



1С-Битрикс: Управление сайтом

Пример миграции базы сайта с MySQL на SQL
Server Express



Содержание

Введение.....	3
Глава 1. Предварительные операции	4
УСТАНОВКА 1С-БИТРИКС В ВАРИАНТЕ MYSQL	4
УСТАНОВКА SQL EXPRESS.....	4
УСТАНОВКА 1С-БИТРИКС В ВАРИАНТЕ SQL EXPRESS	11
Начало установки.....	11
Создание базы данных	12
Задание логина и пароля.....	15
Завершение установки	18
Глава 2. Сравнение структур баз	19
СОЗДАНИЕ СЕРВЕРА	19
СОСТАВ ТАБЛИЦ	20
НАБОР ПОЛЕЙ.....	21
Собственно набор полей	21
Порядковый номер.....	22
Соответствие типов и длины полей	22
Глава 3. Перенос данных.....	33
УСТАНОВКА MYSQL CONNECTOR/NET 6.0	33
ПЕРЕНОС С ПОМОЩЬЮ VISUAL STUDIO.....	35
Комментарии к коду	41
ПЕРЕНОС ДАННЫХ С ПОМОЩЬЮ POWERSHELL	43
Глава 4. Общая последовательность действий.....	46
Заключение	48



Введение

Документ составлен отделом поддержки компании «1С-Битрикс» (Денис Шаромов) в сотрудничестве с компанией Microsoft (Алексей Шуленин) и носит рекомендательный характер. «Пример» разработан для описания технологии миграции базы сайта с MySQL на SQL Express без использования продуктов сторонних разработчиков. Описанная ниже технология справедлива как для «1С-Битрикс: Управление сайтом», так и для «1С-Битрикс: Корпоративный портал».

Для своей работы «1С-Битрикс: Управление сайтом» использует базу данных, в которой хранит контент и настройки сайта. В качестве СУБД используются свободно распространяемые MySQL, SQL Express или Oracle. В последнее время от клиентов, выбравших установку с MySQL, стали поступать вопросы о переходе на SQL Express. Документ предназначен для разработчиков и администраторов сайта, перед которыми стоит задача провести миграцию действующего сайта с базы MySQL на SQL Express. Специалисты, осуществляющие миграцию должны уметь работать с указанными базами данных.

Для обеспечения доступа к БД MySQL из внешних приложений были выбраны MySQL Connector/ODBC 5.1 и MySQL Connector/Net 6.0 (<http://dev.mysql.com/downloads/connector/>). Первый коннектор используется при сравнении структур баз MySQL- и SQL Express-инсталляций, второй – непосредственно при переносе данных.

При миграции не требуется создавать структуры таблиц, индексы, представления, ограничения, триггеры на стороне SQL Express по образцу того, как это было в MySQL.

«1С-Битрикс: Управление сайтом» имеет отдельные инсталляции для MySQL и SQL Express, т.е. структура базы и там, и там создается самостоятельно в процессе установки продукта и изменению не подлежит. В «Примере» приводится сравнение структур таблиц в случае MySQL и SQL Express, соотнесение таблиц и колонок, процесс копирования данных из БД MySQL в SQL Express на основе выстроенной схемы отображения.

Глава 1. Предварительные операции

Установка 1С-Битрикс в варианте MySQL

Согласно постановке задачи первоначально клиент имеет «1С-Битрикс: Управление сайтом» в варианте MySQL. При создании «Примера» использовалась редакция - «Большой бизнес» (<http://www.1c-bitrix.ru/download/cms.php>) как имеющая наиболее сложную и полную структуру базы, установленная согласно «Руководству по установке «1С-Битрикс: Управление сайтом»» (http://www.1c-bitrix.ru/download/manuals/ru/installation_guide.doc).

Установка SQL Express

Для переноса данных необходимо иметь установку «1С-Битрикс: Управление сайтом» с использованием **SQL Express**. В отличие от **MySQL SQL Express** не входит в установку окружения «1С-Битрикс: Управление сайтом». Его необходимо скачать и установить самостоятельно. Рекомендуется использовать последнюю версию продукта.

Скачивание производится по адресу <http://www.microsoft.com/rus/express/sql/download>. Дистрибутив **SQL Express** распространяется в виде пакета **Windows Installer**.

SQL Express доступен в трех комплектациях:

1. **SQL Server 2008 Express (Runtime Only)**
2. **SQL Server 2008 Express with Tools** (Это п.1 + Management Studio (графическая утилита для разработки и администрирования)).
3. **SQL Server 2008 Express with Advanced Services**. (Это п.2 + полнотекстовый поиск и службы отчетности.)

Рекомендуется использовать п.2 или 3. Пунктом 1 обычно пользуются независимые производители программного обеспечения, чтобы распространять его в составе своего приложения. Если планируется в дальнейшем прибегнуть к возможности полнотекстового поиска или разработке отчетов, выбирайте п.3. Если достаточно графической утилиты для разработки и администрирования, то используйте п.2.

В документе установка **SQL Express** иллюстрируется в варианте п.3.

Ø Загрузите дистрибутив.

Ø Запустите установку.

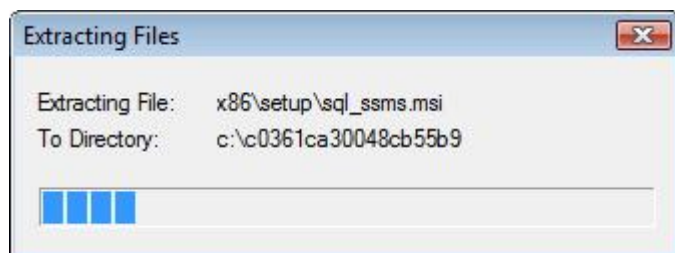


Рисунок 1.1 Процесс распаковки

В начале установки происходит проверка характеристик аппаратной платформы и установленного ПО на предмет совместимости с **SQL Express**.

Если обнаружено несовместимое ПО, либо какого-то ПО не хватает в качестве предварительно установленного, то программа установки выдаст об этом предупреждение (Рисунок 1.2).

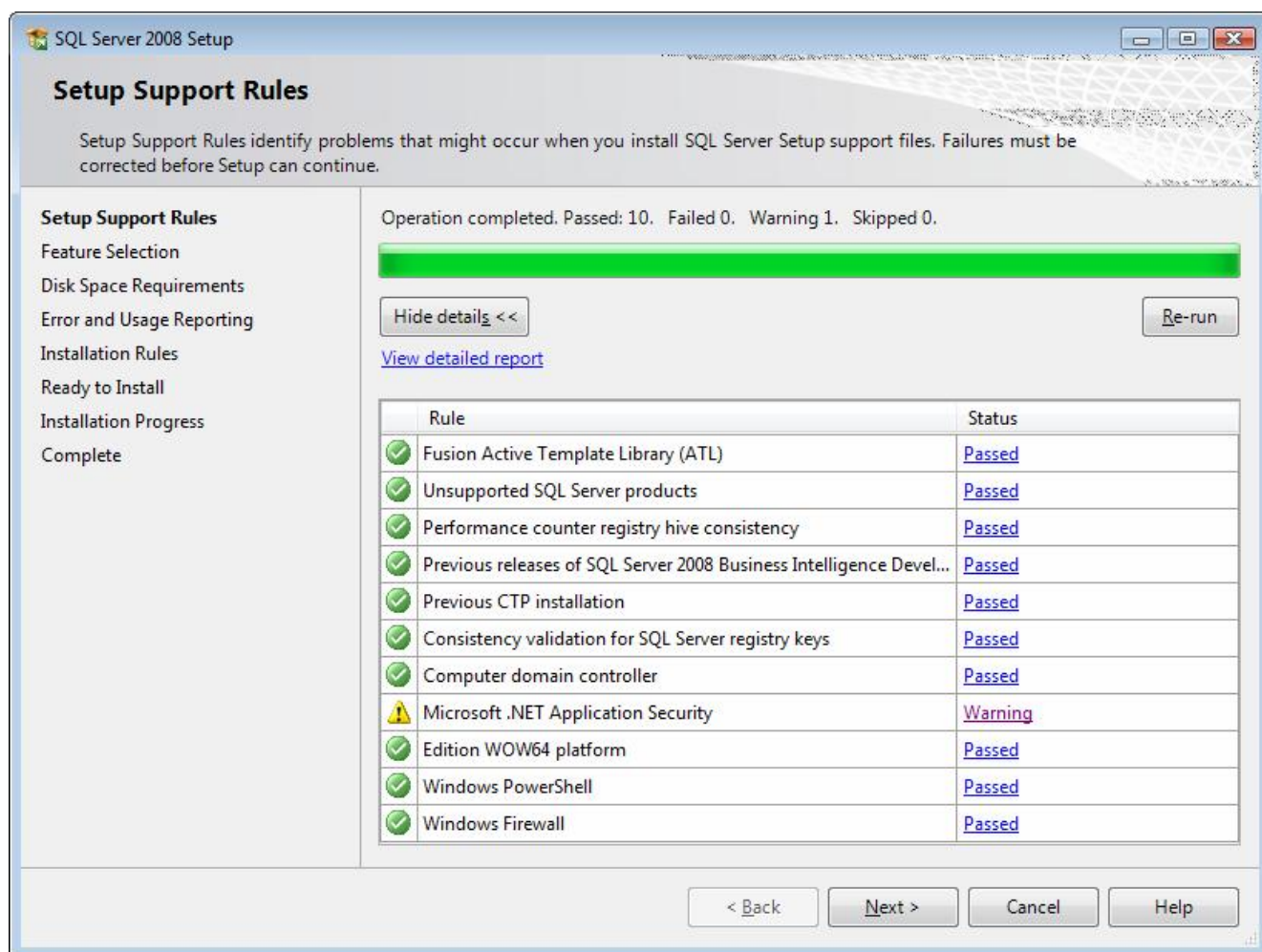


Рисунок 1.2 Сообщение об ошибке

После устранения неисправностей необходимо запустить процесс установки заново. По окончании проверки, если характеристики соблюдены, установщик переходит к собственно установке **SQL Express**.

Вначале устанавливаются вспомогательные файлы для процесса установки, затем в обычном формате **Windows Installer** будет предложено выбрать устанавливаемые опции (Рисунок 1.3):

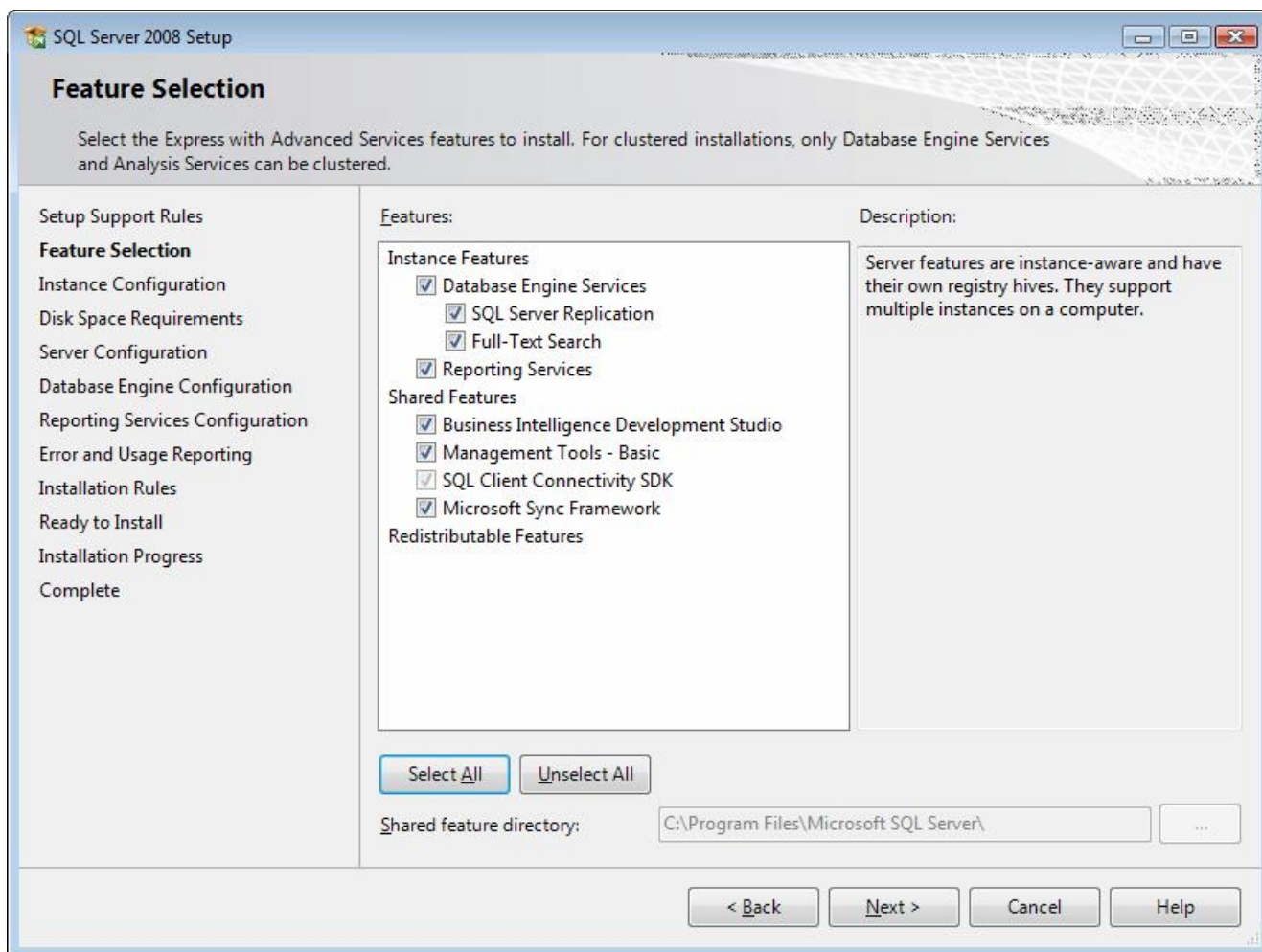


Рисунок 1.3 Выбор устанавливаемых опций

Исполнительный механизм **SQL Express**, обслуживающий базы данных и выполняющий запросы, оформлен как сервис операционной системы. На одну машину можно поставить до 16-ти экземпляров **SQL Express**. Это будет 16 независимых сервисов, каждый из которых ведет своими базами данных. Каждый сервис должен иметь свое уникальное имя.

Если на машине не установлены **SQL Server**'а, то экземпляр можно ставить по умолчанию – поле **Default Instance** на Рисунок 1.4. Тогда его имя будет совпадать с именем машины. Последующие экземпляры надо будет именовать, это осуществляется в поле **Named instance**. К именованному экземпляру обращаются как <имя машины>\<имя экземпляра>.

