

Лекція 4

Технологія роботи з базами даних на платформі .NET Framework.

Архітектура ADO.NET

ADO.NET є технологією роботи з даними, яка заснована на платформі .NET Framework

Інтерфейс DAO – це об'єктна модель, заснована на бібліотеці Microsoft Jet Engine , яка використовується в Microsoft Access .

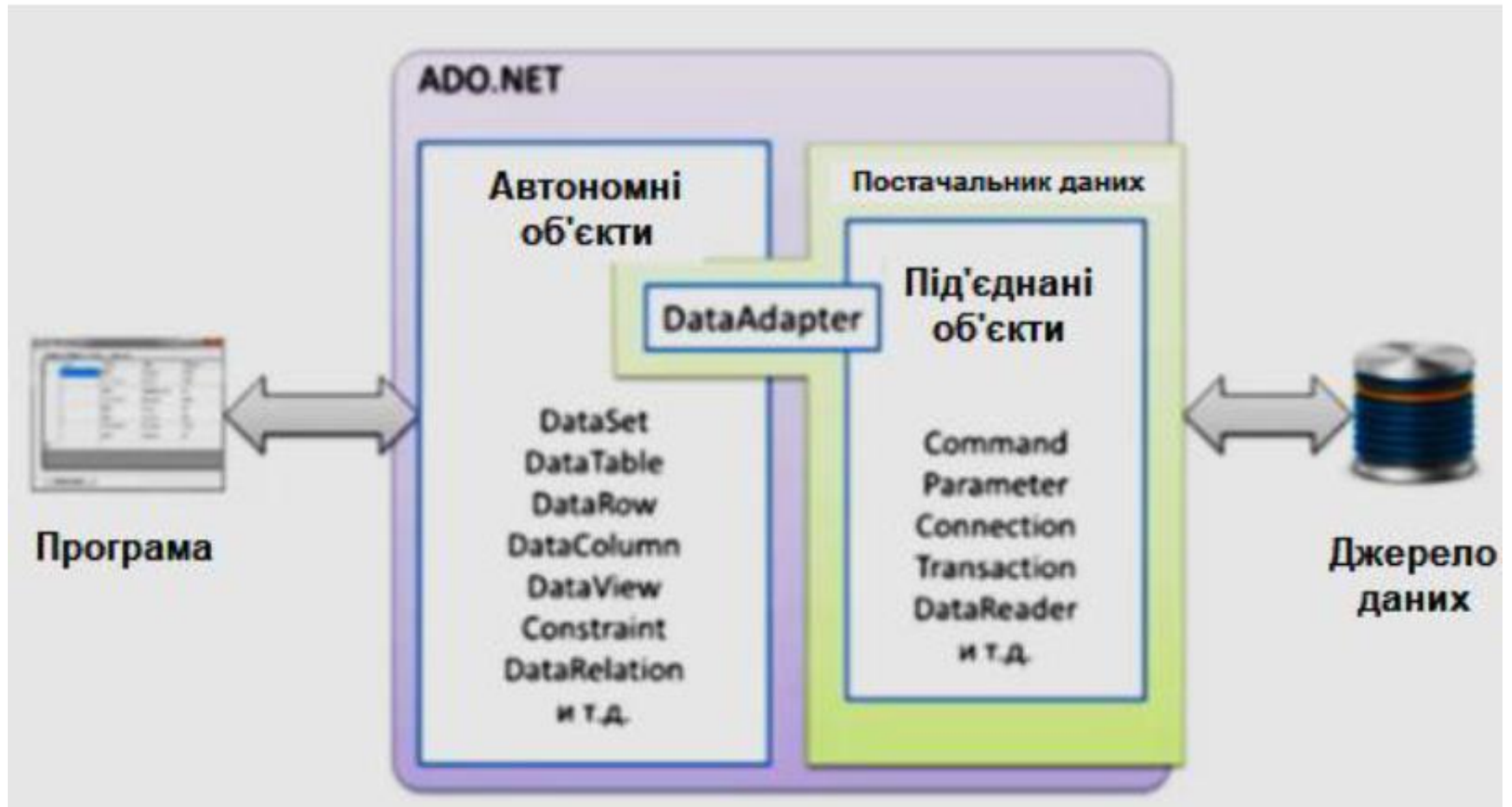
Інтерфейс ODBC є класичний інтерфейс, заснований не на об'єктній моделі, а на використанні дескрипторів API в стилі мови C.

