

УДК 371.694:004:658.336:656.7.071(045)

**В. М. Синеглазов**, д-р техн. наук, проф.,  
**К. А. Хоменко**

## ОРГАНИЗАЦИЯ СИСТЕМЫ КОНТРОЛЯ ВЕРСИЙ БАЗЫ ДАННЫХ

Институт аэрокосмических систем управления НАУ, e-mail: [iasy@nau.edu.ua](mailto:iasy@nau.edu.ua)

*Рассмотрена задача организации системы контроля версий базы данных, используемой в функциональном тренажере. Для решения поставленной задачи предложено использовать программный продукт Database project.*

**Ключевые слова:** база данных, контроль версий, скрипт, бекап, Visual Studio.

**Введение.** Неотъемлемым этапом разработки функциональных тренажёров является проектирование и наполнение базы знаний. В большинстве случаев для тестирования программного обеспечения тренажёра необходимо иметь несколько наборов тестовых данных. С этой целью нужно хранить копии баз с различными данными или скрипты их наполнения. Достаточно типичной является ситуация, когда база данных за время своего существования претерпевает ряд изменений. При этом важно иметь не только последнюю версию, но и несколько предыдущих. В простейшем случае можно просто хранить несколько вариантов резервных копий, соответствующим образом их нумеруя, что является не эффективным.

**Обзор методов и постановка задачи.** Системы контроля версий значительно упрощают этот процесс, однако часто используются нерационально в процессе разработки. Очень распространена практика хранить версии базы в виде бекапов, это приводит к быстрому росту размера репозитория. Более прогрессивный метод подразумевает хранение скриптов создания схемы базы и сами данные. У такого подхода есть ряд преимуществ:

- возможность слияния скриптов, предоставленных несколькими разработчиками;
- отсутствует необходимость заливать в систему контроля версий тяжеловесный бекап;
- значительно упрощается отслеживание изменений других разработчиков команды, к примеру, через почтовую нотификацию;
- можно легко организовать хранение нескольких словарей для одной и той же схемы базы данных: тестовый и реальный.

Традиционные системы управления версиями используют централизованную модель, когда имеется единое хранилище документов, управляемое специальным сервером, который и выполняет большую часть функций по управлению версиями. Пользователь, работающий с документами, должен сначала получить нужную ему версию документа из хранилища; обычно создаётся локальная копия документа, так называемая «рабочая копия». Можно получить последнюю версию или любую из предыдущих, которая выбирается по номеру версии или дате создания, иногда и по другим признакам. После того, как в документ внесены нужные изменения, новая версия помещается в хранилище. В отличие от простого сохранения файла предыдущая версия не стирается, а остаётся в хранилище и в любое время может быть оттуда получена. Сервер может использовать дельта-компрессию – способ хранения документов, при котором сохраняются только изменения между последовательными версиями, что позволяет уменьшить объём хранимых данных. Поскольку обычно наиболее востребованной является последняя версия файла, система может при сохранении новой версии сохранять её целиком, заменяя в хранилище последнюю ранее сохранённую версию на разницу между этой и последней версией. Некоторые системы (например, Clear Case) поддерживают сохранение версий обоих видов: большинство версий сохраняется в виде дельт, но периодически (по специальной команде администратора) выполняется сохранение версий всех файлов в полном виде. Такой подход обеспечивает максимально полное восстановление истории в случае повреждения репозитория.

**Метод решения поставленной задачи.** Одним из наиболее приемлемым инструментов является Database project, который является частью *Visual Studio*, начиная с 2010 версии (к 2005 и 2008 нужно устанавливать специальное дополнение).

Загрузив Visual Studio 2010, перейдя к пункту меню File -> New, можно увидеть следующее окно (рис. 1).

