

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 проект *{Назва тематики}.Infrastructure*. У новоствореному проекті 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 асинхрону 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

```
C:\Users\Artem\Desktop\Net\NUPP_NET_2025_301_pTK_Zubchenko_Lab\Lab3\Delivery.Common\Delivery\Console\bin\Debug\net8.0\Delivery.Console... — □ ×  
== Delivery Console Async CRUD Demo ==  
Creating 10000 packages in parallel (degree 32)...  
Finished creation in 14666 ms. Created count: 10000  
Saving collection to file asynchronously...  
Save result: True  
  
Packages stats: Count=10000, WeightKg => min=0,00, max=49,99, avg=25,02  
  
Pagination demo (page 1, amount 5):  
0fcfa982-8bf8-4dd4-8357-1f369fc18aba -> 41,13337756842914 kg to Street 93  
23b3d67f-918a-461abd351abb -> 32,45318290040591 kg to Street 43  
31e887f3-50d2-4e78-a2b8-f7330f297c6b -> 9,48317963242377 kg to Street 79  
Fe335a4f-79f4-412e-a197-7efd60c7b389 -> 1,9859462352479318 kg to Street 37  
b5a9f277-5344-4c20-9945-d9229e011068 -> 17,573371680359156 kg to Street 43  
  
Pagination demo (page 2, amount 5):  
9c862ad9-0bf0-44d9-9fe4-37c309d13876 -> 40,32065474115451 kg to Street 25  
badc195c-6e56-457f-b759-f4abb91dd87d -> 21,256544062544304 kg to Street 73  
eaf85441-8c03-4c2f-ae2f-46d2fa0f0c6e -> 4,468537552556917 kg to Street 71  
9a144a71-1a6f-487a-83ed-75a939acc71 -> 33,520335359591186 kg to Street 35  
8e15c44a-af7a-4acc-bda8-725c1c4fc74f -> 46,32313906695604 kg to Street 42  
  
--- Demo of sync primitives ---  
Lock demo result: counter=50 (expected 50)  
Semaphore task 0 entered  
Semaphore task 1 entered  
Semaphore task 3 entered  
Semaphore task 2 entered  
Semaphore task 4 entered  
Semaphore task 5 entered  
Semaphore task 6 entered  
Semaphore task 7 entered  
Semaphore task 8 entered  
Semaphore task 9 entered  
SemaphoreSlim demo done  
AutoResetEvent waiter waiting...  
AutoResetEvent waiter received signal!  
--- sync primitives demo finished ---  
  
All done. Press any key to exit.
```

Скріншоти створеної БД

SQLiteStudio (3.4.4) - [L_EFMigrationsHistory (delivery)]

Databases

Tables

Structure Data Constraints Indexes Triggers DDL

Filter by name

11 (Table 1)

var1 (view)

var2 (view)

var3 (view)

var4 (view)

var5 (view)

var6 (view)

Bspvar12 (view)

Bspvar13 (view)

delivery (Table 2)

Tables (1)

1 _EFMigrationHistory

2 _EFMigrationLock

3 DeliveryOrders

4 Drones

5 Packages

6 Trucks

7 Vehicles

Views

Table name: _EFMigrationHistory

WITHOUT ROWID STRUCT

Name Data type Primary Key Foreign Key Unique Check Not NULL Collate Generated Default value

1 MigrationId TEXT Primary Key Unique Check Not NULL Collate Generated NULL

2 ProductVersion TEXT Unique Check Not NULL Collate Generated NULL

Type Name Details

Status

[Locally] Could not restore window 'Navigation(Home) (var1)', because database var1 could not be restored.

[2052-54] Database passed in command line parameters (C:\Users\Artem\Desktop\Net\NUPP_NET_2025_301_pTK_Zubchenko_Lab\Lab3\Delivery.Common\Delivery\Console\delivery.db) has been temporarily added to the list under name: delivery

[2052-54] New updates are available. Click here for details.

