Міністерство освіти і науки України
Криворізький коледж Національного авіаційного університету

Курс лекцій

з предмету

“Об’єктно-орієнтоване програмування”

Спеціальність 5.05010301

м. Кривий Ріг

Лекция №

Тема: Работа с базами данных в приложениях Windows Forms

Вопросы, рассматриваемые на лекции

- **Основы ADO .NET**
- **класс** DataTable**;**
- **класс** DataColumns**;**
- **класс** DataRows**;**
- **методы и свойства;**
- **примеры задач.**

Отсоединенная модель программирования

- В ADO .NET используется модель доступа – доступ к отсоединенным данным. При этом соединение устанавливается лишь на то время, которое необходимо для проведения определенной операции над базой данных.
- Возникает возможность поддержки большего количества параллельно работающих пользователей за счет постепенного увеличения количества клиентских компьютеров. Это преимущество имеет особенно большое значение при создании Web-приложений.

Концепция доступа к данным в ADO .NET основана на использовании двух компонентов:

- НАБОРА ДАННЫХ (представляется объектом класса DataSet) со стороны клиента. Это локальное временное хранилище данных;
- ПРОВАЙДЕРА ДАННЫХ (представляется объектом класса DataProvider). Это посредник, обеспечивающий взаимодействие приложения и базы данных со стороны базы данных (в распределенных приложениях – со стороны сервера).

Провайдеры данных ADO.NET

- Несмотря на подчеркнутое значение отсоединенной модели программирования, для извлечения, обновления, вставки и удаления данных необходимо подключиться к физической базе данных. Программное обеспечение ADO.NET для подсоединения и взаимодействия с физической базой данных называется **провайдером** данных ADO.NET.
- ***Провайдер данных (data provider)*** — это управляемый код .NET, который эквивалентен провайдеру OLEDB или драйверу ODBC.
- Провайдер данных состоит из нескольких объектов, которые реализуют необходимую функциональность в соответствии с определениями своих классов и интерфейсов.

Провайдеры данных ADO.NET

Наиболее часто используются следующие провайдеры:

- SQL Server .NET Data Provider – предназначен для работы с базами данных Microsoft SQL Server;
- OLE DB.NET Data Provider – предназначен для работы с источниками данных SQL Server, Oracle, Access;
- ODBC.NET Data Provider.– обеспечивает доступ к источникам данных через их ODBC-драйвера;
- ORACLE.NET Data Provider – предназначен для работы с базой данных Oracle.

Провайдеры данных и пространства имен

Провайдер данных	Пространство имен провайдера
Microsoft SQL Server 7.0 и выше	System.Data.SqlClient
Oracle 8.1.6 и выше	System.Data.OracleClient
Поддержка SqIXml в SQL Server	System.Data.SqlXml
Любой источник данных ODBC	System.Data.ODBC
Любой источник данных OleDb	System.Data.OleDb

Провайдеры данных и пространства имен

Основные объекты, входящие в состав провайдера, - это Connection, Command, DataAdapter иDataReader.

Connection используется для установления соединения с источником данных, а также для управления транзакциями.

Command позволяет манипулировать данными источника, а также выполнять хранимые процедуры. При этом могут использоваться параметры для передачи данных в обоих направлениях.

DataAdapter служит связующим звеном между DataSet и источником данных. Он использует Command для выполнения команд SQL как для заполнения DataSet данными, так и для обратной передачи измененных клиентом данных к источнику. Для выполнения этих функций объект имеет 4 метода: SelectCommand, InsertCommand, UpdateCommand и Deletecommand.

DataReader представляет однонаправленный поток данных от источника только на чтение. Если приложение клиента не модифицирует данные и не требуется произвольная выборка данных, а достаточно их однократного просмотра, то использование DataReader вместо DataSet позволит сохранить ресурсы RAM и CPU, а также поднять быстродействие приложения.

Создание подключения к БД

1 шаг – создание объекта подключения

Объект подключения источника данных наследуется от класса DbConnection и получает уже готовую логику, реализованную в базовых классах провайдеров (OleDbConnection, SqlConnection, OracleConnection и т. д.).