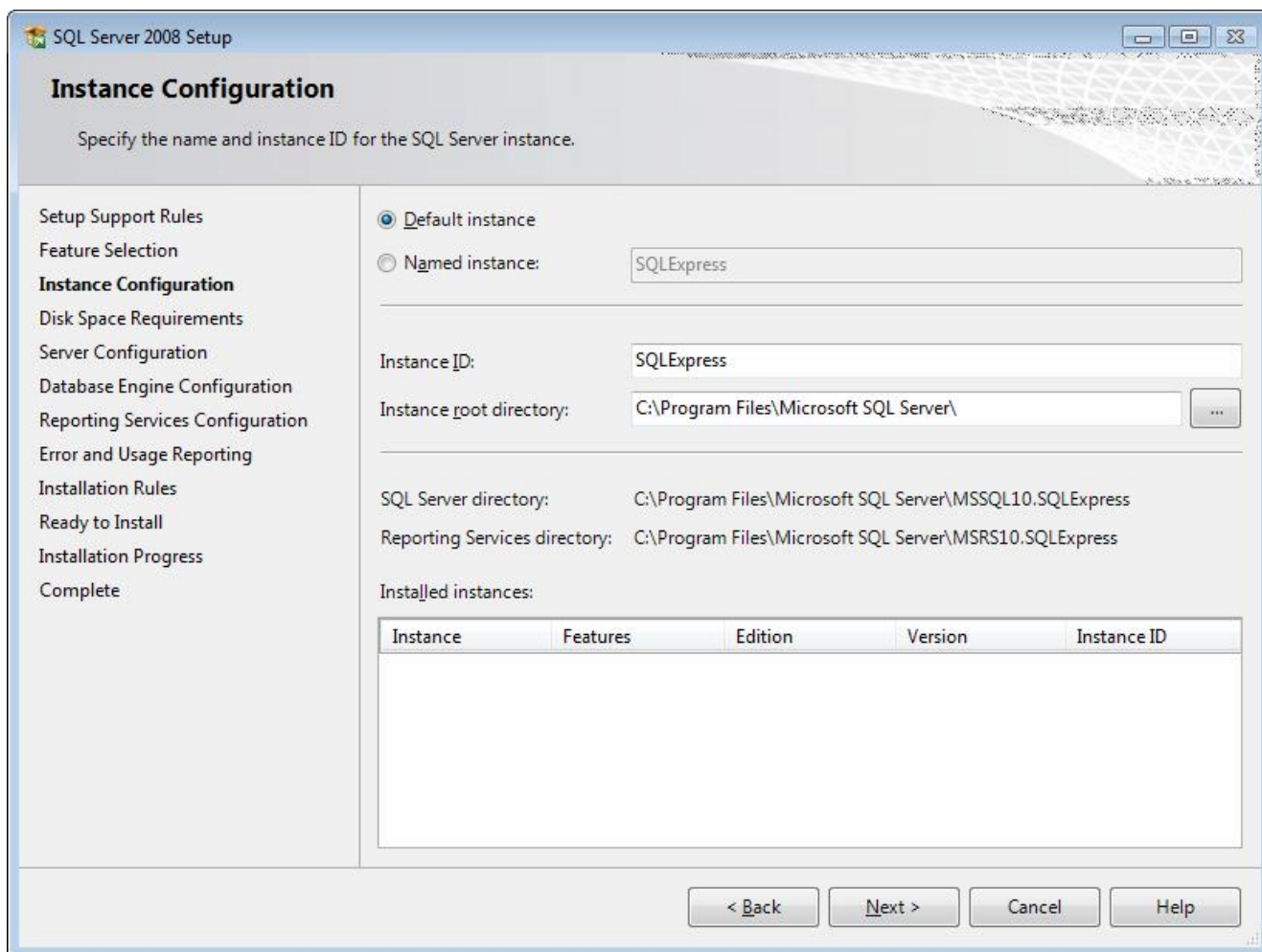


Рисунок 1.4 Настройка конфигурации

⚠ Примечание. Размер одной БД в SQL Express ограничен 4 гигабайтами. SQL Express понимает только 1 процессор (под процессором понимается исполнительная сущность, например, ядро) и умеет работать не более, чем с 1 гигабайтом памяти. О других его ограничениях можно прочитать здесь - http://msdn.microsoft.com/en-us/library/ms345154.aspx#sseover_topic4.

На приведенном ниже экране (Рисунок 1.5) производится выбор юникодовой колляции, применяемой по умолчанию.

⚠ Collation (сопоставление) определяет ряд правил, согласно которым сортируются и сравниваются данные. Символьные данные сортируются, используя правила, которые определяют правильную последовательность символов, в зависимости от регистра, надстрочных знаков, символьных типов капа и ширины символов.

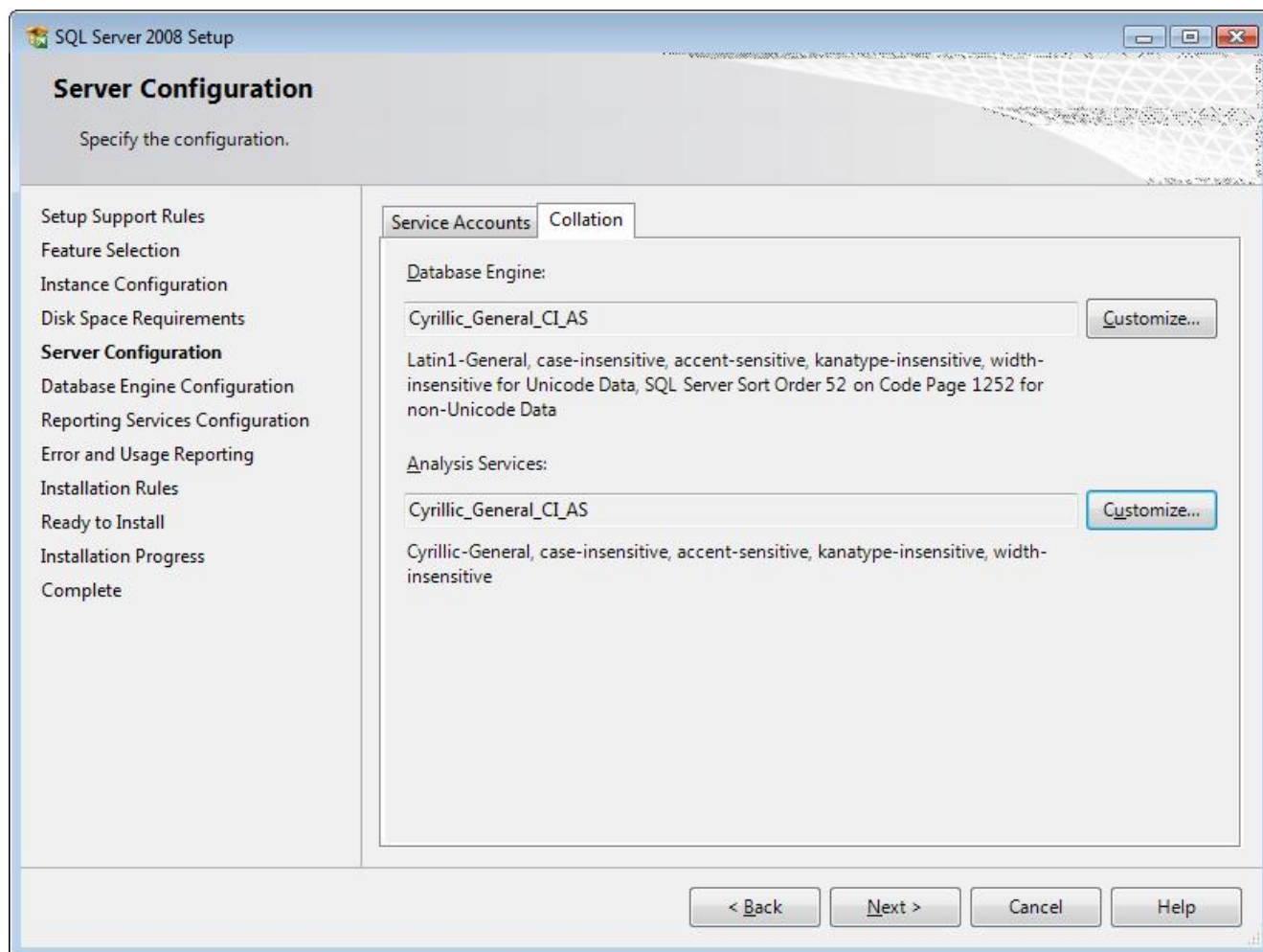


Рисунок 1.5 Настройка сервера

На приведенном ниже экране (Рисунок 1.6) выполняется настройка атрибута **filestream**. Этот атрибут применяется к большим бинарным объектам и означает, что физически они будут храниться не в БД, а в файловой системе. Колонки таблиц **filestream** не попадают под действие упомянутого выше ограничения на размер БД в 4 гигабайта, их размер ограничен только размерами тома файловой системы.

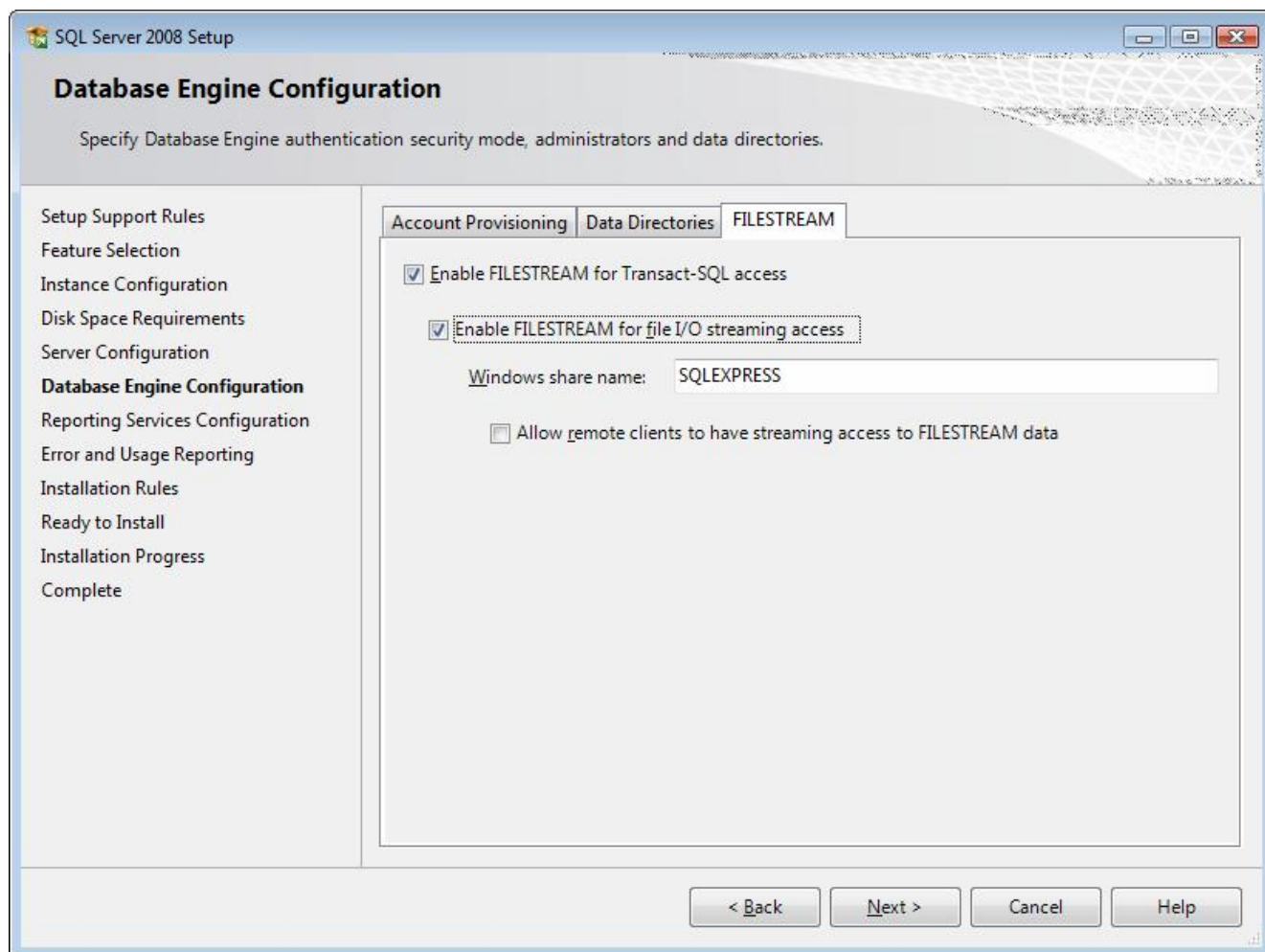


Рисунок 1.6 Настройка базы данных

Экран (Рисунок 1.7) появляется в том случае, если происходит установка **SQL Express** в комплектации №3. На нем предлагается задать, будут ли службы отчетности установлены в режиме интеграции с **Sharepoint**. Оставьте на этом экране все без изменений.