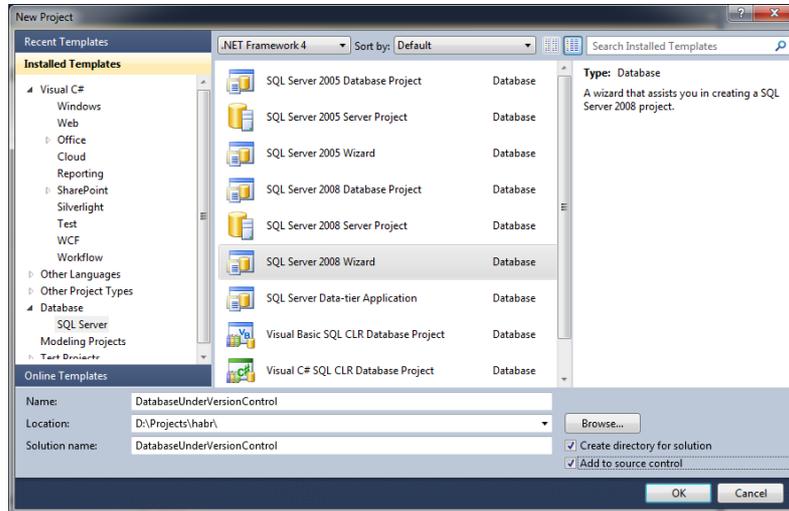


Рис. 1. Создание нового проекта

Здесь можно выбрать или пустой проект и позднее импортировать существующую базу данных, или конструктор, который проведет через все этапы импорта объектов базы еще на этапе создания проекта.

Изменив имя проекта и нажав на «ОК», попадаем на форму конструктора, которая содержит перечень этапов конструктора. Просто нажав «Далее» получаем форму конфигурации, изображенную на рис. 2.

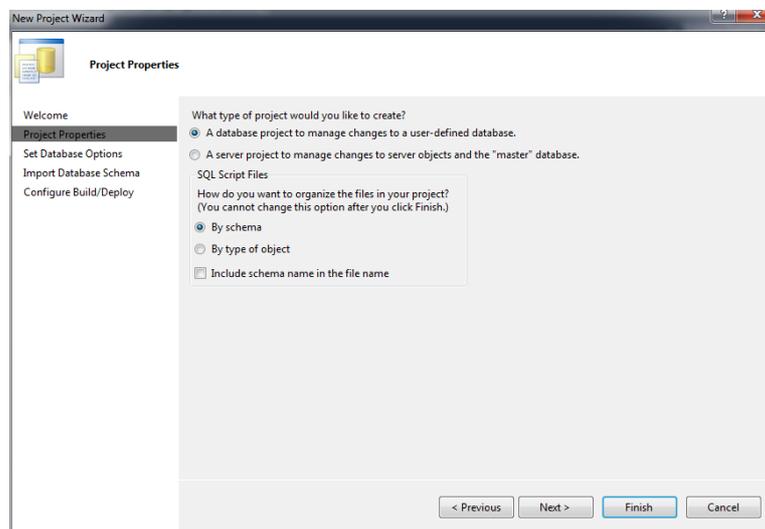


Рис. 2. Форма конфигурации

В этом окне можно выбрать опции, такие как: создание user database или SQL Server project for master database (выбрав первую, создаем пользовательскую базу данных), какой тип скриптовых файлов будет использоваться: создаваемые по имени схемы или по типам объектов. Выбираем *By schema* и нажимаем «Далее» (рис. 3).

В окне, показанном выше, можно определить настройки уровня базы, такие как ANSI paddings, ANSI nulls, Arithmetic abort, database default collation и т. д. Нажимаем «Далее» и получаем на рис. 4.