У технології **OLE DB** доступ до даних здійснюється у вигляді набору чітко визначених абстрактних об'єктів, званих котипами (cotypes).

Бібліотека **ADO** за своєю суттю є «нашаруванням» на OLE DB, проте відрізняється від неї меншою кількістю об'єктів і тим, що кожному типу об'єктів відповідає один головний інтерфейс.

Бібліотека **ADO . NET** підхоплює естафету, розпочату OLE DB і ADO . Код ADO . NET працює в основному в керованому оточенні . NET . Це дозволяє значно підвищити продуктивність програм.

Архітектура ADO.NET



Постачальники даних. NET Framework



Основні об'єкти частини, що підключається.

- **Connection** – клас, який дозволяє встановити підключення до джерела даних.
- **Transaction** – клас, який надає транзакцію для цієї команди
- **Command** – клас, що представляє команду, що виконується в базовому джерелі даних.
- **Parameter** – клас, який надає параметри для зазначеної команди.
- **DataReader** – клас, що є еквівалентом конвеєрного курсору з можливістю тільки читання даних у прямому напрямку.

Класи автономної моделі ADO.NET

- 1) **DataSet** є ядром автономного режиму доступу до даних в ADO.NET.
- 2) **DataTable** . Найбільше цей клас схожий таблицю БД. Він складається з об'єктів DataColumn, DataRow, що являють собою рядки та стовпці.
- 3) **DataView** – об'єкт уявлень бази даних.
- 4) **DataRelation** - клас дозволяє задавати відносини між різними таблицями, з допомогою яких можна перевіряти відповідність даних із різних таблиць.

Підключення бази даних

Об'єкт Connection

Рядок з'єднання – рядок, що складається з пар ім'я-значення, що містить відомості про ініціалізацію, що передаються у вигляді параметра від додатка до джерела даних

```
class Program
{
    static void Main(string[] args)
    {
        string connectionString =
@"Data Source=.\SQLEXPRESS; - вказує на назву сервера
Initial Catalog=usersdb; - Вказує на назву БД на сервері
Integrated Security=True "; - встановлює автентифікацію
        SqlConnection connection = new SqlConnection (conStr);
    }
}
```

Пул з'єднань

Як правило, у програмі використовується одна або кілька одних і тих самих конфігурацій підключень. І щоб розробнику не доводилося створювати кілька разів у кодї програми фактично одне і теж підключення, в ADO.NET використовується механізм пула підключень.

Пул підключень дозволяє використовувати раніше створені підключення. Коли менеджер підключень, який керує пулом, отримує запит на відкриття нового підключення за допомогою методу `Open()`, він перевіряє всі підключення пула.

Пул можуть поміщатися тільки з'єднання з однаковою конфігурацією. ADO.NET підтримує кілька пулів одночасно, по одному для кожної конфігурації.

Отримання даних. Об'єкт SqlCommand

SqlCommand – об'єкт частини технології ADO.NET, що підключається, що дозволяє виконувати інструкції T-SQL над джерелом даних



Способи створення об'єкта SqlCommand:

1. Використання конструктора за умовчанням

```
SqlCommand cmd = New SqlCommand(); // Створюємо екземпляр  
конструктора SqlCommand  
cmd.Connection = connection ; // створюємо з'єднання  
cmd.CommandText = "Some T-SQL Command" ; // створюємо текст  
команди
```