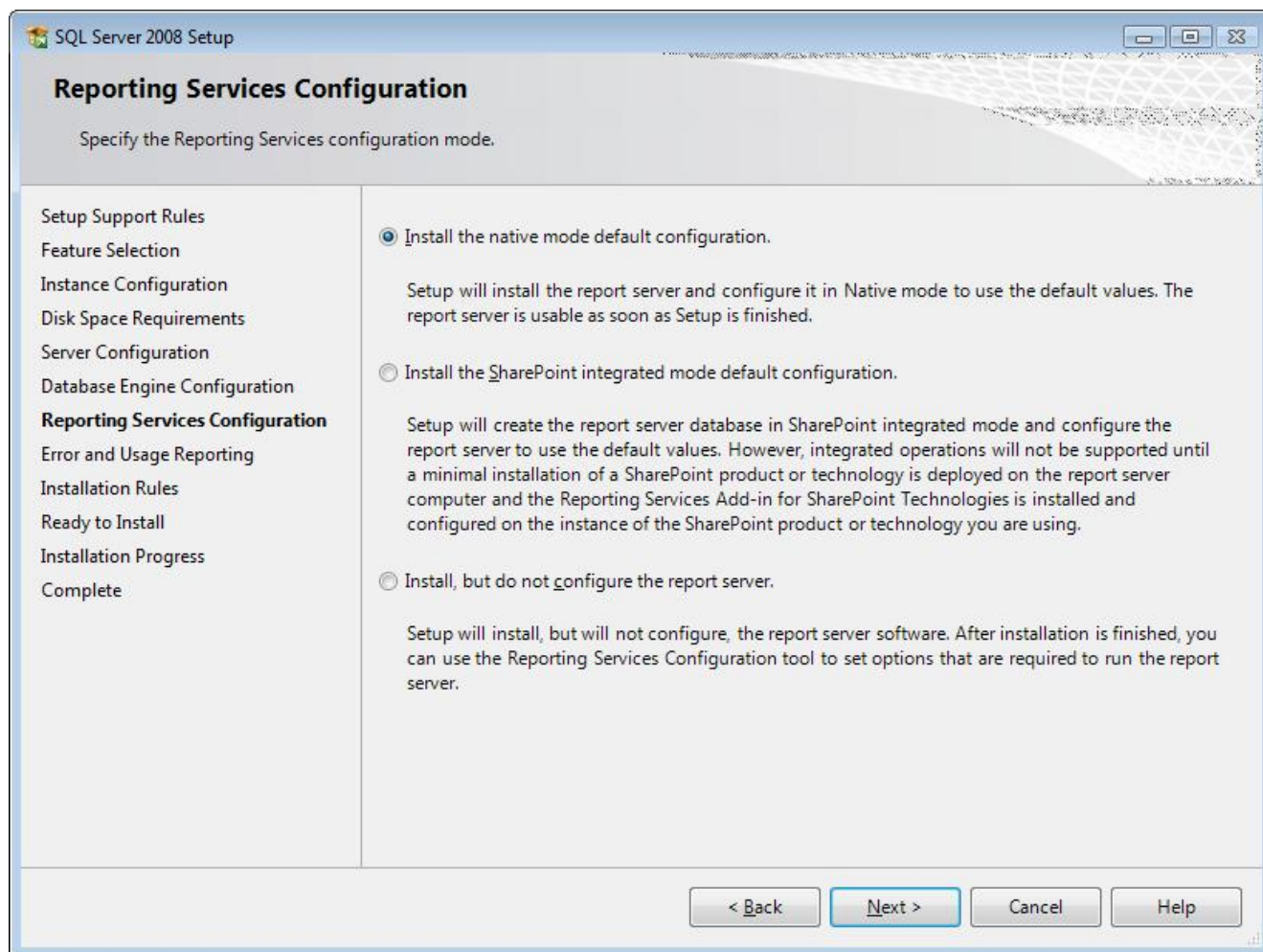


Рисунок 1.7 Установка служб отчетности

После этого мастер установки начнет собственно процесс установки SQL Express.

По его завершении обратитесь к службе **Windows Update** (Пуск > Панель управления > Windows Update) чтобы проверить, какие последние обновления доступны для **SQL Server 2008 Express**. Если такие обновления имеются, то рекомендуется их установить.

После установки обновлений (Рисунок 1.8) **Microsoft SQL Server 2008 Express** готов к работе.

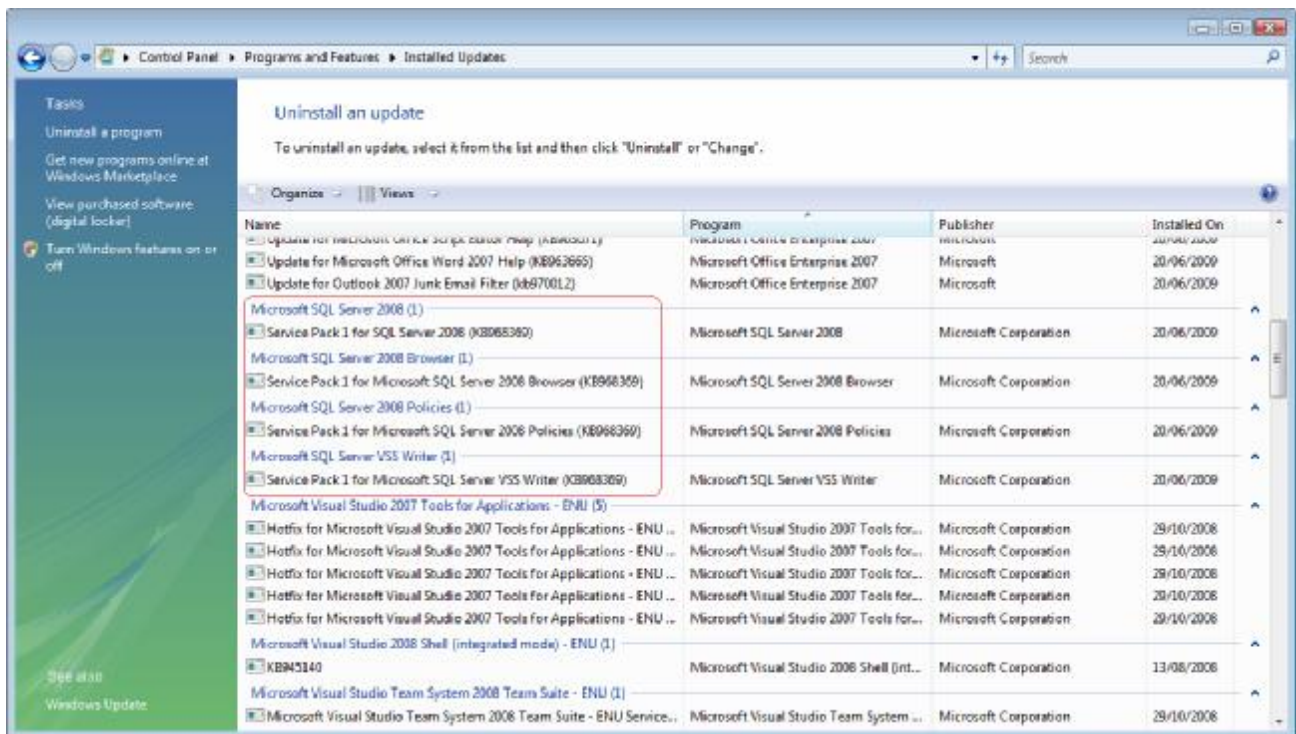


Рисунок 1.8 Список произведенных обновлений

Установка 1С-Битрикс в варианте SQL Express

После установки базы **SQL Express** необходимо установить «1С-Битрикс: Управление сайтом» на базу **SQL Express**. Инсталляцию на **MySQL** при этом убирать не нужно, т.к. база данных **MySQL** содержит текущий контент сайта, который требуется перенести в **SQL Express**.

В связи с постановкой задачи установка «1С-Битрикс: Управление сайтом» имеет свои особенности.

Начало установки

В любом файловом менеджере перейдите в папку, где был установлен *Bitrix Environment* (в рамках локальной установки по умолчанию это будет - *C:\Program Files\Bitrix Environment*).

- Ø Переименуйте подпапку **www** в любую другую.
- Ø Запустите заново процесс установки «1С-Битрикс: Управление сайтом», согласно «Руководству по установке».
- Ø На этапе установки *Bitrix Environment* откажитесь от его установки.
- Ø На этапе выбора места установки самого «1С-Битрикс: Управление сайтом» укажите путь *C:\Program Files\Bitrix Environment\www*.
- Ø Продолжите установку программы до шага выбора базы данных.
- Ø На этом шаге выберите базу **Microsoft SQL Server**.



Создание базы данных

Теперь нам потребуется создать для «1С-Битрикс: Управление сайтом» базу данных на **SQL Express**.

Ø Выполните команду *Пуск > Все программы > SQL Server 2008 > SQL Server Management Studio*.

Ø Соединитесь с экземпляром **SQL Express** (Рисунок 1.9):



Рисунок 1.9 Соединение с базой SQL Express

Ø Кликните правой кнопкой по папке **Databases** в **Object Explorer** и выберите пункт **New Database** (Рисунок 1.10).

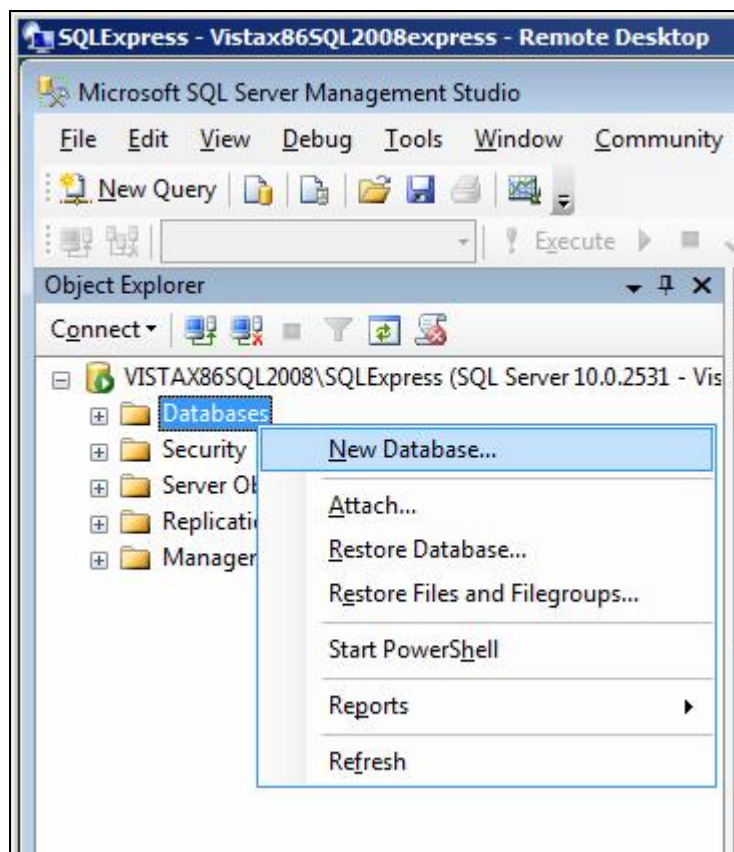


Рисунок 1.10 Создание новой базы

Ø В диалоге создания базы (Рисунок 1.11) введите ее имя, оставив все остальное так, как оно предлагается по умолчанию. (В нашем «Примере» это имя – **Bitrix**). Нажмите внизу **ОК**.

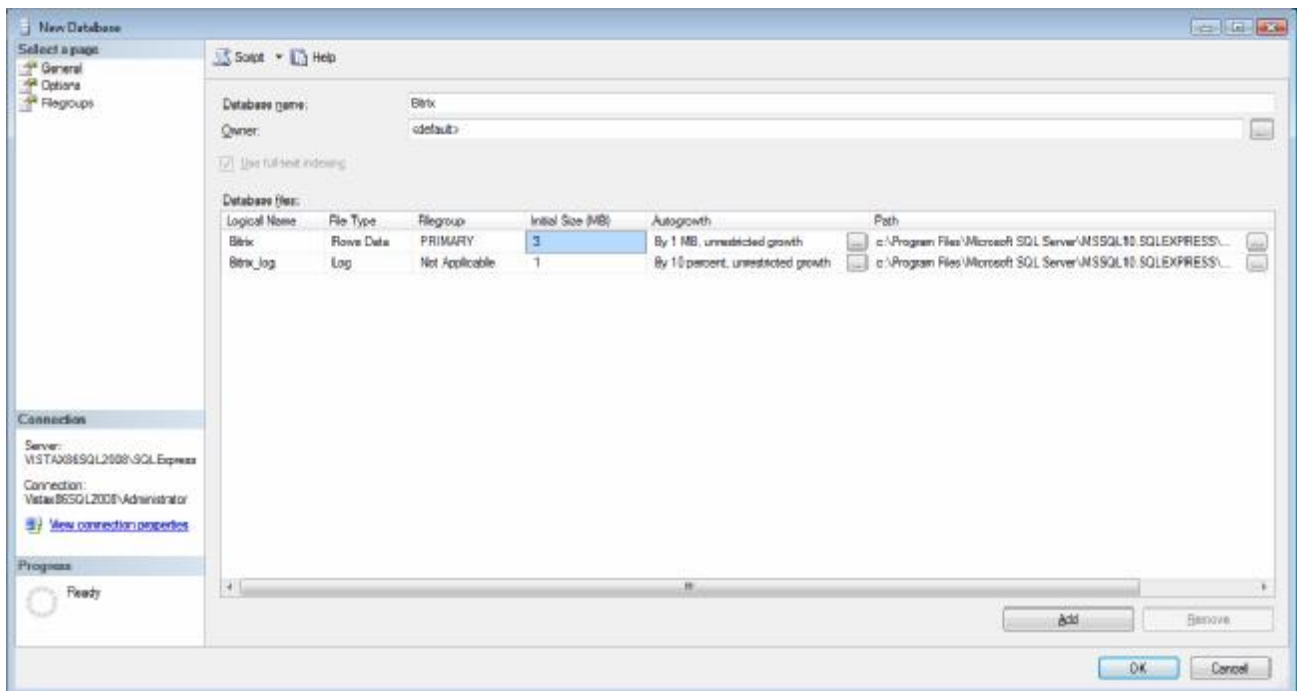


Рисунок 1.11 Задание имени базы

Ø Выполните команду *Control Panel > Administrative Tools > Data Sources (ODBC)*.

Ø Перейдите на закладку **System DSN** и нажмите справа кнопку **Add**. Появится список драйверов. Из списка драйверов выбираем **SQL Server Native Client 10.0** как наиболее свежий на момент написании «Примера» (Рисунок 1.12):

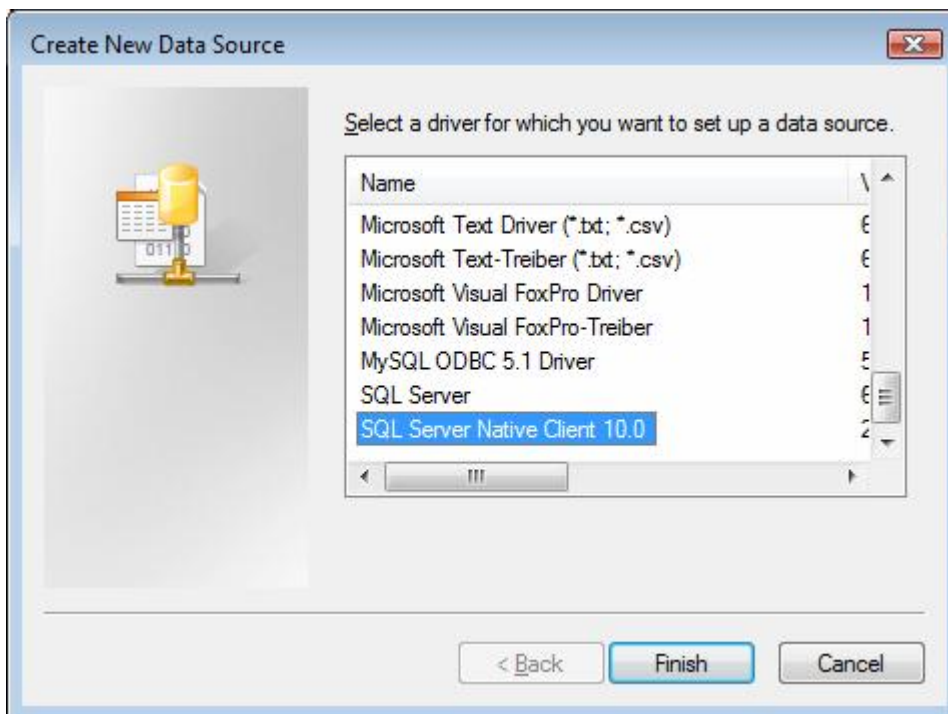


Рисунок 1.12 Выбор драйвера для сервера

Ø Введите название источника и имя сервера, к которому устанавливать соединение (Рисунок 1.13).