SQL editor 1 _EFMigrationHistory (delivery) DeliveryOrders (delivery) Drones (delivery) Packages (delivery) Trucks (delivery) Vehicles (delivery) _EFMigrationLock (delivery)

Windows 9°C Mostly cloudy 21:22 11.10.2025

SQLStudio (3.4.4) - [__EFMigrationsLock (delivery)]

Databases Structure View Tools Help

Tables (7)

- __EFMigrationsHistory
- __EFMigrationsLock
- DeliveryOrders
- Drones
- Packages
- Trucks
- Vehicles

Views

Structure Data Constraints Indexes Triggers DDL

Delivery Table name: __EFMigrationsLock WITHOUT ROWID STRUCT

Name	Data type	Primary Key	Foreign Key	Unique	Check	Not NULL	Collate	Generated	Default value
1 Id	INTEGER	<input checked="" type="checkbox"/>				<input checked="" type="checkbox"/>			NULL
2 Timestamp	TEXT					<input checked="" type="checkbox"/>			NULL

Type Name Details

Status

- [Warning] Could not resolve reference 'navigation(migration) (var1)', because database var1 is not be resolved.
- [Information] Database passed in command line parameters (C:\Users\Aprela\Desktop\Net\NUPP.NET_2025_301_pTK_Zubchenko_Lab3\Delivery\Common\Delivery\Console\delivery.db) has been temporarily added to the list under name: delivery
- [Information] [20532-54] 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

SQLStudio (3.4.4) - [DeliveryOrders (delivery)]

Databases Structure View Tools Help

Tables (7)

- __EFMigrationsHistory
- __EFMigrationsLock
- DeliveryOrders
- Drones
- Packages
- Trucks
- Vehicles

Views

Structure Data Constraints Indexes Triggers DDL

Delivery Table name: DeliveryOrders WITHOUT ROWID STRUCT

Name	Data type	Primary Key	Foreign Key	Unique	Check	Not NULL	Collate	Generated	Default value
1 Id	TEXT	<input checked="" type="checkbox"/>				<input checked="" type="checkbox"/>			NULL
2 AssignedVehicleId	TEXT	<input checked="" type="checkbox"/>				<input checked="" type="checkbox"/>			NULL
3 PackageId	TEXT	<input checked="" type="checkbox"/>				<input checked="" type="checkbox"/>			NULL
4 ScheduledDate	TEXT	<input checked="" type="checkbox"/>				<input checked="" type="checkbox"/>			NULL
5 Status	TEXT	<input checked="" type="checkbox"/>				<input checked="" type="checkbox"/>			NULL

Type Name Details

1 FOREIGN KEY FK_DeliveryOrders_Package_PackageId REFERENCES Packages (id) ON DELETE CASCADE

2 FOREIGN KEY FK_DeliveryOrders_Vehicles_AssignedVehicleId (AssignedVehicleId) REFERENCES Vehicles (id) ON DELETE CASCADE

Status

- [Warning] Could not resolve reference 'navigation(migration) (var1)', because database var1 is not be resolved.
- [Information] Database passed in command line parameters (C:\Users\Aprela\Desktop\Net\NUPP.NET_2025_301_pTK_Zubchenko_Lab3\Delivery\Common\Delivery\Console\delivery.db) has been temporarily added to the list under name: delivery
- [Information] [20532-54] 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 21:23 11.10.2025

SQLiteStudio (3.4.0) - [Drones (delivery)]

Databases

Tables (7)

- 11 (SQLite)
- var1 (empty)
- var2 (empty)
- var3 (empty)
- var4 (empty)
- var5 (empty)
- var6 (empty)
- Battery12 (empty)
- Battery13 (empty)
- Delivery (empty)
- DeliveryOrders
- Drones
- Packages
- Trucks
- Vehicles
- Views

Structure

Table name: Drones

Delivery

Name	Data type	Primary Key	Foreign Key	Unique	Check	Not NULL	Collate	Generated	Default value
1 Id	TEXT	<input checked="" type="checkbox"/>				<input checked="" type="checkbox"/>			NULL
2 BatteryLevel	REAL					<input checked="" type="checkbox"/>			NULL
3 MaxPayloadKg	REAL					<input checked="" type="checkbox"/>			NULL
4 MaxFlightTimeMinutes	REAL					<input checked="" type="checkbox"/>			NULL
5 MaxAltitudeMeters	REAL					<input checked="" type="checkbox"/>			NULL

Details

Type Name

FOREIGN KEY FK_Drones_Vehicles_id (id) REFERENCES Vehicles(id) ON DELETE CASCADE

Status

[20532-54] Could not restore window 'MigrationHistory' because database 'var2' could not be resolved.

