-одного (1:1):** Кожен об'єкт одного класу пов'язаний з одним об'єктом іншого класу.
- **Один-до-багатьох (1:N):** Один об'єкт може мати колекцію багатьох інших об'єктів.
- **Багато-до-багатьох (M:N):** Об'єкти з обох класів можуть мати колекції один одного; реалізується через проміжну таблицю (join entity) у EF Core 5+ можна без явного класу join.

6. Як реалізувати зв'язок один-до-одного за допомогою Fluent API?

```
modelBuilder.Entity<User>()
    .HasOne(u => u.Profile)
    .WithOne(p => p.User)
    .HasForeignKey<Profile>(p => p.UserId);
```

- HasOne — визначає навігаційну властивість одного боку
- WithOne — навігація на інший бік
- HasForeignKey — вказує поле зовнішнього ключа

7. Приклад реалізації зв'язку багато-до-багатьох у EF Core

```
modelBuilder.Entity<Student>()
    .HasMany(s => s.Courses)
    .WithMany(c => c.Students)
    .UsingEntity<Enrollment>()
        .Join(j => j.HasOne(e => e.Course).WithMany(),
              j => j.HasOne(e => e.Student).WithMany())
    ;
```


- UsingEntity – проміжна таблиця для M:N

8. Різниця між Data Annotations та Fluent API

Data Annotations	Fluent API
Атрибути безпосередньо в класі	Конфігурація в OnModelCreating
Просте використання	Більше контролю, складні конфігурації
Менше гнучкості	Підтримує всі можливі конфігурації EF Core

9. Як створюється та застосовується міграція в EF Core?

1. Створення міграції:

```
dotnet ef migrations add InitialCreate --project <Infrastructure> --startup-project <Console>
```

2. Застосування міграції до БД:

```
dotnet ef database update --project <Infrastructure> --startup-project <Console>
```

10. Мета проєкту {Назва тематики}.Infrastructure

- Містить **шари доступу до даних** (Repository, DbContext)
- Відокремлює бізнес-логіку від збереження даних
- Забезпечує **інкапсуляцію логіки БД** для інших частин проєкту

11. Різниця між доменною моделлю та моделлю бази даних

- **Доменна модель** – відображає бізнес-логіку та правила системи
- **Модель бази даних** – відображає структуру таблиць, зв'язки, типи даних у БД

12. Яку проблему вирішує шаблон «Репозиторій»

- Інкапсулює доступ до даних
- Забезпечує **уніфікований інтерфейс** для CRUD
- Полегшує тестування і підтримку бізнес-логіки

13. Принципова різниця між реляційними та нереляційними БД

РБД	NoSQL
Таблиці, рядки, колонки	Документи, ключ-значення, графи, колонки
Строгі схеми	Гнучкі схеми
SQL-запити	API або специфічна мова запитів
ACID	Може підтримувати BASE