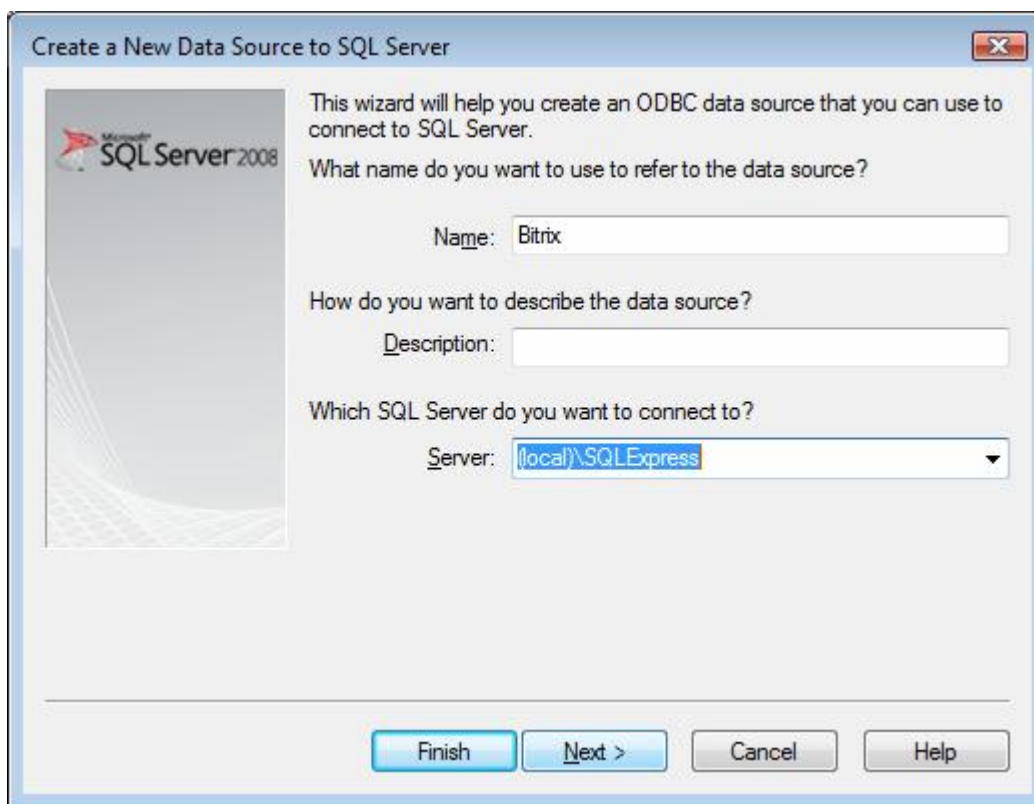


Рисунок 1.13 Выбор источника и имени сервера

 **Примечание.** Напомним, что имя сервера состоит из имени_машины\имени_экземпляра, которое задается SQL Server при установке. Если SQL Server ставился как экземпляр по умолчанию, достаточно просто имени машины. Если SQL Browser Service выключен, а порт используется нестандартный, то нужно в имени указать и порт.

 **Примечание.** Создать соединение можно и по примеру <http://support.microsoft.com/kb/194641>.

На остальных шагах создания базы данных сохраняются настройки по умолчанию.

Задание логина и пароля

Необходимо задать логин и пароль администратора **SQL Server**.

Ø Выполните команду Пуск > Все программы > SQL Server 2008 > SQL Server Management Studio.

Ø Object Explorer раскройте папку Security\Logins, выберите логин sa (Рисунок 1.14).

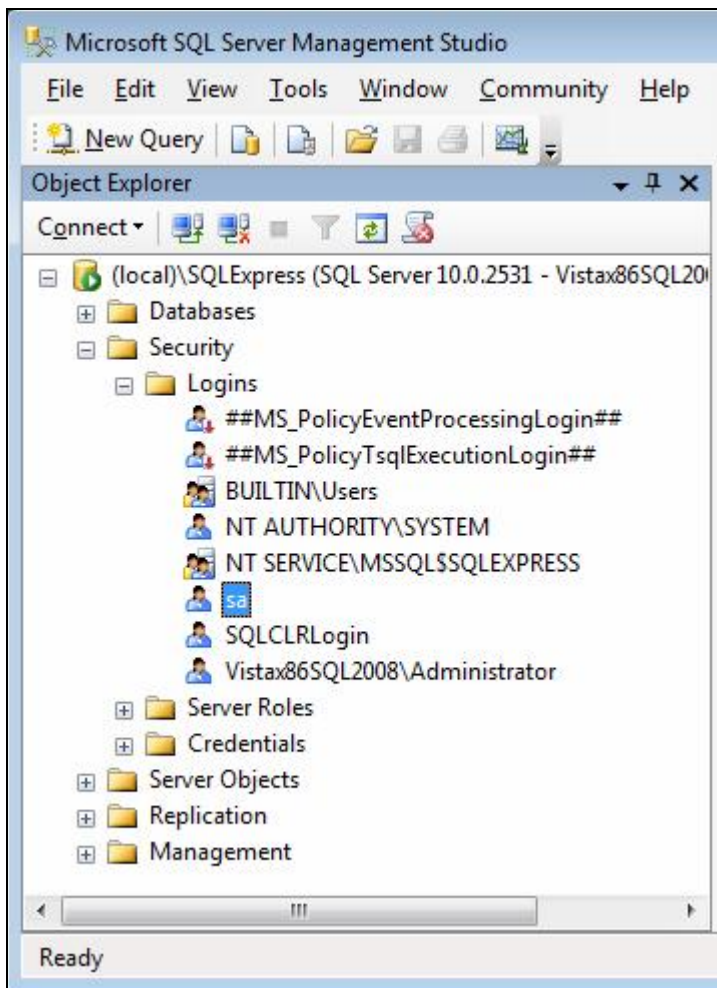


Рисунок 1.14 Задание логина и пароля

Ø Дважды кликните на нем. В открывшемся окне введите пароль и подтвердите его (Рисунок 1.15).

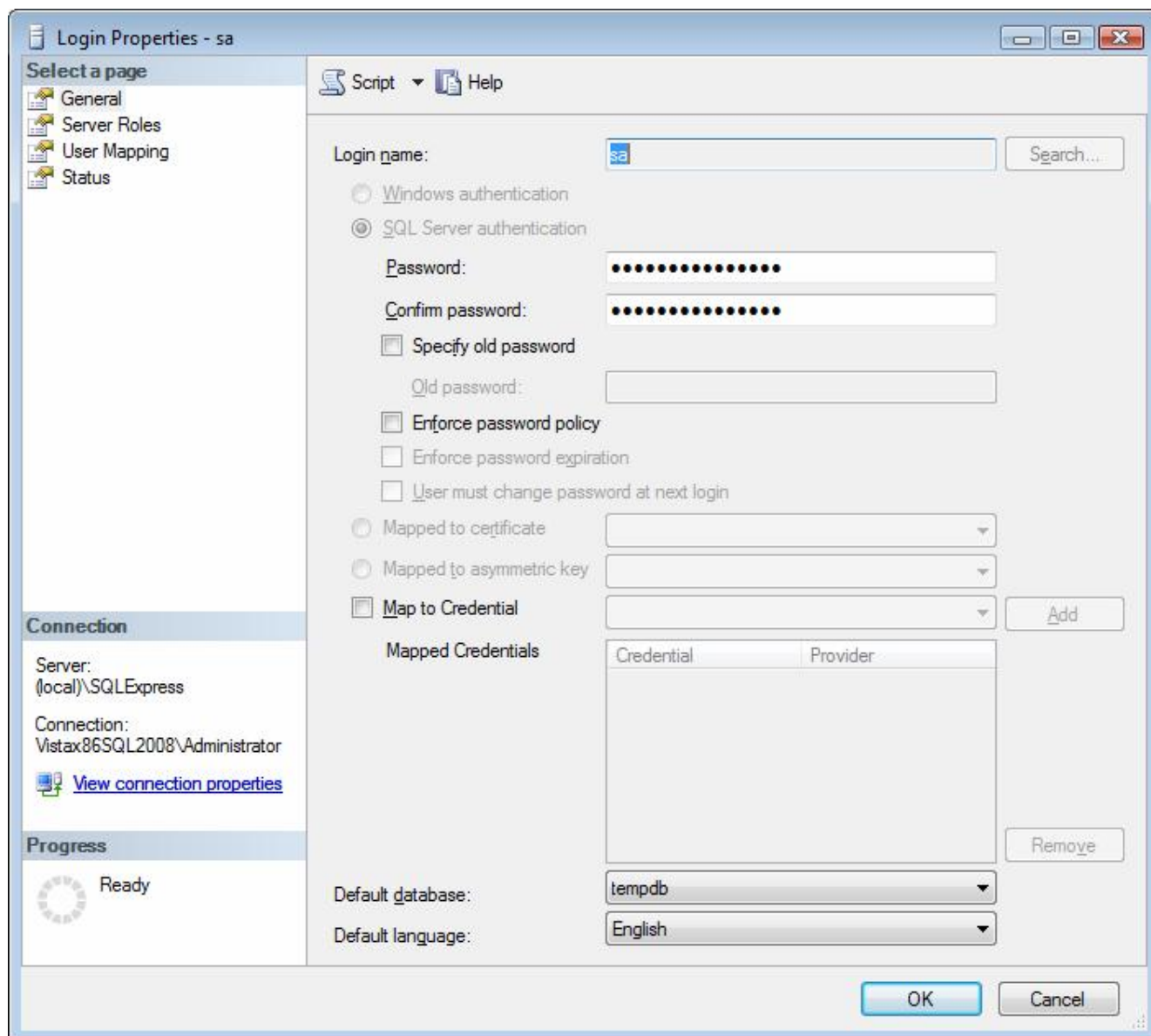


Рисунок 1.15 Задание пароля для логина sa

Ø Нажмите **ОК**. Диалог закроется.

Если логин **sa** не отображается в общем списке, то:

Ø В верхней строчке **Object Explorer** вызовите через контекстное меню свойства созданного сервера.

Ø Перейдите на раздел **Security** и отметьте справа **SQL Server** и **Windows Authentication mode** (Рисунок 1.16).

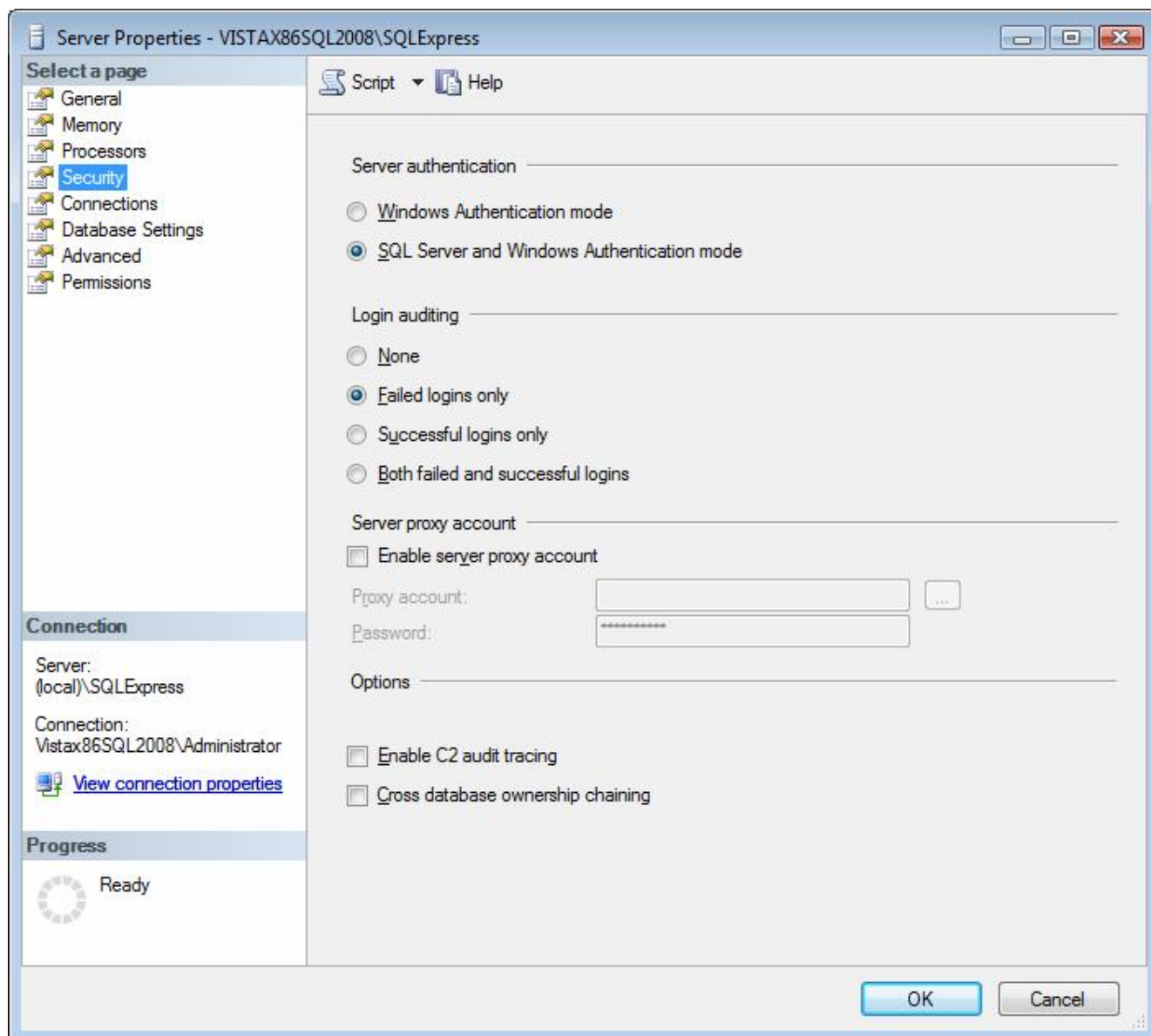


Рисунок 1.16 Настройка параметров сервера

Логин **sa** появится.

Завершение установки

Продолжите установку «1С-Битрикс: Управление сайтом» согласно «Руководству по установке». В поле **DSN** используйте название созданного вами источника **ODBC**. В качестве **Базы данных** используем созданную только что базу.

Создайте нового пользователя для указанной базы, используя логин и пароль администратора и завершите установку продукта.



Глава 2. Сравнение структур баз

На данный момент имеется старый контент сайта в БД **bsm_demo** на **MySQL** и новый, но пустой сайт в БД **bitrix** на **SQL Express**. Перед переносом данных необходимо выяснить тождественность структуры баз в случае **MySQL** и **SQL Express**.