Создание объектов для разных провайдеров:

```
OleDbConnection oledbconn = newOleDbConnection();
SqlConnection sqlconn = new SqlConnection();
OdbcConnection odbcconn = new OdbcConnection();
OracleConnection oracleconn = new OracleConnection();
```

SqlConnection.ConnectionString
http://msdn.microsoft.com/ru-ru/library/system.data.sqlclient.sqlconnection.connectionstring(v=vs.110).aspx

Свойства SqlConnection

Имя	Описание
ClientConnectionId	Идентификатор соединения последней попытки подключения, независимо от того, успешно ли выполнена попытка или завершилась ошибкой.
ConnectionString	Получает или задает строку, используемую для открытия базы данных SQL Server.
ConnectionTimeout	Получает время ожидания при попытке установки подключения, по истечении которого попытка подключения завершается и создается ошибка.
Database	Получает имя текущей базы данных или базы данных, которая будет использоваться после открытия подключения.
DataSource	Получает имя экземпляра SQL Server, к которому осуществляется подключение.
PacketSize	Получает размер сетевых пакетов (в байтах), используемых при взаимодействии с экземпляром SQL Server.
ServerVersion	Получает строку, содержащую версию экземпляра SQL Server, к которому подключается клиент.
State	Отображает состояние SqlConnection во время последней сетевой операции, выполненной по подключению.
StatisticsEnabled	Когда задано значение true, разрешает сбор статистических сведений для текущего подключения.
WorkstationId	Получает строку, определяющую клиента базы данных.

Свойства OleDbConnection

Имя	Описание
ConnectionString	Возвращает или задает строку, используемую для открытия базы данных.
ConnectionTimeout	Получает время ожидания при попытке установки подключения, по истечении которого попытка подключения завершается и создается ошибка.
Container	Возвращает контейнер IContainer, содержащий компонент Component.
Database	Получает имя текущей базы данных или базы данных, которая будет использоваться после открытия подключения
DataSource	Получает имя сервера или имя файла источника данных.
Provider	Получает имя поставщика OLE DB, указанное в выражении "Provider= " строки подключения.
ServerVersion	Получает строку, содержащую номер версии сервера, к которому подключается клиент.
Site	Получает или задает экземпляр ISite для компонента Component.
State	Получает текущее состояние подключения.

Значения свойства State (перечисление ConnectionState

Имя члена	Описание
Broken	Подключение к источнику данных разорвано. Это может произойти только после открытия подключения. Подключение в этом режиме может быть закрыто, а затем повторно открыто. (Это значение зарезервировано для будущих версий продукта.)
Closed	Подключение закрыто.
Connecting	Объект подключения подключается к источнику данных.
Executing	Объект подключения выполняет команду. (Это значение зарезервировано для будущих версий продукта.)
Fetching	Объект подключения получает данные. (Это значение зарезервировано для будущих версий продукта.)
Open	Подключение открыто.

Пример:

```
if (connect.State == ConnectionState.Open)
connect.Close();
```

Строка подключения

2 шаг – задание строки подключения

Для того, чтобы открыть подключение необходимо указать, какая информация необходима для выполнения этой задачи, например, имя сервера, идентификатор пользователя, пароль и т. Д.

Поскольку каждому целевому источнику подключения может понадобиться особый набор информации, позволяющей ADO.NET подключится к источнику данных, выбирают гибкий механизм указания всех параметров через строку подключения **ConnectionString**.

Строка подключения - это набор разделенных точкой с запятой пар атрибут-значений, определяющих, как следует устанавливать подключение к источнику данных. Хотя строки подключения имеют тенденцию выглядеть одинаково, возможные и обязательные атрибуты различаются в зависимости от поставщика данных и источника данных.

Строка подключения

Перечень некоторых атрибутов строк подключения для установления соединений:

- ✔ Provider - имя провайдера (только для OLE DB);
- ✔ Driver - ODBC-драйвер (только для ODBC);
- ✔ Connection Timeout - время в секундах для попыток установки соединения (по умолчанию 15 сек.)
- ✔ InitialCatalog - имя базы данных в SQLServer;
- ✔ Data Source - имя сервера или путь к базе данных Access;
- ✔ Password - пароль;
- ✔ User ID - логин;
- ✔ Workstation ID - имя рабочей станции или компьютера
- ✔ Integrated Security - true, если выбирается Windows Authentication, false - если SQL Authentication, sspi – эквивалентно true.