2. Використання методу CreateCommand() об'єкта SqlConnection

```
cmd = connection.CreateCommand();  
cmd.CommandText = "Some T-SQL Command";
```

3. Використання перевантаження конструктора з двома параметрами

```
cmd = new SqlCommand("Some T-SQL Command", connection);
```

Методи SqlCommand:

- **ExecuteNonQuery ()**: просто виконує sql-вираз і повертає кількість змінених записів. Підходить для sql-виразів INSERT, UPDATE, DELETE.
- **ExecuteReader ()**: виконує sql-вираз і повертає рядки з таблиці. Підходить для sql-виразу SELECT.
- **ExecuteScalar ()**: виконує sql-вираз і повертає одне скалярне значення, наприклад, число. Підходить для sql-виразу SELECT у парі з однією із вбудованих функцій SQL, як, наприклад, Min, Max, Sum, Count.

Об'єкт SqlDataReader

Об'єкт DataReader дозволяє переглядати табличні дані, які команда повертає. Отримати об'єкт DataReader можна за допомогою методу ExecuteReader() об'єкту Command

```
SqlDataReader reader = cmd.ExecuteReader();
```

SqlDataReader має ряд методів, властивостей та індексаторів:

Read () – цей метод повертає значення true або false в залежності від того, чи досягнуто кінець набору рядків і при кожному його виклику переміщується до наступного рядка.

FieldCount - властивість, що дозволяє отримати кількість полів у рядків, що прийшли від джерела.

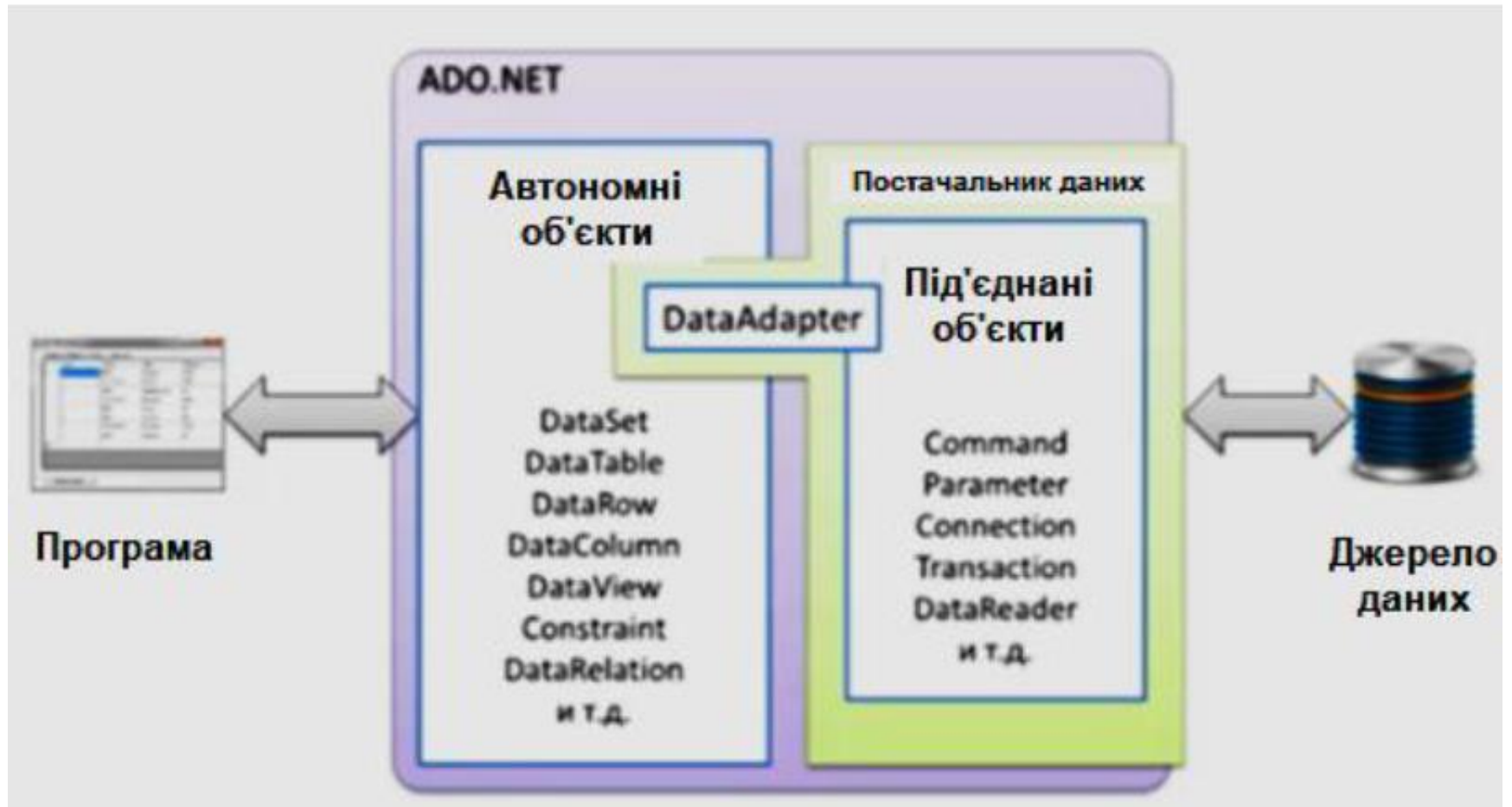
GetName(int index) - метод, що дозволяє отримати ім'я поля за індексом .