Создание сервера

В примере используется прилинкованный сервер со стороны **SQL Express** на **MySQL**, который использует **MSDASQL (OLE DB поверх ODBC)**.

⚠ Примечание. Сервер Microsoft позволяет посылать запросы не только к «родной» базе **SQL Express**, но и к базе **MySQL**. Для этого необходимо использовать специальный **ODBC-драйвер**.

Создание сервера:

- Ø Загрузите **ODBC-драйвер** для **MySQL** **5.1** (<http://dev.mysql.com/downloads/connector/odbc/5.1.html>).
- Ø Установите драйвер в систему.
- Ø Выполните команду *Пуск > Все программы > SQL Server 2008 > SQL Server Management Studio*.
- Ø Создайте прилинкованный сервер без создания DSN:

```
if exists (select 1 from sys.servers where name = 'MySQL')
exec sp_dropserver @server = 'MySQL', @droplogins = 'droplogins'
go
exec sp_addlinkedserver @server = 'MySQL', @srvproduct = 'MySQLDatabase', @provider = 'MSDASQL',
@provstr = 'DRIVER={MySQL ODBC 5.1 Driver}; SERVER=localhost; PORT=31006; DATABASE=bsm_demo; UID=root; PWD=; OPTION=3'
go
exec sp_addlinkedsrvlogin @rmtsrvname = 'MySQL', @useself = 'false', @locallogin = NULL,
@rmtuser = 'root', @rmtpassword = ''
go
exec sp_serveroption @server = 'MySQL', @optname = 'rpc', @optvalue = 'true'
exec sp_serveroption @server = 'MySQL', @optname = 'rpc out', @optvalue = 'true'
```

Теперь сервер позволяет выполнять на **MySQL** запросы, адресованные к **SQL Server**, и возвращать результат **MySQL**, как если бы это был результат **SQL Server** (Рисунок 2.1):

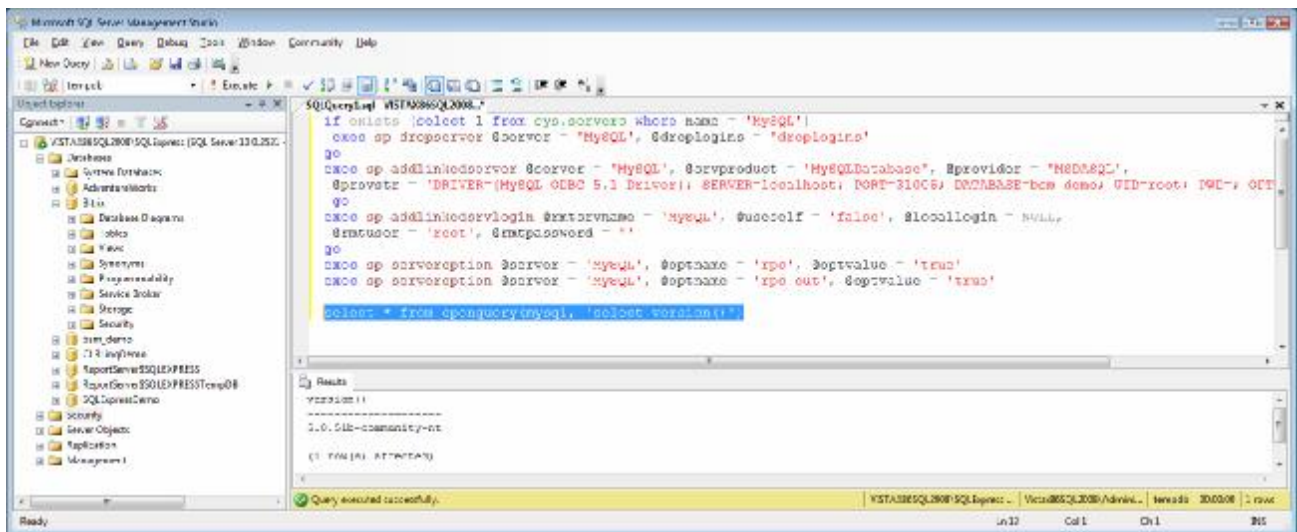


Рисунок 2.1 Выполнение скрипта

Состав таблиц

В первую очередь необходимо выяснить, насколько отличаются базы по составу таблиц. Данный запрос выводит несовпадения:

```
use bitrix
```

```
with
```

```
tbl_rows_sqlsrv as (
```

```
select t.name, p.n from sys.tables t
```

```
join (select object_id, sum(row_count) n from sys.dm_db_partition_stats where index_id in (0, 1)
group by object_id) p
```

```
on t.object_id = p.object_id
```

```
)
```

```
, tbl_rows_mysql as (
```

```
select * from openquery(mysql, 'select table_name name, table_rows n from
information_schema.tables where table_schema = "bsm_demo"')
```

```
)
```

```
select sqlsrv.name, sqlsrv.n, mysql.name, mysql.n from tbl_rows_sqlsrv sqlsrv full outer join
tbl_rows_mysql mysql on sqlsrv.name = mysql.name
```

```
where sqlsrv.name is null or mysql.name is null or sqlsrv.n <> mysql.n
```

name	n	name	n
B_OPTION	112	b_option	113
B_STAT_SESSION_DATA	0	b_stat_session_data	1
B_FILE_ACTION	0	NULL	NULL



<i>B_POSTING_LOCK</i>	0	NULL	NULL
<i>B_FAVORITE_LANG</i>	0	NULL	NULL

Вывод. Все 319 таблиц в базе при инсталляции **MySQL** имеют соответствия (т.е. таблицы с тем же именем) в инсталляции на **SQL Server**.

В инсталляции на **SQL Server** имеются 3 таблицы, не имеющих соответствия в инсталляции на **MySQL**:

- *B_FILE_ACTION*
- *B_POSTING_LOCK*
- *B_FAVORITE_LANG*

Это благоприятная ситуация для миграции, поскольку перенос данных планируется именно из **MySQL** в **SQL Server**. Хуже, если бы, наоборот, в **MySQL** имелись таблицы, которые бы было непонятно куда переносить в **SQL Server**.

Из 319 таблиц инсталляции **MySQL** две не совпадают по числу строк с соответствующими им таблицами в инсталляции на **SQL Server**:

<i>SQL Server</i>		<i>MySQL</i>	
<i>B_OPTION</i>	112	<i>b_option</i>	113
<i>B_STAT_SESSION_DATA</i>	0	<i>b_stat_session_data</i>	1

Этот факт имеет, роль комментария, поскольку в случае «Примера» инсталляции «1С-Битрикс: Управление сайтом» как в случае **MySQL**, так и **SQL Server** - "чистые", то есть без данных. В реальной жизни клиент уже будет работать продолжительный период с **MySQL**'ной базой, поэтому данных там будет больше, чем в только что установленной базе **SQL Server**.

Речь не идет о каком-то слиянии просто данные из **MySQL** требуется перенести в соответствующие таблицы **SQL Server**, перетерев все, что туда уже успел добавить процесс инсталляции CMS. Инсталляция «1С-Битрикс: Управление сайтом» в варианте **SQL Server** создает, по сути, готовые структуры для приема данных со стороны **SQL Server** и избавляет нас от необходимости рассматривать миграцию метаданных.

Набор полей

Необходимо исследовать наборов полей в соответствующих таблицах: какие поля есть в таблице **MySQL**, которых нет в таблице **SQL Server** и наоборот.

Собственно набор полей

```
use bitrix
```

```
with
```

```
col_sqlsrv as (
```

```
select t.name as tbl_name, c.name as col_name from sys.columns c join sys.tables t on c.object_id = t.object_id
```



```
)  
, col_mysql as (  
select * from openquery(mysql, 'select table_name tbl_name, column_name col_name from  
information_schema.columns where table_schema = "bsm_demo")  
)  
select * from col_sqlsrv sqlsrv full outer join col_mysql mysql  
on sqlsrv.tbl_name = mysql.tbl_name and sqlsrv.col_name = mysql.col_name  
where sqlsrv.col_name is null or mysql.col_name is null
```

Вывод. Сравнение выявило абсолютное тождество наборов полей в соответствующих таблицах. Исключением являются три таблицы в **SQL Server**, которых нет в варианте установки **MySQL**.

Порядковый номер

Также существенным фактором является совпадение порядкового номера одноименной колонки, потому что если это так, процесс переноса данных будет проще. В противном случае придется явно перечислять поля в списке в нужном порядке или строить соответствие полей источника с полями назначения.

```
with  
col_sqlsrv(tbl_name, col_name, col_pos) as (  
select object_name(object_id), name, column_id from sys.columns  
)  
, col_mysql(tbl_name, col_name, col_pos) as (  
select * from openquery(mysql, 'select table_name, column_name, ordinal_position from  
information_schema.columns where table_schema = "bsm_demo")  
)  
select * from col_sqlsrv sqlsrv join col_mysql mysql  
on sqlsrv.tbl_name = mysql.tbl_name and sqlsrv.col_name = mysql.col_name  
where sqlsrv.col_pos <> mysql.col_pos
```

Вывод. Порядковый номер колонки не совпадает в 80 случаях, что необходимо учитывать при переносе данных.

Соответствие типов и длины полей

Необходимо выяснить соответствие типов и длины в одноименных колонках. Этим скриптом можно получить типы колонок, использующиеся «1С-Битрикс: Управление сайтом» в случае установки на **MySQL** и на **SQL Server**:

```
select * from openquery(mysql, 'select distinct data_type from information_schema.columns where  
table_schema = "bsm_demo")  
select distinct type_name(c.user_type_id) from sys.columns c join sys.tables t on c.object_id =  
t.object_id where type = 'U'
```



Вывод. В **SQL Server** типов колонок - 11, в **MySQL** – 16. Например, в случае инсталляции **MySQL** используется тип **smallint**, а в варианте инсталляции **SQL Server** он не используется. В **MySQL** есть разновидности блобовских типов, которых действительно нет в **SQL Server**, например, **mediumtext**, **longtext**. В случае **SQL Server** им соответствует один тип **text**.

«1С-Битрикс: Управление сайтом» в **SQL Server** использует **text/image**, **datetime**. в **MySQL** - тип **date**. Необходимо вручную построить таблицу соответствия типов, т.е. какой тип **MySQL** в какой тип **SQL Server** можно без потерь переносить.

⚠ Примечание. Если существует экземпляр типа А, который не перенесется в тип Б без обрезания или дополнительных преобразований, такой перенос считается невозможным. Все типы и в **MySQL**, и в **SQL Server** можно разбить на числовые, строковые, бинарные и календарные. Перенос внутри каждой категории считается допустимым, при этом длина поля приемника должна быть не меньше, чем у источника, а, в случае численных полей с фиксированной точкой то же распространяется и на количество знаков после запятой.

Пример запроса, который проверяет нарушения этого правила:

```
with
col_sqlsrv(tbl_name, col_name, col_type, col_len, col_prec, col_scal) as (
select object_name(object_id), name, type_name(user_type_id), max_length, precision, scale from
sys.columns
),
col_sqlsrv1(tbl_name, col_name, col_type, col_len, col_prec, col_scal) as (
select tbl_name, col_name,
    case when col_type in ('bigint', 'int', 'tinyint', 'decimal', 'numeric') then 'N'
        when col_type in ('float') then 'F'
        when col_type in ('datetime') then 'D'
        when col_type in ('char', 'varchar', 'text') then 'C'
        when col_type in ('image') then 'B'
    end,
    case when col_type = 'text' then power(cast(2 as bigint), 31) - 1
        else col_len
    end,
    col_prec, col_scal from col_sqlsrv
)
, col_mysql(tbl_name, col_name, col_type, col_len, col_prec, col_scal) as (
select * from openquery(mysql, 'select table_name, column_name, data_type,
character_maximum_length, numeric_precision, numeric_scale from information_schema.columns
where table_schema = "bsm_demo"')
)
```



```

, col_mysql1(tbl_name, col_name, col_type, col_len, col_prec, col_scal) as (
select tbl_name, col_name,
    case when col_type in ('bigint', 'int', 'tinyint', 'decimal', 'smallint') then 'N'
        when col_type in ('float') then 'F'
        when col_type in ('datetime', 'timestamp') then 'D'
        when col_type in ('char', 'varchar', 'text', 'mediumtext', 'longtext') then 'C'
        when col_type in ('longblob') then 'B' end,
    col_len, col_prec, col_scal from col_mysql
)
select sqlsrv.*, mysql.col_type, mysql.col_len, mysql.col_prec, mysql.col_scal from col_sqlsrv1 sqlsrv
join col_mysql1 mysql
on sqlsrv.tbl_name = mysql.tbl_name and sqlsrv.col_name = mysql.col_name
where mysql.col_type <> sqlsrv.col_type
    or mysql.col_type = sqlsrv.col_type and mysql.col_len > sqlsrv.col_len
    or mysql.col_type = sqlsrv.col_type and (mysql.col_prec > sqlsrv.col_prec or mysql.col_scal >
sqlsrv.col_scal)
order by 1, 2

```

Первое условие в **where** не нарушается, что означает, что с преобразованием типов проблем нет – числовые типы переносятся в числовые, текстовые в текстовые и т.д. Однако выявлено 128 колонок, нарушающих второе и третье условие в **where**. Это означает, что при переносе данных возможно возникновение ошибки из-за недостаточного размера поля приемника.