Строка подключения

Например, строка соединения при использовании провайдера OLE DB и подключении к базе данных lab.mdb будет выглядеть так:

string str = "Provider = Microsoft.Jet.OLEDB.4.0; DataSource = lab.mdb";

Теперь свойства ConnectionString объекта подключения oledbconn можно задать необходимое значение:

oledbconn.ConnectionString = str;

Эти действия можно выполнить с помощью одной строки, например:

oledbconn.ConnectionString = "Provider = Microsoft.Jet.OLEDB.4.0;

DataSource = lab.mdb ";

http://msdn.microsoft.com/ru-ru/library/system.data.sqlclient.sqlconnection.connectionstring(v=vs.110).aspx

Методы SqlConnection

Имя	Описание
BeginTransaction	Начинает транзакцию базы данных.
ChangeDatabase	Изменяет текущую базу данных для открытого подключения SqlConnection.
ChangePassword	Заменяет пароль SQL Server для пользователя, указанного в строке подключения, заданным новым паролем.
ClearAllPools	Очищает пул подключений.
ClearPool	Очищает пул подключений, связанный с заданным подключением.
Close	Закрытие подключения к базе данных. Рекомендуется использовать этот метод для закрытия любого открытого подключения.
CreateCommand	Создает и возвращает объект SqlCommand, связанный cSqlConnection.
Dispose()	Освобождает все ресурсы, используемые объектом Component.

Методы SqlConnection

Имя	Описание
EnlistDistributed Transaction	Выполняет присоединение указанной транзакции как распределенной транзакции.
EnlistTransaction	Выполняет присоединение указанной транзакции как распределенной транзакции.
GetSchema()	Возвращает информацию схемы для источника данных этого объекта SqlConnection. Дополнительные сведения о схеме см. Коллекции схем SQL Server.
GetSchema	Возвращает сведения схемы для источника данных этого объекта SqlConnection, используя указанную строку для имени схемы.
Open	Открывает подключение к базе данных со значениями свойств, определяемыми объектом ConnectionString.
ResetStatistics	Если сбор статистики разрешен, все значения сбрасываются в нуль.
RetrieveStatistics	Возвращает коллекцию пар имя-значение статистических данных на момент вызова метода.

Методы SqlConnection

3 шаг – открытие базы данных (для OleDb)