Асинхронне виконання команд.

При створенні UI додатків, які використовують об'єкт SqlCommand, виконання команд слід проводити асинхронно, щоб уникнути блокування інтерфейсу користувача. Для цього об'єкт Command має методи асинхронного виконання команд. Усі методи асинхронного виконання команд мають префікс Async.

```
await cmd.ExecuteNonQueryAsync();  
await cmd.ExecuteReaderAsync();  
await cmd.ExecuteScalarAsync();
```

Автономна частина архітектури ADO . NET

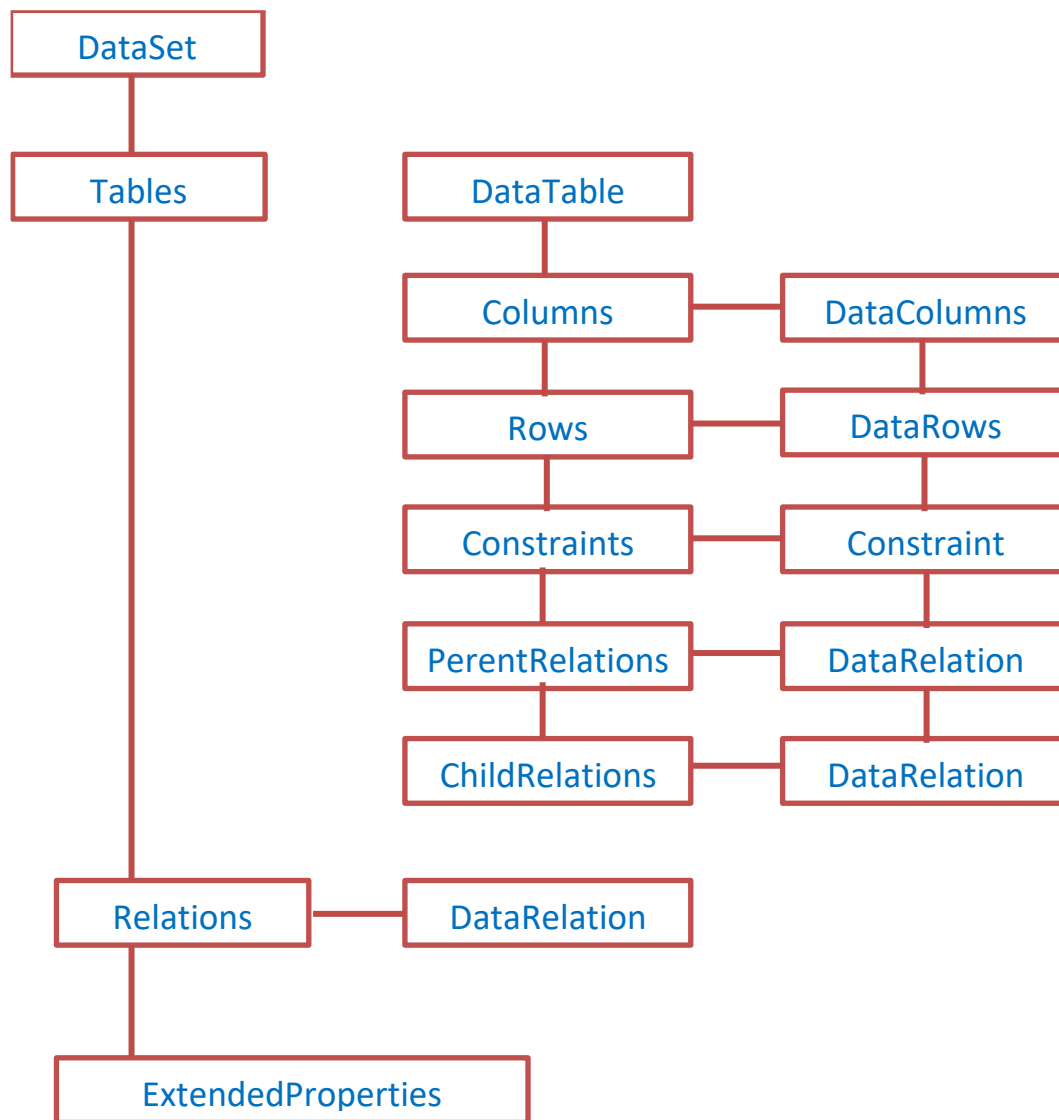


- Набір даних, представлений об'єктом DataSet, складається з таблиць (об'єкти DataTable), які у свою чергу складаються зі стовпців (об'єкти DataColumn) та рядків (об'єкти DataRow). Метадані існують лише на рівні стовпців; іншими словами, кожен стовець DataColumn може мати дані лише одного типу.
- Об'єкти DataTable можуть мати первинний ключ, а також зовнішні ключі, відносини та унікальні обмеження.
- Об'єкт DataSet може враховувати реєстр імен таблиць і стовпців, що входять до його складу, а також даних, що містяться в них.
- Підтримуються обчислювані стовпці.
- Існують методи для виконання множинних змін так, щоб траплялися всі зміни або вони не траплялися взагалі.

Відмінності класу DataSet від реляційних баз даних:

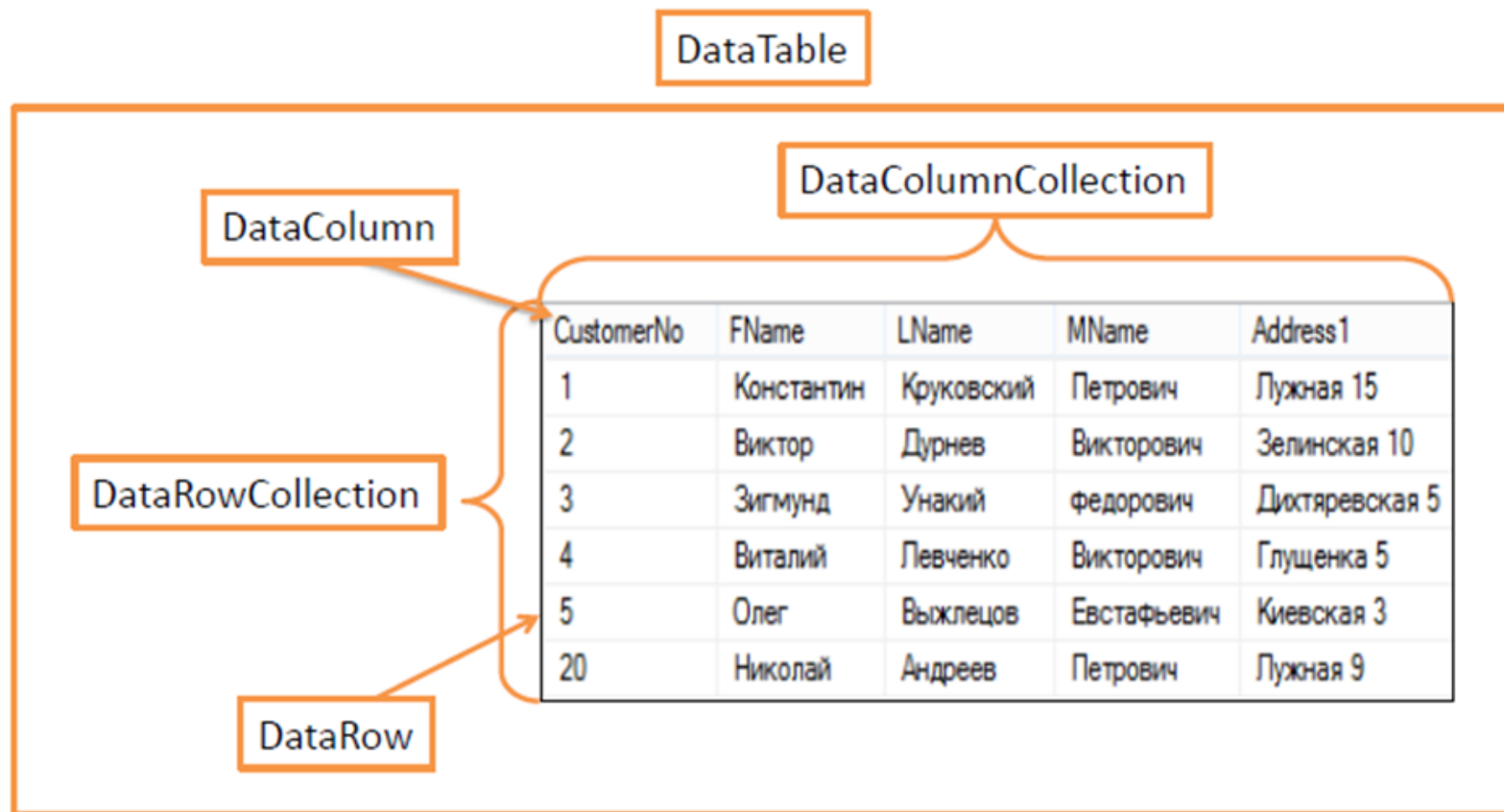
- не реалізовано концепцію автентифікацій чи авторизації;
- підтримується лише підмножина операторів SQL-подібної мови запитів, і ці оператори можуть застосовуватися лише вибірки наборів даних;
- не реалізовано концепцію ізоляції транзакцій або блокування записів у розрахованому на багато користувачів середовищі;
- деякі з властивостей об'єкта DataSet мають пряме відношення до відображення XML.

Об'єктна модель класу DataSet



Класи DataColumn, DataRow та DataTable

Объект DataTable



Об'єкт DataColumn

Об'єкт DataColumn є окремим стовпець таблиці. При створенні об'єкта DataColumn слід вказати ім'я колонки та тип даних, які міститиме колонка

```
DataColumn firstColumn = New DataColumn("First Column",  
typeof(int));
```

```
DataColumnCollection columnCollection = table.Columns;  
  
columnCollection.AddRange(new DataColumn[]{firstColumn});
```

Об'єкт DataRow

Об'єкт DataRow є окремим рядком таблиці.

Рядок може бути створена на основі схеми існуючої таблиці.