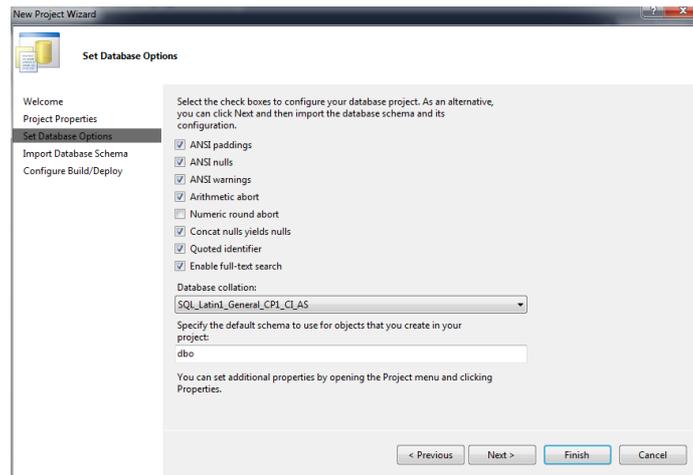


Рис. 3. Настройки уровня базы

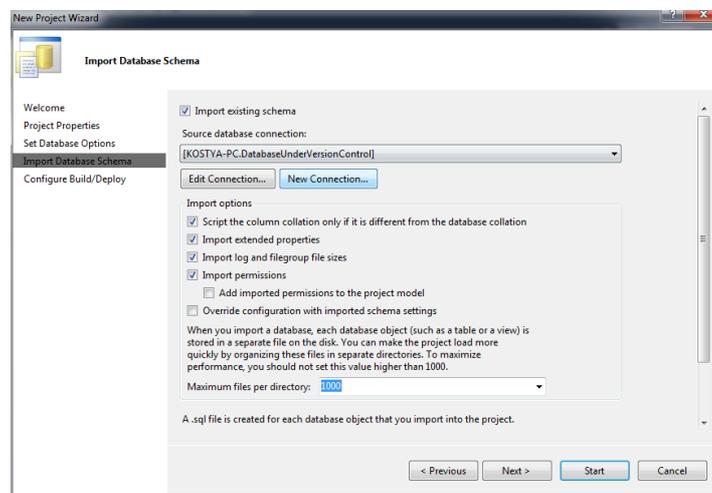


Рис. 4. Строка подключения к базе данных

На этом экране определяется строка подключения к базе данных, для которой надо сгенерировать скрипты. Помимо этого, можно определить различные опции импорта и максимальное количество файлов со скриптами (каждый объект базы данных будет храниться в отдельном файле) в одной папке. Нажав «Далее», попадаем на форму настройки развертывания базы (рис. 5).

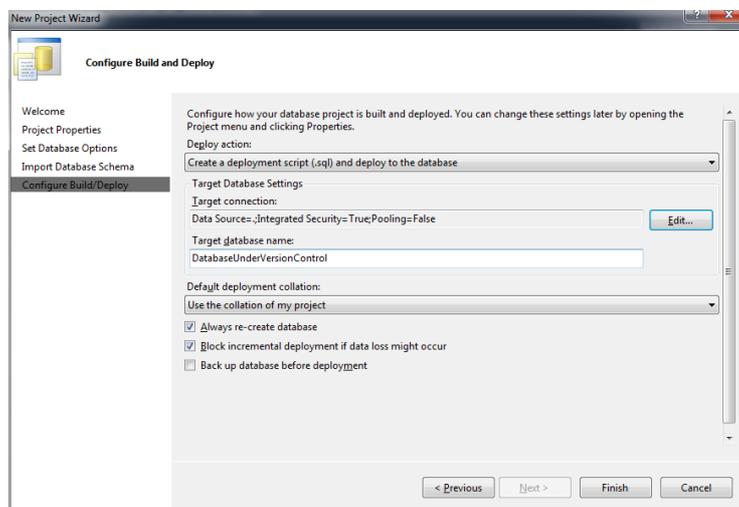


Рис. 5. Форма настройки развертывания

Здесь можно задать строку подключения к серверу, на котором будет развернута база данных, и ее имя. Нажимая «Финиш», получим проект с отдельными файлами для каждого объекта базы.

Еще одна удобная возможность Visual Studio 2010 – переключение между схемой (Schema View) и обозревателем сборки (Solution Explorer). По двойному клику на любой объект, находясь в Schema View, среда открывает скрипт для соответствующего объекта (рис. 6).

Таким образом, получена первая версия схемы нужной базы и можно «залить» ее в репозиторий.

**Развертывание проекта.** Прежде чем развертывать проект, необходимо удостовериться, что он правильно сконфигурирован. Для этого найдем свойства проекта, нажав правой кнопкой мыши на проекте -> Properties (рис. 7).