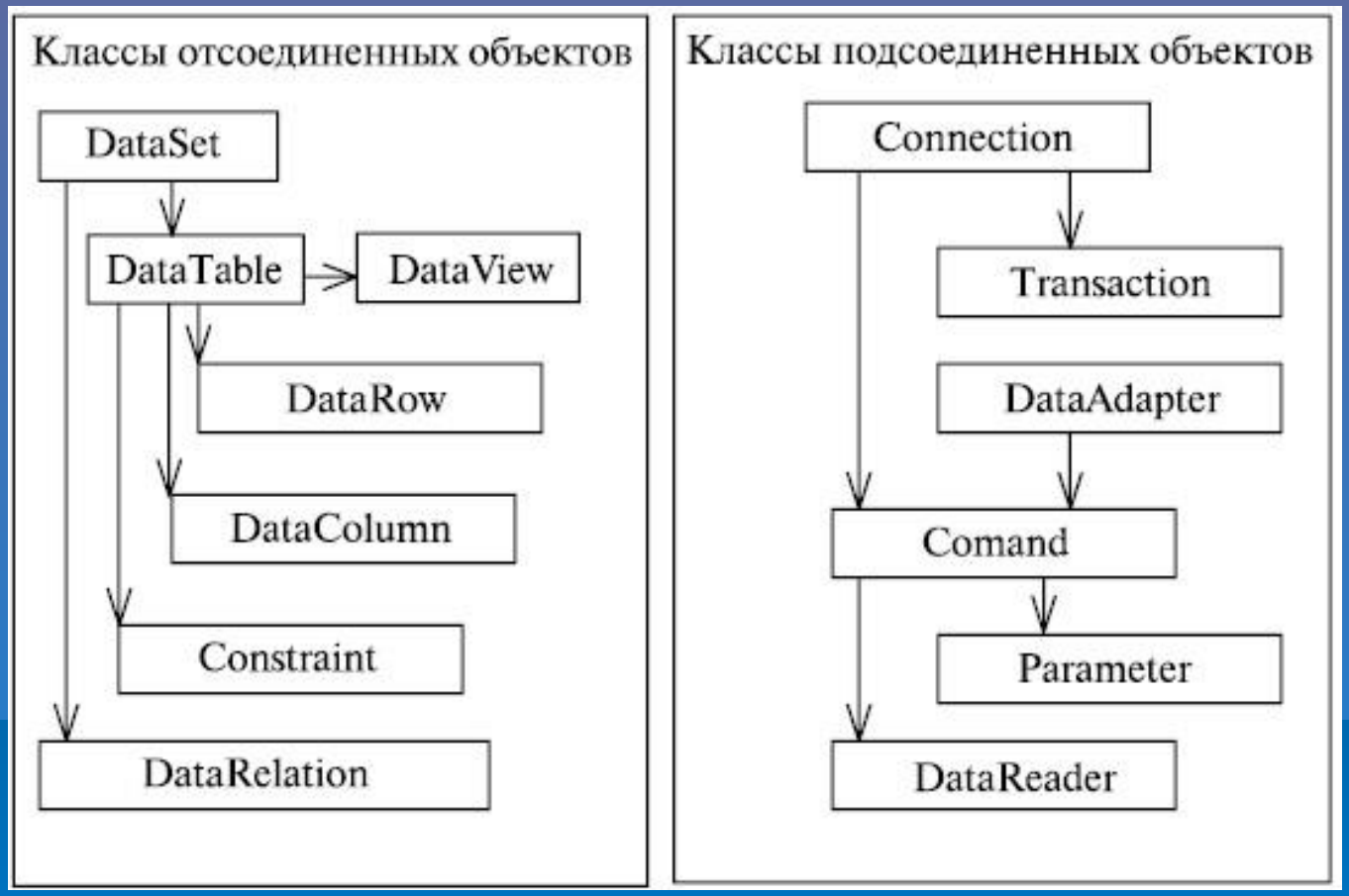
```
OleDbConnection oledbconn = new OleDbConnection();
oledbconn.ConnectionString=@"Provider=Microsoft.Jet.OLEDB.4.0;
DataSource=c:\1\lab.mdb";
try
{
oledbconn.Open();
MessageBox.Show( "З’єднання відчинено" );
// Виконання корисних дій
oledbconn.Close();
MessageBox.Show( " З’єднання зачинено " )
}
catch(System.Data.OleDb.OleDbException e1)
// Виникнення OleDbException
{
MessageBox.Show("З’єднання відчинено "+e1.Message);
}
catch (System.InvalidOperationException e2)
// Помилка відкриття бази даних
{
MessageBox.Show("З’єднання не відчинено " + e2.Message);
} }
```

Строка подключения

3 шаг – открытие базы данных (для Sql)

```
try
{
using (SqlConnection conn = new SqlConnection ( source ))
{
// Відкрити з'єднання
conn.Open () ;
// Зробити що-небудь корисне. Закрити з'єднання самостійно conn.Close () ;
}
}
catch (SqlExceptione)
{
//
```

ADO .NET. Объектная модель



- **Классы подсоединенных объектов** обеспечивают установление соединения с базой данных и управление базой со стороны приложения;
- **Классы отсоединенных объектов** обеспечивают сохранение, использование и преобразование полученной от базы данных информации на стороне приложения.

Свойства класса DataSet

Имя	Описание
CaseSensitive	Возвращает или задает значение, определяющее, учитывается ли регистр при сравнении строк в объектах DataTable.
Container	Возвращает контейнер для компонента.
DataSetName	Возвращает или задает имя текущего DataSet.
DefaultViewManager	Получает новое представление данных класса DataSet для осуществления фильтрации, поиска или перехода с помощью настраиваемого класса DataViewManager.
DesignMode	Возвращает значение, показывающее, находится ли компонент в настоящий момент в режиме разработки.
EnforceConstraints	Возвращает или задает значение, определяющее соблюдение правил ограничения при попытке совершения операции обновления.
Events	Возвращает список обработчиков событий, которые прикреплены к этому компоненту.
ExtendedProperties	Возвращает коллекцию настраиваемых данных пользователя, связанных с DataSet.
IsInitialized	Получает значение, указывающее, инициализирован ли набор данных DataSet.
Locale	Возвращает или задает сведения о языке, используемые для сравнения строк таблицы.
Namespace	Возвращает или задает пространство имен класса DataSet.
Prefix	Возвращает или задает префикс XML, который является псевдонимом пространства имен класса DataSet.
Relations	Возвращает коллекцию соотношений, связывающих таблицы и позволяющих переходить от родительских таблиц к