[20532-54] Database passed in command line parameters (C:\Users\Apreva\Desktop\Net\NUPP.NET_2025_301_pTK_Zubchenko_Lab\Lab3\Delivery\Common\Delivery\Console\delivery.db) has been temporarily added to the list under name: delivery

[20532-54] 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)

SQLiteStudio (3.4.0) - [Packages (delivery)]

Databases

Tables (7)

- 11 (SQLite)
- var1 (empty)
- var2 (empty)
- var3 (empty)
- var4 (empty)
- var5 (empty)
- var6 (empty)
- Battery12 (empty)
- Battery13 (empty)
- Delivery (empty)
- DeliveryOrders
- Drones
- Packages
- Trucks
- Vehicles
- Views

Structure

Table name: Packages

Delivery

Name	Data type	Primary Key	Foreign Key	Unique	Check	Not NULL	Collate	Generated	Default value
1 Id	TEXT	<input checked="" type="checkbox"/>				<input checked="" type="checkbox"/>			NULL
2 DestinationAddress	TEXT					<input checked="" type="checkbox"/>			NULL
3 WeightKg	REAL					<input checked="" type="checkbox"/>			NULL
4 IsFragile	INTEGER					<input checked="" type="checkbox"/>			NULL

Details

Type Name

Status

[20532-54] Could not restore window 'MigrationHistory' because database 'var2' could not be resolved.

[20532-54] Database passed in command line parameters (C:\Users\Apreva\Desktop\Net\NUPP.NET_2025_301_pTK_Zubchenko_Lab\Lab3\Delivery\Common\Delivery\Console\delivery.db) has been temporarily added to the list under name: delivery

[20532-54] 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)

SQLiteStudio (3.4.4) - [Trucks (delivery)]

Database Structure View Tools Help

Databases

Tables (7)

- var1 (empty)
- var2 (empty)
- var3 (empty)
- var4 (empty)
- var5 (empty)
- var6 (empty)
- var7 (empty)
- var8 (empty)
- var9 (empty)
- var10 (empty)
- var11 (empty)
- var12 (empty)
- Baptain13 (empty)
- delivery (33 rows)

 - EFMigrationHistory
 - EFMigrationLock
 - DeliveryOrders
 - Drones
 - Packages
 - Trucks
 - Vehicles
 - Views

Trucks Table name: Trucks

Structure Data Constraints Indexes Triggers DDL

Table name: Trucks

Name Data type Primary Key Foreign Key Unique Check Not NULL Collate Generated Default value

1 Id TEXT Primary Key Unique Check Not NULL Collate Generated Default value

2 HasRefrigeration INTEGER Unique Check Not NULL Collate Generated Default value

3 NumberOfAxles INTEGER Unique Check Not NULL Collate Generated Default value

Details

Type Name Details

FOREIGN KEY FK_Trucks_Vehicles_id (Id) REFERENCES Vehicles (id) ON DELETE CASCADE

Status

[20532-54] Could not restore window 'MigrationHistory' because database 'var2' could not be resolved.

[20532-54] Database passed in command line parameters (C:\Users\Apreva\Desktop\NetNUPP.NET_2025_301_pTK_Zubchenko_Lab\Lab3\Delivery\Common\Delivery\Console\delivery.db) has been temporarily added to the list under name: delivery

[20532-54] New updates are available. [Click here for details.](#)

SQL editor 1 EFMigrationHistory (delivery) DeliveryOrders (delivery) Drones (delivery) Packages (delivery) Trucks (delivery) Vehicles (delivery) EFMigrationLock (delivery)