Для створення рядків об'єкта DataTable є метод NewRow(), що повертає об'єкт DataRow з полями, аналогічними стовпцям таблиці.

```
DataRow newRow = table.NewRow();
```

Ключі, відносини та обмеження

Первинний ключ має бути унікальним. Якщо один стовпець таблиці буде позначений як первинний ключ (властивість `PrimaryKey`), значення властивості цього стовпця `AllowDBNull` автоматично стане рівним `false`, проте якщо для стовпця буде просто встановлено властивість `Unique`, значення властивості `AllowDBNull` не стане рівним `false`.

Клас `Constraint` - це абстрактний базовий клас, що має два спеціалізовані похідні класи: `ForeignKeyConstraint` і `Uniqueconstraint`. Клас `Uniqueconstraint` призначений реалізації концепції унікальності, визначеної у стандарті SQL -92. У свою чергу об'єкт `ForeignKeyConstraint` описує "дочірню" половину відносини "батько-нащадок" між двома таблицями `DataTable`.

Властивості об'єкту DataColumn

Для перевірки даних лише на рівні таблиць об'єкт DataColumn надає такі характеристики:

- 1) Readonly – повертає чи задає значення, що вказує на допустимість зміни стовпця після додавання рядка до таблиці.
- 2) AllowDBNull - повертає або задає значення, що вказує на допустимість нульових значень у цьому стовпці для рядків, що належать таблиці.
- 3) MaxLength – повертає чи задає максимальну довжину текстового стовпця.
- 4) Unique – повертає або задає значення, що показує, чи мають значення у кожному рядку стовпця бути унікальними.

Обмеження об'єкту DataTable

Для об'єкта DataTable можна встановити такі обмеження:

- 1) UniqueConstraint – надає обмеження на набір стовпців, у яких усі значення мають бути унікальними. Слід користуватися цим обмеженням у разі, коли необхідно гарантувати унікальність комбінацій значень різних полів таблиці
- 2) PrimaryKey – особливий вид обмеження унікальності. Первинний ключ таблиці може бути лише один
- 3) ForeignKeyConstraint – обмеження, що гарантує, що не можна створити рядок у дочірній таблиці, яка посилається на неіснуючий рядок батьківської таблиці.

Стан та версії рядків DataRow

Кожен об'єкт DataRow має властивість Rowstate . Воно є стан рядка DataRow по відношенню до колекції рядків DataTable . Rows .

Властивість RowState може приймати елементи з переліку DataRowState .

Елементи переліку DataRowState:

- 1) Detached – рядок був створений, але не є частиною будь-якої DataRowCollection. Об'єкт DataRow має цей стан одразу після свого створення та перед додаванням до колекції, а також якщо він був видалений з колекції .
- 2) Unchanged – рядок не змінено з моменту останнього виклику AcceptChanges ().
- 3) Added – рядок був доданий у колекцію DataRowCollection і метод AcceptChanges() не був викликаний .
- 4) Deleted – рядок було видалено з допомогою методу Delete() об'єкта DataRow .
- 5) Modified – рядок було змінено і метод AcceptChanges() не було викликано.

Для того, щоб застосувати всі зміни, зроблені з рядками таблиці після передачі відкладених змін, об'єкт DataRow має метод AcceptChanges.

На практиці іноді виникає потреба подивитися значення зміненого рядка до зміни або ж подивитися дані рядка, підготовленого до видалення.

Для цього об'єкт DataRow має навантаження індексатора, яка приймає як другий індекс елемент перерахування DataRowVersion .

Елементи переліку DataRowVersion :

- 1) Current – дозволяє дізнатися про поточне значення рядка.
- 2) Original - дозволяє дізнатися вихідне значення рядка.
- 3) Proposed – дозволяє дізнатися передбачуване значення поля (дійсно лише за редагування поля з допомогою методу BeginEdit()).
- 4) Default – версія за промовчанням для рядка. Для DataRowState Added, Modified або Deleted версією за замовчуванням є Current. Для значення DataRowState Detached стандартною версією є Proposed .