Свойства класса DataSet

Примеры :

1) DataColumn parentColumn = dataSet.Tables["Suppliers"].Columns["SupplierID"];

2) dataSet.Relations.Add(relation);

3)

```
private void CopyDataSet(DataSet dataSet)
{
DataSet copyDataSet;
copyDataSet = dataSet.Copy();
}
```

Методы класса DataSet

Имя	Описание
AcceptChanges	Сохраняет все изменения, внесенные в класс DataSet после его загрузки или после последнего вызова методаAcceptChanges.
BeginInit	Начинает инициализацию класса DataSet, используемого в форме или другим компонентом. Инициализация происходит во время выполнения.
Clear	Удаляет из класса DataSet любые данные путем удаления всех строк во всех таблицах.
Clone	Копирует структуру класса DataSet, включая все схемы, соотношения и ограничения объекта DataTable. Данные не копируются.
Copy	Копирует структуру и данные для DataSet.
CreateDataReader	Возвращает объект DataTableReader с одним результирующим набором для каждой последовательности DataTable в той же последовательности, в которой таблицы отображаются в коллекцииTables.
Dispose	Освобождает все ресурсы, используемые объектом MarshalByValueComponent.(Унаследовано отMarshalByValueComponent.)
EndInit	Завершает инициализацию объектаDataSet, который используется в форме или другим компонентом.Инициализация происходит во время выполнения.
Equals(Object)	Определяет, равен ли заданный объект текущему объекту. (Унаследовано отObject.)
Finalize	Позволяет объекту попытаться освободить ресурсы и выполнить другие операции очистки, перед тем как объект будет утилизирован в процессе сборки мусора.
GetChanges()	Получает копию класса DataSet, содержащую все изменения

Методы класса DataSet

Имя	Описание
Load(IDataReader, LoadOption, DataTable[])	Заполняет набор данных DataSetзначениями из источника данных с помощью предоставляемого объектаIDataReader
MemberwiseClone	Создает неполную копию текущего объекта Object. (Унаследовано от Object.)
Merge(DataRow[])	Осуществляет слияние массива объектовDataRow и текущего класса DataSet.
Merge(DataSet)	Осуществляет слияние указанного объекта DataSet и его схемы с текущим объектом DataSet.
Merge(DataTable)	Осуществляет слияние указанного объекта DataTable и его схемы с текущим объектом DataSet.
RejectChanges	Отменяет все изменения, внесенные в класс DataSet после его создания или после последнего вызова методаDataSet.AcceptChanges.
Reset	Очищает все таблицы и удаляет все связи, внешние ограничения и таблицы из DataSet. Для восстановления исходного состояния класса DataSetнеобходимо переопределить метод Resetв подклассах.

Пример создание объекта.

```
DataSet data=new DataSet("База_даних");
DataSet data=new DataSet();

DataColumn parentColumn = dataSet.Tables["Suppliers"].Columns["SupplierID"];
```

DataTable

Каждый объект DataTable представляет одну таблицу базы данных.

Таблица в каждый конкретный момент своего существования характеризуется:

- СХЕМОЙ таблицы,
- СОДЕРЖИМЫМ таблицы (информацией).

СХЕМА таблицы (структура объекта DataTable) определяется двумя наборами:

- множеством столбцов таблицы (набор DataColumns, состоящий из множества объектов DataColumn),
- множеством ограничений таблицы (набор Constraints, состоящий из множества объектов Constraint).