Windows Taskbar: SQL editor 1, EFMigrationHistory (delivery), DeliveryOrders (delivery), Drones (delivery), Packages (delivery), Trucks (delivery), Vehicles (delivery), EFMigrationLock (delivery)

System tray: 9°C Mostly cloudy, 21:23, 11.10.2023

SQLiteStudio (3.4.4) - [Vehicles (delivery)]

Database Structure View Tools Help

Databases

Tables (7)

- var1 (empty)
- var2 (empty)
- var3 (empty)
- var4 (empty)
- var5 (empty)
- var6 (empty)
- var7 (empty)
- var8 (empty)
- var9 (empty)
- var10 (empty)
- var11 (empty)
- var12 (empty)
- Baptain13 (empty)
- delivery (33 rows)

 - EFMigrationHistory
 - EFMigrationLock
 - DeliveryOrders
 - Drones
 - Packages
 - Trucks
 - Vehicles
 - Views

Vehicles Table name: Vehicles

Structure Data Constraints Indexes Triggers DDL

Table name: Vehicles

Name Data type Primary Key Foreign Key Unique Check Not NULL Collate Generated Default value

1 Id TEXT Primary Key Unique Check Not NULL Collate Generated Default value

2 LicensePlate TEXT Unique Check Not NULL Collate Generated Default value

3 Model TEXT Unique Check Not NULL Collate Generated Default value

4 MaxLoadkg REAL Unique Check Not NULL Collate Generated Default value

5 FuelConsumption REAL Unique Check Not NULL Collate Generated Default value

6 IsOperational INTEGER Unique Check Not NULL Collate Generated Default value

Details

Type Name Details

Status

[20532-54] Could not restore window 'MigrationHistory' because database 'var2' could not be resolved.

[20532-54] Database passed in command line parameters (C:\Users\Apreva\Desktop\NetNUPP.NET_2025_301_pTK_Zubchenko_Lab\Lab3\Delivery\Common\Delivery\Console\delivery.db) has been temporarily added to the list under name: delivery

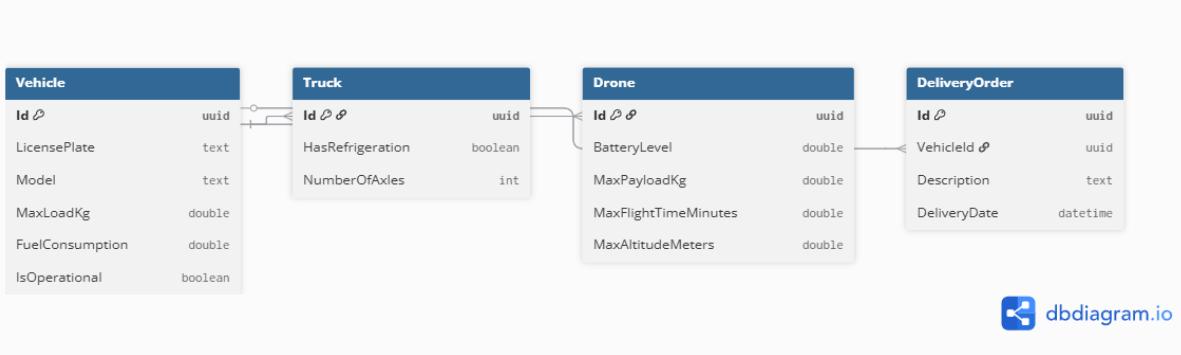
[20532-54] New updates are available. [Click here for details.](#)

SQL editor 1 EFMigrationHistory (delivery) DeliveryOrders (delivery) Drones (delivery) Packages (delivery) Trucks (delivery) Vehicles (delivery) EFMigrationLock (delivery)

Windows Taskbar: SQL editor 1, EFMigrationHistory (delivery), DeliveryOrders (delivery), Drones (delivery), Packages (delivery), Trucks (delivery), Vehicles (delivery), EFMigrationLock (delivery)

System tray: 9°C Mostly cloudy, 21:24, 11.10.2023

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> дляожної 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>(
        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 проекту {Назва тематики}.Infrastructure

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

11. Різниця між доменною моделлю та моделлю бази даних

- **Домenna 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