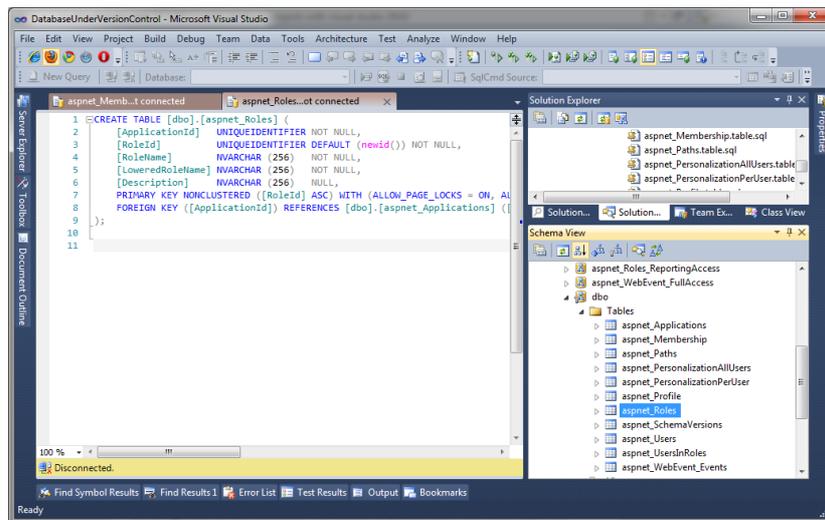


Рис. 6. Просмотр сформированных скриптов

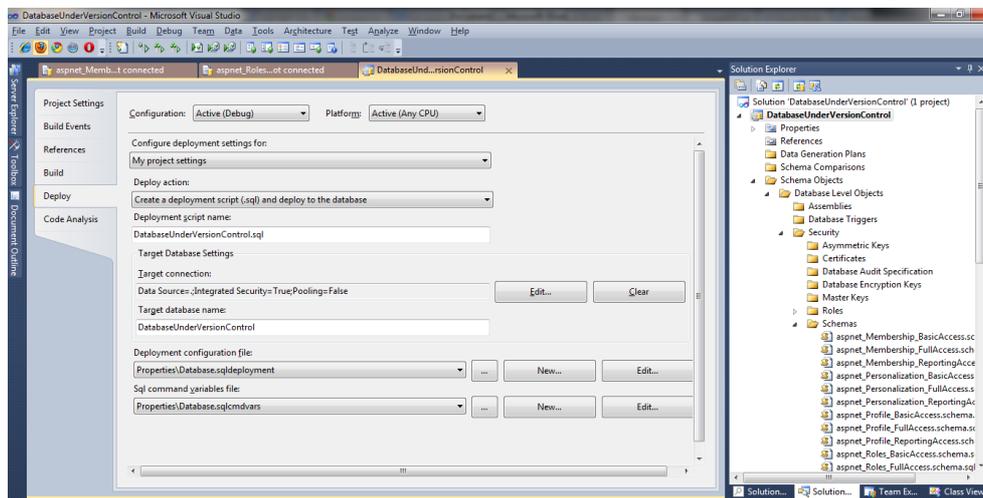


Рис. 7. Окно определяющее свойства проекта

Здесь можно указать, нужно просто создавать скрипт развертывания (Create a deployment script (.sql)) или сразу же выполнять его (Create a deployment script (.sql) and deploy to the database).

**Примечание.** С этими опция нужно быть предельно осторожными, чтобы не перетереть локальные изменения в базе.

После того, как вся настройка закончена, чтобы выполнить развертывание, нажимаем Build – Deploy Solution.

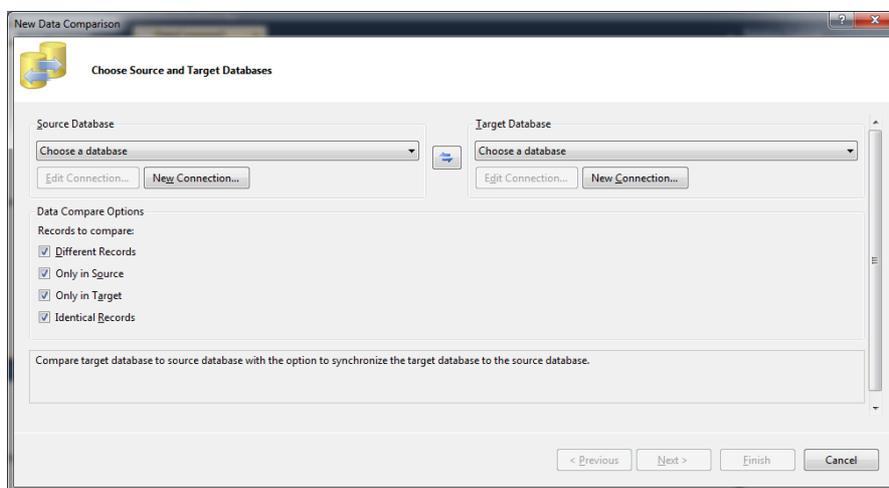
**Хранение словарей (данных) в Database project.** Для полноценной версионности базы не хватает скрипта, который будет хранить данные (к примеру, роли пользователей, метаданные и т. д.). Чтобы сформировать такой скрипт, воспользуемся возможностью Visual Studio 2010 сравнивать данные (Data Compare).