		SQL Express				MySQL			
Таблица	Колонка	Категория типа	Размер поля	Численность длина	После запятый	Категория типа	Размер поля	Численность длина	После запятый
B_ADV_BANNER	COMMENTS	C	1000	0	0	C	65535	NULL	NULL
B_ADV_BANNER	KEYWORDS	C	1000	0	0	C	65535	NULL	NULL
B_ADV_BANNER	STATUS_COMMENTS	C	500	0	0	C	65535	NULL	NULL
B_ADV_BANNER	URL	C	8000	0	0	C	65535	NULL	NULL



B_ADV_CONTRACT	ADMIN_COMMENTS	C	500	0	0	C	65535	NULL	NULL
B_ADV_CONTRACT	DESCRIPTION	C	2000	0	0	C	65535	NULL	NULL
B_ADV_CONTRACT	KEYWORDS	C	1000	0	0	C	65535	NULL	NULL
B_ADV_TYPE	DESCRIPTION	C	500	0	0	C	65535	NULL	NULL
B_CATALOG_LOADED	VALUE	C	2000	0	0	C	65535	NULL	NULL
B_EVENT	C_FIELDS	C	2147483647	0	0	C	4294967295	NULL	NULL
B_EVENT_LOG	REQUEST_URI	C	2000	0	0	C	65535	NULL	NULL
B_EVENT_LOG	USER_AGENT	C	2000	0	0	C	65535	NULL	NULL
B_EVENT_TYPE	LID	C	2	0	0	C	201	NULL	NULL
B_FAVORITE	COMMENTS	C	8000	0	0	C	65535	NULL	NULL
B_FAVORITE	URL	C	8000	0	0	C	65535	NULL	NULL
B_FORM	DESCRIPTION	C	8000	0	0	C	65535	NULL	NULL
B_FORM	FILTER_RESULT_TEMPLATE	C	8000	0	0	C	65535	NULL	NULL
B_FORM	TABLE_RESULT_TEMPLATE	C	8000	0	0	C	65535	NULL	NULL
B_FORM_2_GROUP	PERMISSION	N	1	3	0	N	NULL	10	0
B_FORM_ANSWER	FIELD_PARAM	C	8000	0	0	C	65535	NULL	NULL
B_FORM_ANSWER	MESSAGE	C	8000	0	0	C	65535	NULL	NULL



B_FORM_FIELD	COMMENTS	C	8000	0	0	C	65535	NULL	NULL
B_FORM_FIELD	FILTER_TITLE	C	8000	0	0	C	65535	NULL	NULL
B_FORM_FIELD	RESULTS_TABLE_TITLE	C	8000	0	0	C	65535	NULL	NULL
B_FORM_FIELD	TITLE	C	8000	0	0	C	65535	NULL	NULL
B_FORM_RESULT_ANSWER	ANSWER_TEXT	C	8000	0	0	C	65535	NULL	NULL
B_FORM_RESULT_ANSWER	ANSWER_TEXT_SEARCH	C	21474 83647	0	0	C	42949 67295	NULL	NULL
B_FORM_RESULT_ANSWER	ANSWER_VALUE_SEARCH	C	21474 83647	0	0	C	42949 67295	NULL	NULL
B_FORM_RESULT_ANSWER	USER_TEXT	C	21474 83647	0	0	C	42949 67295	NULL	NULL
B_FORM_RESULT_ANSWER	USER_TEXT_SEARCH	C	21474 83647	0	0	C	42949 67295	NULL	NULL
B_FORM_STATUSES	DESCRIPTION	C	8000	0	0	C	65535	NULL	NULL
B_FORUM	DESCRIPTION	C	1000	0	0	C	65535	NULL	NULL
B_FORUM_FILTER	USE_IT	C	1	0	0	C	50	NULL	NULL
B_FORUM_MESSAGE	ID	N	4	10	0	N	NULL	19	0
B_FORUM_MESSAGE	TOPIC_ID	N	4	10	0	N	NULL	19	0
B_FORUM_PRIVATE_MESSAGE	ID	N	4	10	0	N	NULL	19	0
B_FORUM_PRIVATE_MESSAGE	IS_READ	C	1	0	0	C	50	NULL	NULL
B_FORUM_PRIVATE_MESSAGE	USE_SMILES	C	1	0	0	C	50	NULL	NULL



B_FORUM_STAT	ID	N	4	10	0	N	NULL	19	0
B_FORUM_SUBSCRIBE	NEW_TOPIC_ONLY	C	1	0	0	C	50	NULL	NULL
B_FORUM_TOPIC	ID	N	4	10	0	N	NULL	19	0
B_FORUM_TOPIC	LAST_MESSAGE_ID	N	4	10	0	N	NULL	19	0
B_FORUM_TOPIC	TOPIC_ID	N	4	10	0	N	NULL	19	0
B_FORUM_USER	ID	N	4	10	0	N	NULL	19	0
B_FORUM_USER_TOPIC	ID	N	4	10	0	N	NULL	19	0
B_IBLOCK_ELEMENT	DETAIL_TEXT	C	2147483647	0	0	C	4294967295	NULL	NULL
B_IBLOCK_ELEMENT	PREVIEW_TEXT	C	2000	0	0	C	65535	NULL	NULL
B_IBLOCK_ELEMENT_PROPERTY	VALUE	C	2000	0	0	C	65535	NULL	NULL
B_IBLOCK_FIELDS	DEFAULT_VALUE	C	2147483647	0	0	C	4294967295	NULL	NULL
B_LDAP_SERVER	DESCRIPTION	C	5000	0	0	C	65535	NULL	NULL
B_LDAP_SERVER	FIELD_MAP	C	2000	0	0	C	65535	NULL	NULL
B_LEARN_CHAPTER	DETAIL_TEXT	C	2147483647	0	0	C	4294967295	NULL	NULL
B_LEARN_LESSON	DETAIL_TEXT	C	2147483647	0	0	C	4294967295	NULL	NULL
B_LIST_RUBRIC	DESCRIPTION	C	2000	0	0	C	65535	NULL	NULL
B_MAIL_FILTER	ACTION_VARS	C	5000	0	0	C	65535	NULL	NULL
B_MAIL_FILTER	DESCRIPTION	C	2000	0	0	C	65535	NULL	NULL
B_MAIL_FILTER_	STRINGS	C	5000	0	0	C	65535	NULL	NU



COND									LL
B_MAIL_MAILBOX	DESCRIPTION	C	5000	0	0	C	65535	NULL	NULL
B_MAIL_MESSAGE	BODY	C	21474 83647	0	0	C	42949 67295	NULL	NULL
B_MAIL_MESSAGE	FULL_TEXT	C	21474 83647	0	0	C	42949 67295	NULL	NULL
B_MAIL_MSG_ATTACHMENT	FILE_DATA	B	16	0	0	B	42949 67295	NULL	NULL
B_PERF_ERROR	ERRFILE	C	2000	0	0	C	65535	NULL	NULL
B_PERF_ERROR	ERRSTR	C	2000	0	0	C	65535	NULL	NULL
B_PERF_HIT	REQUEST_URI	C	2000	0	0	C	65535	NULL	NULL
B_PERF_HIT	SCRIPT_NAME	C	2000	0	0	C	65535	NULL	NULL
B_PERF_SQL	COMPONENT_NAME	C	2000	0	0	C	65535	NULL	NULL
B_PERF_SQL	MODULE_NAME	C	2000	0	0	C	65535	NULL	NULL
b_search_content	PARAM1	C	1000	0	0	C	65535	NULL	NULL
b_search_content	PARAM2	C	1000	0	0	C	65535	NULL	NULL
b_search_content	SEARCHABLE_CONTENT	C	21474 83647	0	0	C	42949 67295	NULL	NULL
b_search_custom_rank	PARAM1	C	2000	0	0	C	65535	NULL	NULL
b_search_custom_rank	PARAM2	C	2000	0	0	C	65535	NULL	NULL
B_SEC_SESSION	SESSION_DATA	C	21474 83647	0	0	C	42949 67295	NULL	NULL
B_STAT_ADV	DESCRIPTION	C	8000	0	0	C	65535	NULL	NULL



B_STAT_EVENT	DESCRIPTION	C	8000	0	0	C	65535	NULL	NULL
B_STAT_EVENT_LIST	REDIRECT_URL	C	8000	0	0	C	65535	NULL	NULL
B_STAT_EVENT_LIST	REFERER_URL	C	8000	0	0	C	65535	NULL	NULL
B_STAT_EVENT_LIST	URL	C	8000	0	0	C	65535	NULL	NULL
B_STAT_GUEST	FIRST_URL_FROM	C	8000	0	0	C	65535	NULL	NULL
B_STAT_GUEST	FIRST_URL_TO	C	8000	0	0	C	65535	NULL	NULL
B_STAT_GUEST	LAST_CITY_INFO	C	8000	0	0	C	65535	NULL	NULL
B_STAT_GUEST	LAST_COOKIE	C	8000	0	0	C	65535	NULL	NULL
B_STAT_GUEST	LAST_URL_LAST	C	8000	0	0	C	65535	NULL	NULL
B_STAT_GUEST	LAST_USER_AGENT	C	8000	0	0	C	65535	NULL	NULL
B_STAT_HIT	COOKIES	C	8000	0	0	C	65535	NULL	NULL
B_STAT_HIT	URL	C	8000	0	0	C	65535	NULL	NULL
B_STAT_HIT	URL_FROM	C	8000	0	0	C	65535	NULL	NULL
B_STAT_HIT	USER_AGENT	C	8000	0	0	C	65535	NULL	NULL
B_STAT_PAGE	URL	C	2000	0	0	C	65535	NULL	NULL
B_STAT_PATH	PAGES	C	8000	0	0	C	65535	NULL	NULL
B_STAT_PATH_CACHE	PATH_PAGES	C	8000	0	0	C	65535	NULL	NULL



B_STAT_PHRASE_LIST	URL_FROM	C	8000	0	0	C	65535	NULL	NULL
B_STAT_PHRASE_LIST	URL_TO	C	8000	0	0	C	65535	NULL	NULL
B_STAT_REFERER_LIST	URL_FROM	C	8000	0	0	C	65535	NULL	NULL
B_STAT_REFERER_LIST	URL_TO	C	8000	0	0	C	65535	NULL	NULL
B_STAT_SEARCHER	USER_AGENT	C	8000	0	0	C	65535	NULL	NULL
B_STAT_SEARCHER_HIT	URL	C	8000	0	0	C	65535	NULL	NULL
B_STAT_SEARCHER_HIT	USER_AGENT	C	8000	0	0	C	65535	NULL	NULL
B_STAT_SESSION	URL_FROM	C	8000	0	0	C	65535	NULL	NULL
B_STAT_SESSION	URL_LAST	C	8000	0	0	C	65535	NULL	NULL
B_STAT_SESSION	URL_TO	C	8000	0	0	C	65535	NULL	NULL
B_STAT_SESSION	USER_AGENT	C	8000	0	0	C	65535	NULL	NULL
B_STAT_SESSION_DATA	SESSION_DATA	C	8000	0	0	C	65535	NULL	NULL
B_STOP_LIST	COMMENTS	C	8000	0	0	C	65535	NULL	NULL
B_STOP_LIST	MESSAGE	C	8000	0	0	C	65535	NULL	NULL
B_STOP_LIST	URL_FROM	C	8000	0	0	C	65535	NULL	NULL
B_STOP_LIST	URL_REDIRECT	C	8000	0	0	C	65535	NULL	NULL
B_STOP_LIST	URL_TO	C	8000	0	0	C	65535	NULL	NULL



B_STOP_LIST	USER_AGENT	C	8000	0	0	C	65535	NULL	NULL
B_TICKET	AUTO_CLOSE_DAYS	N	1	3	0	N	NULL	10	0
B_TICKET	LAST_MESSAGE_SID	C	8000	0	0	C	65535	NULL	NULL
B_TICKET	OWNER_SID	C	8000	0	0	C	65535	NULL	NULL
B_TICKET	SUPPORT_COMMENTS	C	8000	0	0	C	65535	NULL	NULL
B_TICKET	TITLE	C	2000	0	0	C	65535	NULL	NULL
B_TICKET_DICTIONARY	DESCR	C	8000	0	0	C	65535	NULL	NULL
B_TICKET_MESSAGE	EXTERNAL_FIELD_1	C	8000	0	0	C	65535	NULL	NULL
B_TICKET_MESSAGE	MESSAGE	C	2147483647	0	0	C	4294967295	NULL	NULL
B_TICKET_MESSAGE	MESSAGE_SEARCH	C	2147483647	0	0	C	4294967295	NULL	NULL
B_TICKET_MESSAGE	OWNER_SID	C	8000	0	0	C	65535	NULL	NULL
B_TICKET_SLA	DESCRIPTION	C	8000	0	0	C	65535	NULL	NULL
B_VOTE	DESCRIPTION	C	5000	0	0	C	65535	NULL	NULL
B_VOTE_ANSWER	FIELD_TYPE	N	1	3	0	N	NULL	10	0
B_VOTE_ANSWER	MESSAGE	C	5000	0	0	C	65535	NULL	NULL
B_VOTE_EVENT_ANSWER	MESSAGE	C	8000	0	0	C	65535	NULL	NULL
B_VOTE_QUESTION	QUESTION	C	5000	0	0	C	65535	NULL	NULL



B_WORKFLOW_DOCUMENT	COMMENTS	C	8000	0	0	C	65535	NULL	NULL
B_WORKFLOW_LOG	COMMENTS	C	8000	0	0	C	65535	NULL	NULL
B_WORKFLOW_STATUS	DESCRIPTION	C	8000	0	0	C	65535	NULL	NULL

⚠ Примечание. Строго говоря, следует исследовать тождественность признаков NULL у соответствующих полей и других ограничений. Например, если некоторое поле допускает NULLы в MySQL, но является NOT NULL в SQL Server, перенос данных может завершиться с ошибкой. Если сразу перейти к процессу переноса данных и такие несоответствия имеются, то они будут выявлены в ходе переноса.

Глава 3. Перенос данных

Теперь можно приступить к собственно переносу данных. Однако использовать **MySQL ODBC Connector 5.1** для переноса данных - не лучший вариант, поскольку есть ситуации, в которых его работоспособность нарушается. В их числе отсутствие поддержки полей типа **longtext**.

```
select * from openquery(mysql, 'select DETAIL_TEXT from b_learn_lesson')
```

Msg 0, Level 11, State 0, Line 0

A severe error occurred on the current command. The results, if any, should be discarded.

Можно обеспечить перенос подстроками по 4000 символов, но такой способ нельзя признать оптимальным.

Установка MySQL Connector/Net 6.0

Рекомендуется использовать для переноса данных MySQL Connector/Net 6.0, который можно скачать по адресу <http://dev.mysql.com/downloads/connector/net/6.0.html>.

Первый шаг установки коннектора:



Рисунок 3.1 Первый шаг инсталлятора

Ø На втором шаге (Рисунок 3.2) выберите нужные компоненты для установки. Если не уверены в выборе, то оставьте параметры по умолчанию.

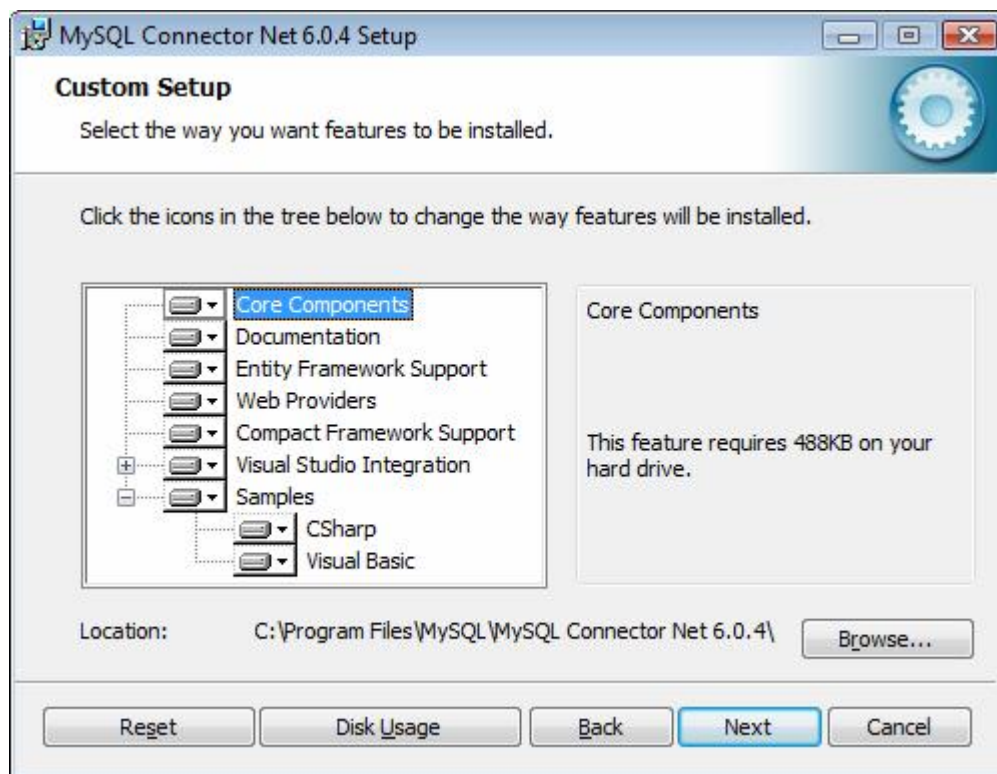


Рисунок 3.2 Второй шаг установки

На третьем шаге произойдет собственно установка коннектора.