Свойства класса DataTable

Имя	Описание
ChildRelations	Получает коллекцию дочерних отношений для объекта DataTable.
Columns	Получает коллекцию столбцов, принадлежащих данной таблице.
DataSet	Получает класс DataSet, к которому принадлежит данная таблица.
DesignMode	Возвращает значение, показывающее, находится ли компонент в настоящий момент в режиме разработки.
Events	Возвращает список обработчиков событий, которые прикреплены к этому компоненту.
HasErrors	Получает значение, указывающее наличие ошибок в строках таблиц класса DataSet, к которому принадлежат таблицы.
IsInitialized	Получает значение, указывающее, инициализирована ли таблица DataTable.
Locale	Возвращает или задает сведения о языке, используемые для сравнения строк таблицы.
MinimumCapacity	Возвращает или задает начальный размер таблицы.
Namespace	Возвращает или задает пространство имен для представления данных объекта DataTable в формате XML.
ParentRelations	Получает коллекцию родительских отношений для объекта DataTable.
Prefix	Возвращает или задает пространство имен для представления данных объекта DataTable в формате XML.
PrimaryKey	Возвращает или задает массив столбцов, которые являются столбцами первичного ключа для таблицы данных.

Свойства класса DataTable

Примеры :

1)
```
private void ClearDataSet(DataSet dataSet)
{
foreach(DataTable table in dataSet.Tables)
{
Console.WriteLine(table.TableName +
"Rows.Count = " +
table.Rows.Count.ToString());
}
dataSet.Clear();
}
```

2)
```
private void PrintValues(DataTable table)
{
 foreach(DataRow row in table.Rows)
{
 foreach(DataColumn column in table.Columns)
 { Console.WriteLine(row[column]); }
} }
```

3)
```
private void GetTableNames(DataSet
dataSet)
 {
foreach(DataTable table in dataSet.Tables)
{
Console.WriteLine(table.TableName);
 }
}
```

4)
```
table.Columns.Add(column);
```

5)
```
DataSet ds = new DataSet();
DataTable tablica = new
DataTable("Products");
ds.Tables.Add(tablica);
```
или
```
DataSet ds = new DataSet();
DataTable tablica =
ds.Tables.Add("Products");
```

События класса DataTable

Имя	Описание
ColumnChanged	Происходит после изменения значения указанного объекта DataColumn в DataRow.
ColumnChanging	Происходит при изменении значения указанного объекта DataColumn и DataRow.
Disposed	Добавляет обработчик событий, чтобы прослушивать событие Disposed для компонента. (Унаследовано от MarshalByValueComponent.)
Initialized	Происходит после инициализации таблицы DataTable.
RowChanged	Происходит после успешного изменения DataRow.
RowChanging	Происходит при изменении объекта DataRow.
RowDeleted	Происходит после удаления строки таблицы.
RowDeleting	Происходит перед удалением строки таблицы.
TableCleared	Происходит после очистки DataTable.
TableClearing	Происходит, когда очищается таблица DataTable.
TableNewRow	Происходит, когда вставляется новая строка DataRow.

Класс DataColumn

Имя	Описание
AllowDBNull	Возвращает или задает значение, указывающее на допустимость нулевых значений в этом столбце для строк, принадлежащих таблице.
AutoIncrement	Возвращает или задает значение, показывающее, увеличивать ли автоматически значение столбца для новых строк, добавляемых в таблицу.
AutoIncrementSeed	Возвращает или задает начальное значение для столбца, свойство которого AutoIncrement установлено на true. Значение по умолчанию — 0.
AutoIncrementStep	Возвращает или задает инкремент для столбца, свойство AutoIncrement которого установлено на true.
Caption	Возвращает или задает заголовок для столбца.
ColumnMapping	Возвращает или задает MappingType столбца.
ColumnName	Возвращает или задает имя столбца в DataColumnCollection.
Container	Возвращает контейнер для компонента. (Унаследовано отMarshalByValueComponent.)
DataType	Возвращает или задает тип данных, хранимых в столбце.
DateTimeMode	Возвращает или задает DateTimeMode столбца.
DefaultValue	Возвращает или задает значение по умолчанию для столбца при создании новых строк.
Expression	Возвращает или задает выражение, используемое для фильтрации строк, расчета значений в столбце либо создания составного столбца.
MaxLength	Возвращает или задает максимальную длину текстового

Класс DataColumn

Имя	Описание
Ordinal	Возвращает позицию (с нуля) столбца в коллекции DataColumnCollection.
ReadOnly	Возвращает или задает значение, указывающее на допустимость изменения столбца после добавления строки в таблицу.
Table	Получает DataTable, к которому принадлежит столбец.
Unique	Возвращает или задает значение, показывающее, должны ли значения в каждой строке столбца быть уникальными.