Для того чтобы получить скрипт с данными, нужно сравнить пустую базу с эталонной (со словарями). Для этого развернем (deploy) нужный проект, используя инструкцию, описанную выше, но перед этим изменив имя базы на DataBase Empty.

Далее, нажав на Data – Data Compare – New Data Compare, студия откроет окно с настройками сравнения (рис. 8):



а



б

Рис. 8. Окно с настройками сравнения: а – меню создание нового сравнения; б – настройка сравниваемых баз данных

В рассматриваемом случае Source Database – это эталонная база, которая содержит все данные, а Target Database – DataBase Empty (рис. 9).

Object (check to include in update)	Different Records	Only in Source	Only in Target	Identical Records
11 tables and/or views were compared.				
Source: [KOSTYA-PC.DatabaseUnderVersionControl] Target: [KOSTYA-PC.DatabaseUnderVersionControlEmpty]				
Tables				
[dbo].[aspnet_Applications]	0	0	0	0
[dbo].[aspnet_Membership]	0	0	0	0
[dbo].[aspnet_Paths]	0	0	0	0
[dbo].[aspnet_PersonalizationAllUsers]	0	0	0	0
[dbo].[aspnet_PersonalizationPerUser]	0	0	0	0
[dbo].[aspnet_Profile]	0	0	0	0
[dbo].[aspnet_Roles]	0	0	0	0
[dbo].[aspnet_SchemaVersions]	0	6 (Add 6)	0	0
[dbo].[aspnet_Users]	0	0	0	0
[dbo].[aspnet_UsersInRoles]	0	0	0	0
[dbo].[aspnet_WebEvent_Events]	0	0	0	0
Views				

Рис. 9. Результат сравнения

Результат сравнения сохраним в файле под именем Data.sql в папке Post-Deployment (рис. 10):

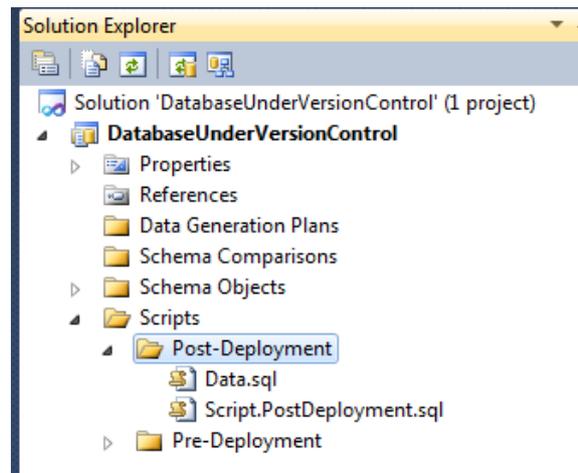


Рис. 10. Результаты сравнения пустой базы данных с эталонной в виде деревьев файлов

Теперь добавим следующую строку в Script.Post-Deployment.sql, для того, чтобы студия выполняла скрипт наполнения после каждого развертывания (deploy) (рис. 11).