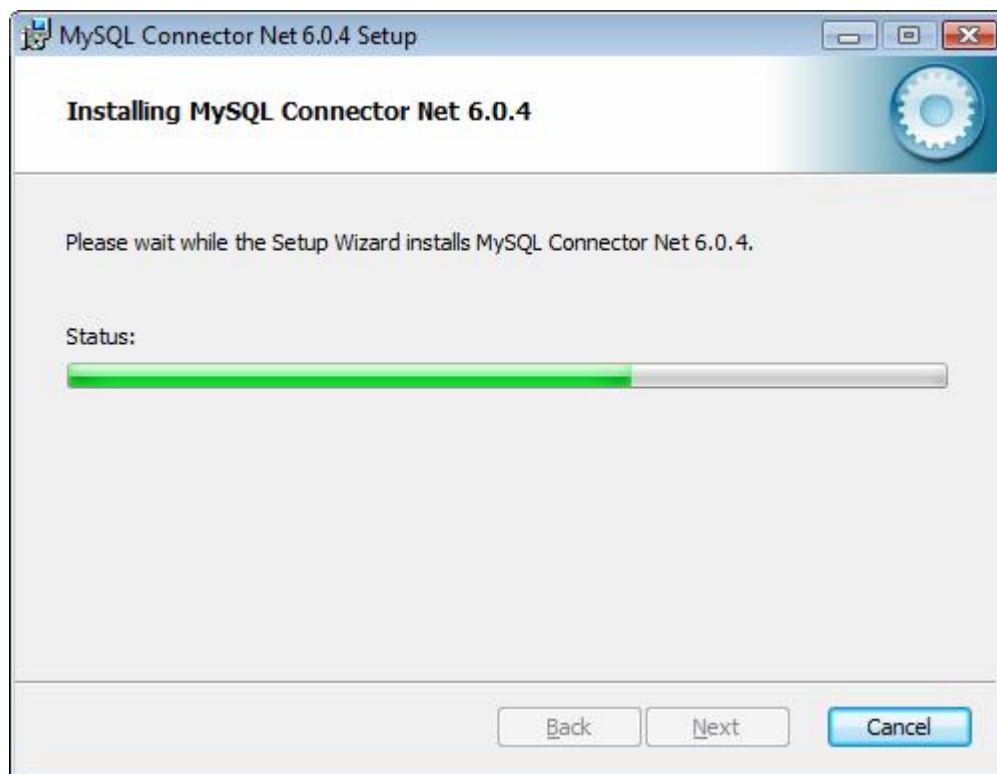


Рисунок 3.3 Собственно установка коннектора

После завершения установки **MySQL Connector/Net 6.0** автоматически добавит новые пространства имен **MySql.Data** в .NET.



Перенос с помощью Visual Studio

В качестве рабочего инструмента для миграции можно использовать **Visual Studio**, в частности, ее бесплатную редакцию - <http://www.microsoft.com/express/download/>.

- Ø Установите **Visual Studio** как это описано в документации Microsoft.
- Ø Откройте **Visual Studio** и добавьте ссылку на добавленные пространства имен (Рисунок 3.4).

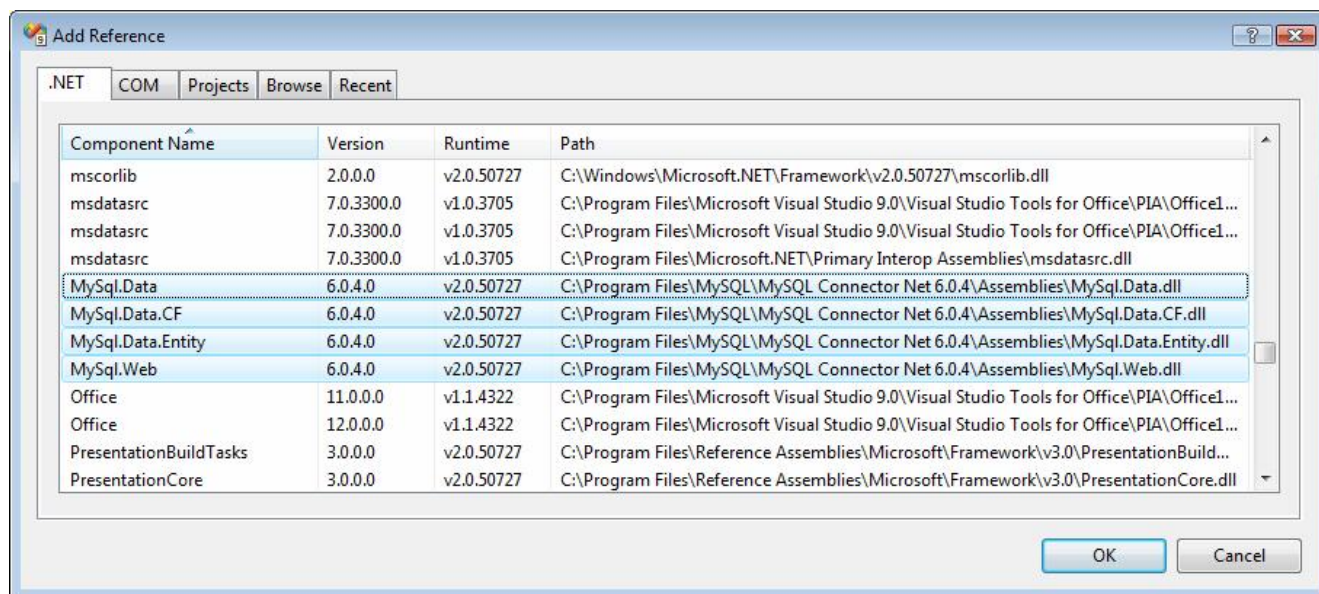


Рисунок 3.4 Добавление нового пространства имен

- Ø выберите в меню **File > New > Project** (Рисунок 3.5).

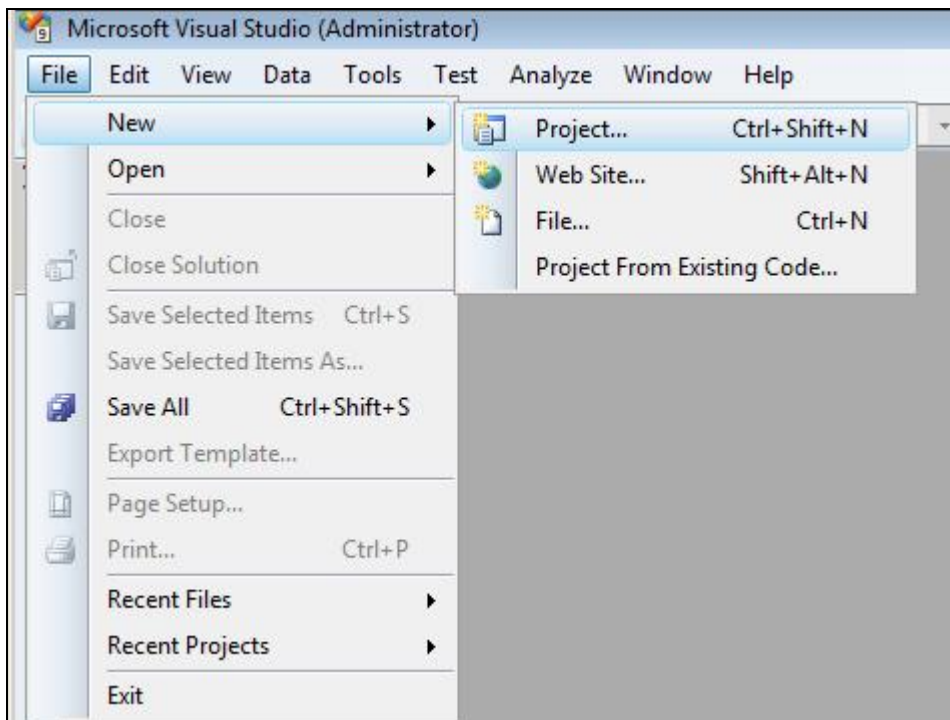


Рисунок 3.5 Создание нового проекта

Ø Выберите в качестве шаблона проекта создание нового консольного приложения на **C#** (Рисунок 3.6):

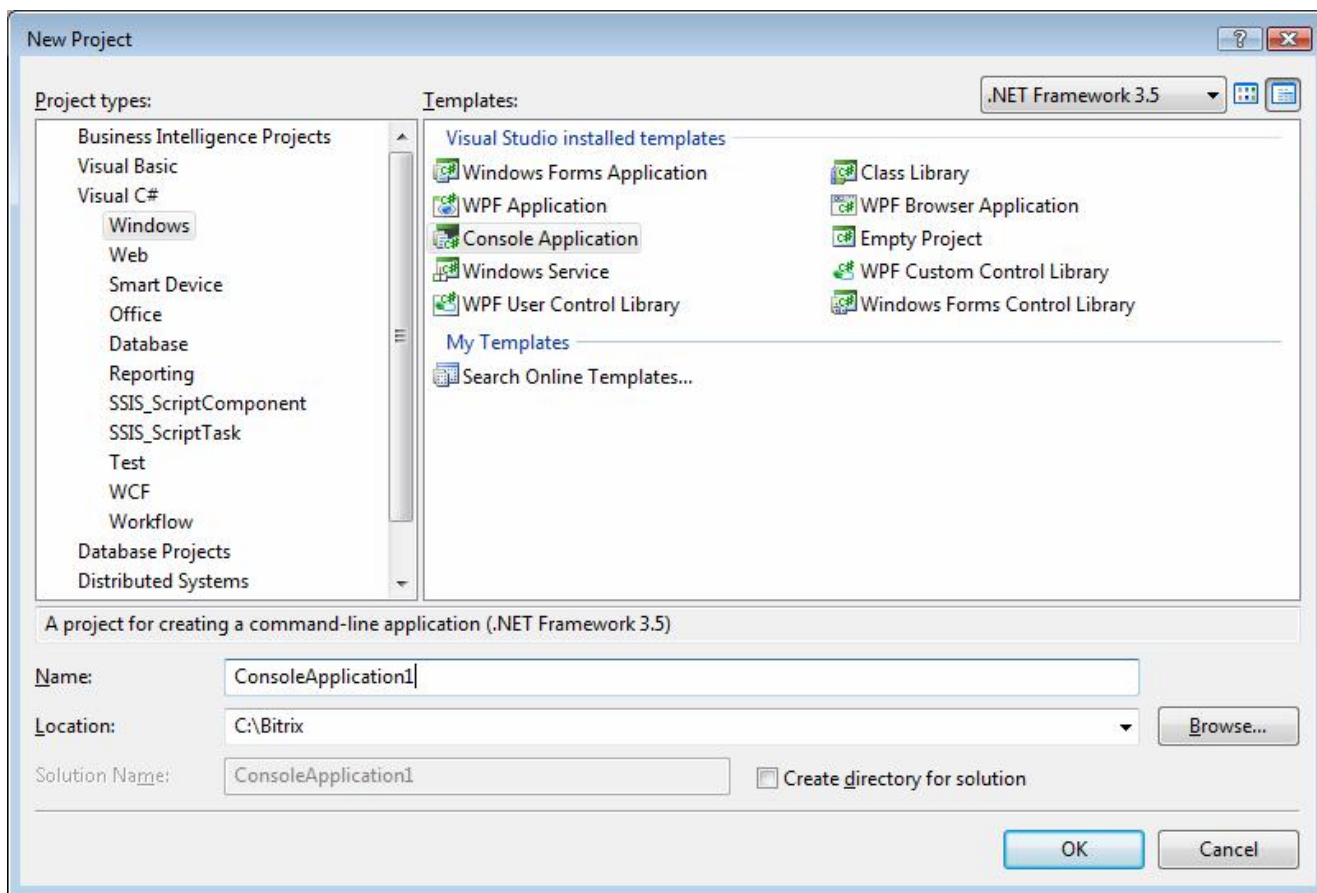


Рисунок 3.6 Выбор шаблона

Ø Добавьте к ссылкам (**References**) проекта пространство имен **MySQL.Data**, которое добавляет **MySQL Connector/Net 6.0** (Рисунок 3.7).

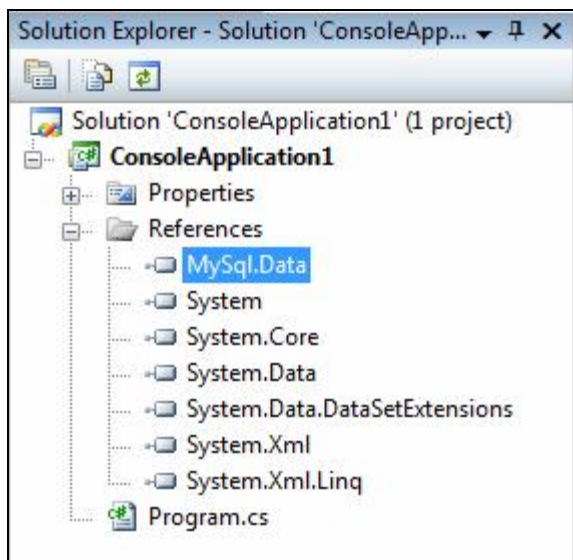


Рисунок 3.7 Добавление пространства имен MySQL.Data

Ø выполните следующий код:



```
using System;
using System.Text;

using MySql.Data.MySqlClient;
using System.Data;
using System.Diagnostics;
using System.Data.SqlClient;

class Program
{
    static MySqlConnection mySqlCnn;
    static SqlConnection sqlSrvCnn;

    static void Main(string[] args)
    {
        sqlSrvCnn = new
        SqlConnection(@"server=(local)\SQLEXPRESS;database=bitrix;trusted_connection=true;MultipleActive
        ResultSets=true");
        sqlSrvCnn.Open();
        mySqlCnn = new
        MySqlConnection("server=127.0.0.1;port=31006;uid=root;pwd=;database=bsm_demo;Pooling=False"
        );
        mySqlCnn.Open();

        DisableFKConstraints(true);
        DataTable tblList = GetSourceTablesFromMySQLDB();
        CleanDestTablesInSQLSrvDB(tblList);
        TransferData(tblList);
        DisableFKConstraints(false);

        mySqlCnn.Close();
        sqlSrvCnn.Close();
    }

    /// <summary>
    /// Копирует данные из таблицы в MySQL в одноименную таблицу в SQL Server
```



/// Предполагается, что множества имен полей в таблицах совпадают. Порядок может отличаться.