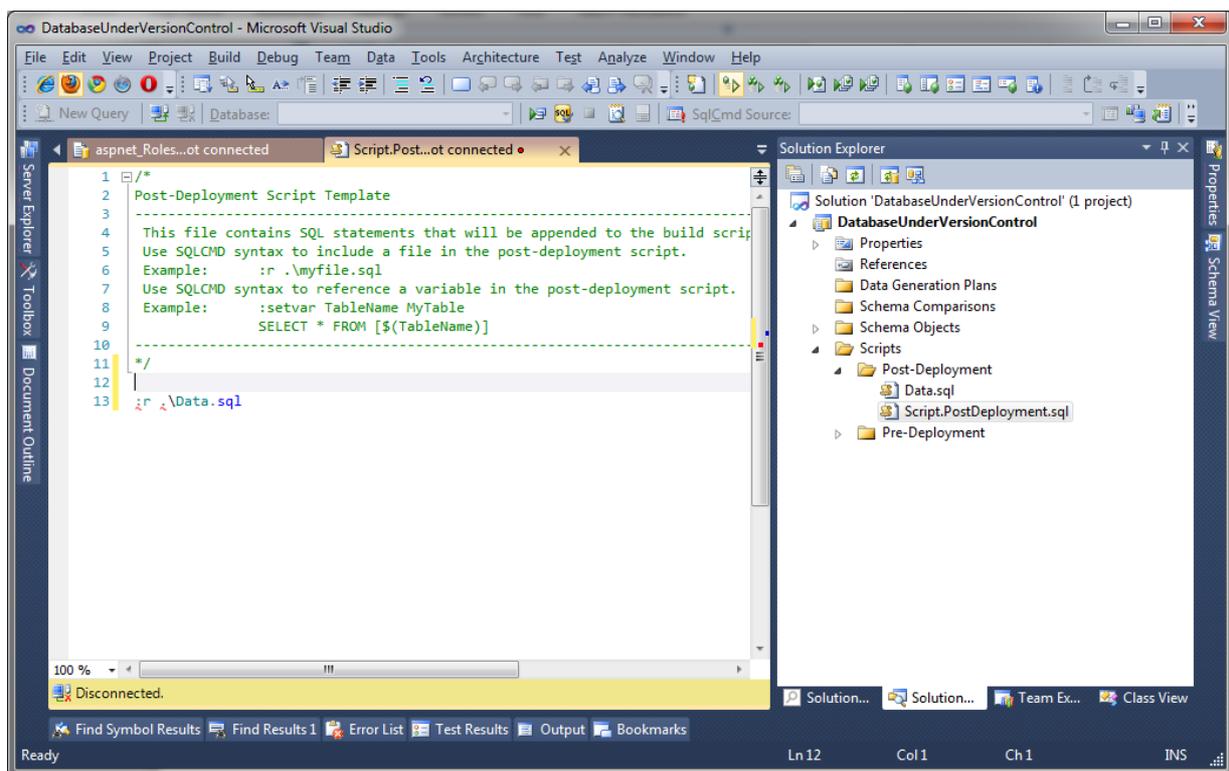


Рис. 11. Строка вызова скрипта наполнения данных

Таким образом, можно добавить множество скриптов с тестовыми данными и у каждого разработчика будет возможность развернуть свою тестовую базу в один клик.

Для дальнейшего внесения изменений в проект нужно снова использовать Schema Compare и Data Compare.

**Выводы.** Предложен эффективный подход построения системы контроля версий базы данных на основе использования программного продукта Database project с детальным рассмотрением последовательности операций по его применению для решения поставленной задачи.

### Список литературы

1. Рэндал Л. Шварц, Брайан Д. Фой и Том Феникс Perl: изучаем глубже. – М.: О’Reilly, «Символ», 2007. – ISBN 978-5-93286-093-9.
2. Линкольн Д. Штайн Разработка сетевых программ на Perl / Линкольн Д. Штайн: пер. с англ. – М.: Изд-во «Вильямс». – 2001. – 752 с.
3. Кузнецов М. MySQL 5 / М. Кузнецов, И. Симдянов. – СПб.: БХВ-Петербург, 2006. – С. 1024. – ISBN 5-94157-928-4.
4. Поль Дюбуа MySQL= MySQL / Поль Дюбуа. – 3-е изд. – М.: Вильямс, 2006. – 1168 с. – ISBN 0-672-32673-6.
5. Немеет Э. Руководство администратора Linux. Установка и настройка= Linux Administration Handbook / Э. Немеет, Г. Снайлер, Т. Хейн – 2-е изд. – М.: Вильямс, 2007. – 1072 с. – ISBN 0-13-148004-9 Официальный сайт Фрейморка Ext <http://extjs.com>.

В. М. Синеглазов, К. А. Хоменко

#### **Організація системи контролю версій бази даних**

Розглянуто завдання організації системи контролю версій бази даних, використовуваної у функціональному тренажері. Для рішення поставленого завдання запропоновано використати програмний продукт Database project.

V. M. Sineglazov, K. A. Khomenko

#### **Organizing the version control process for the data base**

It is described the process of organizing version control system for the data base which is used for developing functional trainer. It is suggested to use software service such as Database project to solve the issue.