/// </summary>

/// <param name="tblName">Имя таблицы</param>

static void CopyDataFromMySQLTblToCorrespondingSQLSrvTbl(string tblName)

{

///Читаем по порядку поля в таблице-назначения

SqlCommand sqlSrvCmd = sqlSrvCnn.CreateCommand();

sqlSrvCmd.CommandText = "select name from sys.columns where object_id = object_id(@tblName) order by column_id";

sqlSrvCmd.Parameters.AddWithValue("@tblName", tblName);

SqlDataReader sqlSrvDr = sqlSrvCmd.ExecuteReader(CommandBehavior.SingleResult);

///Составляем строку запроса для источника, перечисляя туда поля в том порядке, как они следуют в назначении

StringBuilder mySqlCommandText = new StringBuilder("select ");

///Имя поля заключаем в аналог квадратных скобок - на случай, если оно будет совпадать с одним из зарезервированных слов MySQL.

while (sqlSrvDr.Read()) mySqlCommandText.Append("'" + sqlSrvDr.GetSqlString(0).Value + "',");

sqlSrvDr.Close();

mySqlCommandText.Remove(mySqlCommandText.Length - 1, 1);

mySqlCommandText.Append(" from " + tblName);

MySqlCommand mySqlCommand = new MySqlCommand(mySqlCommandText.ToString(), mySqlCnn);

MySqlDataReader mySqlDr = mySqlCommand.ExecuteReader();

SqlBulkCopy bcp = new SqlBulkCopy(sqlSrvCnn, SqlBulkCopyOptions.KeepIdentity, null);

///KeepIdentity означаем set identity_insert <tblName> on/off

///Поскольку в mySqlDr поля идут в том же порядке, что и в назначении, SqlBulkCopy.ColumnMappings не требуется.

bcp.DestinationTableName = tblName;

// Заправляем шланг ридера объекту SqlBulkCopy, чтобы он качал из него содержимое в bcp.DestinationTableName

bcp.WriteToServer(mySqlDr);

mySqlDr.Close();

}



```
/// <summary>
/// Получает список таблиц из MySQLной базы
/// </summary>
/// <returns>Список таблиц</returns>
static DataTable GetSourceTablesFromMySQLDB()
{
    DataTable tbl = new DataTable();
    tbl.Load(new MySqlCommand("show tables", mySqlCnn).ExecuteReader());
    return tbl;
}

/// <summary>
/// Удаляет в каждой таблице из списка все ее записи
/// </summary>
/// <param name="tblList">Список таблиц</param>
static void CleanDestTablesInSQLSrvDB(DataTable tblList)
{
    Debug.WriteLine("Очистка таблиц назначения...");
    foreach (DataRow r in tblList.Rows)
    {
        new SqlCommand("delete " + r[0].ToString(), sqlSrvCnn).ExecuteNonQuery();
        Debug.WriteLine("Очищена таблица " + r[0].ToString());
    }
    Debug.WriteLine("Очистка закончена.");
}

static void TransferData(DataTable tblList)
{
    Debug.WriteLine("Загрузка данных...");
    foreach (DataRow r in tblList.Rows)
    {
        CopyDataFromMySQLTblToCorrespondingSQLSrvTbl(r[0].ToString());
        Debug.WriteLine("Перенесена таблица " + r[0].ToString());
    }
    Debug.WriteLine("Загрузка завершена.");
}
```



```
}

/// <summary>
/// Процедура отключает/включает все ограничения внешнего ключа над таблицами в БД
SQL Server
/// </summary>
/// <param name="switchOff">Если да, то отключить, нет - включить</param>
static void DisEnableFKConstraints(bool switchOff)
{
    string prefix = switchOff ? "Om" : "B";
    Debug.WriteLine(prefix + "ключение FK-ограничений...");
    SqlDataReader sdr = new SqlCommand("select name, object_name(parent_object_id) from
sys.foreign_keys", sqlSrvCnn).ExecuteReader();
    while (sdr.Read())
    {
        string fkName = sdr.GetString(0), tblName = sdr.GetString(1);
        new SqlCommand(String.Format("alter table {0} {1}check constraint {2}", tblName, switchOff ?
"no" : "", fkName), sqlSrvCnn).ExecuteNonQuery();
        Debug.WriteLine(String.Format("{0}ключено ограничение {1} в таблице {2}", prefix,
fkName, tblName));
    }
    sdr.Close();
    Debug.WriteLine(prefix + "ключение FK-ограничений завершено.");
}
}
```

Комментарии к коду

Необходимо сделать некоторые комментарии к коду.

Как показывает

```
select * from sys.objects where type = 'F'
```

(или `sys.foreign_keys`/ `sys.foreign_key_columns`) в базе имеются ограничения внешнего ключа. Следовательно, первоначально следует вставлять данные в **referenced_object** (PK), а затем в **parent_object** (FK), чтобы избежать нарушений ограничений внешнего ключа.

Возможны ситуации, когда **referenced_object** сам, в свою очередь, имеет **referenced_object**. Следовательно, требуется упорядочить таблицы, выбрав сначала те **referenced_objects**, которые не имеют FK-ограничений, вставить данные в них, затем в те таблицы, для которых они являются РК-таблицами и т.д.



Чтобы не усложнять скрипт миграции, было принято решение на время переноса данных отключить все FK-ограничения, вставить данные, а затем снова включить. Отключение FK-ограничений выполняется при помощи команды:

```
ALTER TABLE <имя FK таблицы> NOCHECK CONSTRAINT <имя ограничения>
```

А включение, соответственно, - *CHECK*.

Отключение/включение ограничений внешнего ключа делает процедура **DisableFKConstraints** (bool switchOff). В том, что FK-ограничения отключены, можно убедиться по запросу

```
select * from sys.foreign_keys
```

в результатах которого колонка **is_disabled** стала 1 для всех записей.

Перед загрузкой данных содержимое таблиц **SQL Express** следует очистить. Несмотря на отключенные ограничения чистить таблицы при помощи команды *TRUNCATE TABLE* не получится. Приходится использовать команду *DELETE <имя таблицы>* для удаления из каждой таблицы всех ее записей.

По команде

```
select * from sys.columns where is_identity = 1
```

или

```
select * from sys.identity_columns
```

можно увидеть, что на некоторых таблицах имеются колонки с автоинкрементом. Однако специально отключать автоинкремент перед вставкой *SET IDENTITY_INSERT <имя таблицы> ON | OFF* не требуется, т.к. это "за сценой" делает объект **SqlBulkCopy** при помощи параметра **KeepIdentity**.

Общая последовательность действий по переносу данных выглядит следующим образом:

- Открываются соединения с БД **MySQL** и **SQL Express**.
- Включается **MARS** в **SQL Server**'ном соединении для выполнения процедуры **DisableFKConstraints**, где держится на соединении открытый **DataReader** со списком FK. По этому списку на каждой записи выполняется команда *ExecuteNonQuery()* на том же соединении.
- Отключаются все ограничения внешнего ключа в БД **SQL Express**, чтобы не заботиться о последовательности очистки и заливки.
- Получается список таблиц из БД **MySQL**. Он сохраняется в **DataTable tblList**.
- Очищаются таблицы из этого списка.
- Последовательно по этому списку переносятся данные из каждой таблицы **MySQL** в одноименную таблицу **SQL Express**.

Перед выполнением загрузки из **MySQL** рекомендуется выполнить резервное копирование базы данных на **SQL Express** несмотря на то, что она пуста, т.е. содержит только "заводские" установки и весь наработанный контент хранится в базе **MySQL**. Резервное копирование базы данных **SQL Server** можно выполнить при помощи команды:

```
backup database Bitrix to disk = 'c:\Bitrix\bitrix.bak' with noformat, init, name = N'Bitrix-Full Database
```

Backup', skip, stats = 10

Восстановление (при необходимости) выполняется командой

alter database Bitrix set single_user with rollback immediate

use master

restore database Bitrix from disk = 'c:\Bitrix\Bitrix.bak' with recovery, stats = 20

Перенос данных с помощью PowerShell

Если у клиента нет Visual Studio или он по каким-либо причинам не может установить ее Express-редакцию, ниже приводится вариация скрипта, мигрирующего базу «1С-Битрикс: Управление сайтом» с **MySQL** на **SQL Express**, на языке сценариев **PowerShell**:

cls

function DisableFKConstraints([bool] \$switchOff)

{

[string] \$prefix; if (\$switchOff) { \$prefix = "Om" } else { \$prefix = "B" };

Write-Host (\$prefix + "ключенiе FK-ограничений...")

*[System.Data.SqlClient.SqlDataReader] \$sdr = (New-Object
System.Data.SqlClient.SqlCommand("select name, object_name(parent_object_id) from
sys.foreign_keys", \$sqlSrvCnn)).ExecuteReader()*

while (\$sdr.Read())

{

[string] \$fkName = \$sdr.GetString(0); [string] \$tblName = \$sdr.GetString(1)

[string] \$prefix1 = ""; if (\$switchOff) {\$prefix1 = "no"}

*[string] \$cmdText = "alter table {0} {1}check constraint {2}" -f \$tblName, \$prefix1,
\$fkName*

*(New-Object System.Data.SqlClient.SqlCommand(\$cmdText,
\$sqlSrvCnn)).ExecuteNonQuery()*

Write-Host ("{0}ключено ограничение {1} в таблице {2}" -f \$prefix, \$fkName, \$tblName)

}

\$sdr.Close();

Write-Host (\$prefix + "ключенiе FK-ограничений завершено.")

}

function CleanDestTablesInSQLSrvDB([System.Data.DataTable] \$tblList)

{

Write-Host "Очистка таблиц назначения..."



```

foreach ($r in $tblList.Rows)
{
    [string] $cmdText = "delete " + $r[0]
    (New-Object System.Data.SqlClient.SqlCommand($cmdText,
    $sqlSrvCnn)).ExecuteNonQuery()
    Write-Host ("Очищена таблица " + $r[0])
}
Write-Host "Очистка закончена."
}

function TransferData([System.Data.DataTable] $tblList)
{
    Write-Host "Загрузка данных..."
    foreach ($r in $tblList.Rows)
    {
        CopyDataFromMySQLTblToCorrespondingSQLSrvTbl($r[0])
        Write-Host ("Перенесена таблица " + $r[0])
    }
    Write-Host "Загрузка завершена."
}

function CopyDataFromMySQLTblToCorrespondingSQLSrvTbl([string] $tblName)
{
    [System.Data.SqlClient.SqlCommand] $sqlSrvCmd = $sqlSrvCnn.CreateCommand()
    $sqlSrvCmd.CommandText = "select name from sys.columns where object_id =
    object_id(@tblName) order by column_id"
    $sqlSrvCmd.Parameters.AddWithValue("@tblName", $tblName)
    [System.Data.SqlClient.SqlDataReader] $sqlSrvRdr = $sqlSrvCmd.ExecuteReader()
    [System.Text.StringBuilder] $mySqlCommandText = New-Object System.Text.StringBuilder("select ")
    while ($sqlSrvRdr.Read()) { $mySqlCommandText.Append("`" + $sqlSrvRdr.GetSqlString(0) + "`",") }
    $sqlSrvRdr.Close()
    $mySqlCommandText.Remove($mySqlCommandText.Length - 1, 1)
    $mySqlCommandText.Append(" from " + $tblName)
    [MySql.Data.MySqlClient.MySqlCommand] $mySqlCommand = New-Object
    MySql.Data.MySqlClient.MySqlCommand($mySqlCommandText.ToString(), $mySqlCnn);
    [MySql.Data.MySqlClient.MySqlDataReader] $mySqlRdr = $mySqlCommand.ExecuteReader()

```



```
[System.Data.SqlClient.SqlBulkCopy]                $bcp                =                New-Object
System.Data.SqlClient.SqlBulkCopy($sqlSrvCnn,
[System.Data.SqlClient.SqlBulkCopyOptions]::KeepIdentity, $null)

    $bcp.DestinationTableName = $tblName
    $bcp.WriteToServer($mySqlRdr)
    $bcp.Close()

$mySqlRdr.Close()
}

##### MAIN
#####
#####

[System.Data.SqlClient.SqlConnection]                $sqlSrvCnn                =                New-Object
System.Data.SqlClient.SqlConnection("server=(local)\SQLEXPRESS;database=bitrix;trusted_connection=true;MultipleActiveResultSets=true")
$sqlSrvCnn.Open()
[void][system.reflection.Assembly]::LoadWithPartialName("MySql.Data")
[MySql.Data.MySqlClient.MySqlConnection]                $mySqlCnn                =                New-Object
MySql.Data.MySqlClient.MySqlConnection("server=127.0.0.1;port=31006;uid=root;pwd=;database=bsm_demo;Pooling=False")
$mySqlCnn.Open()

$mySqlRdr    =    (New-Object    MySql.Data.MySqlClient.MySqlCommand("show    tables;",
$mySqlCnn)).ExecuteReader()

[System.Data.DataTable] $tblList = New-Object System.Data.DataTable
$tblList.Load($mySqlRdr)
$mySqlRdr.Close()

DisableFKConstraints $true | Out-Null
CleanDestTablesInSQLSrvDB $tblList
TransferData $tblList | Out-Null
DisableFKConstraints $false | Out-Null

$sqlSrvCnn.Close()
$mySqlCnn.Close()
```

Глава 4. Общая последовательность действий

Общая последовательность действий по переносу данных на локальной установке:

- Ø Установить **SQL Express**.
- Ø Переименовать каталог **www** в папке **Bitrix Environment** и установить «1С-Битрикс: Управление сайтом» на **SQL Express**.
- Ø Установить **.NET Connector** к **MySQL**.
- Ø Закрыть все вкладки в браузере с локальной версией сайта, остановить процесс **Bitrix Environment**.
- Ø Проверить, что **MySQL** по-прежнему запущен (**mysqld-opt.exe** значится в числе работающих процессов в Диспетчере задач). Если нет, то запустить процесс командой:

```
"C:\Program Files\Bitrix Environment\mysql\bin\mysqld-opt.exe" --port=31006
```

- Ø Провести перенос данных с помощью **Visual Studio** или **PowerShell** (Рисунок 4.1).

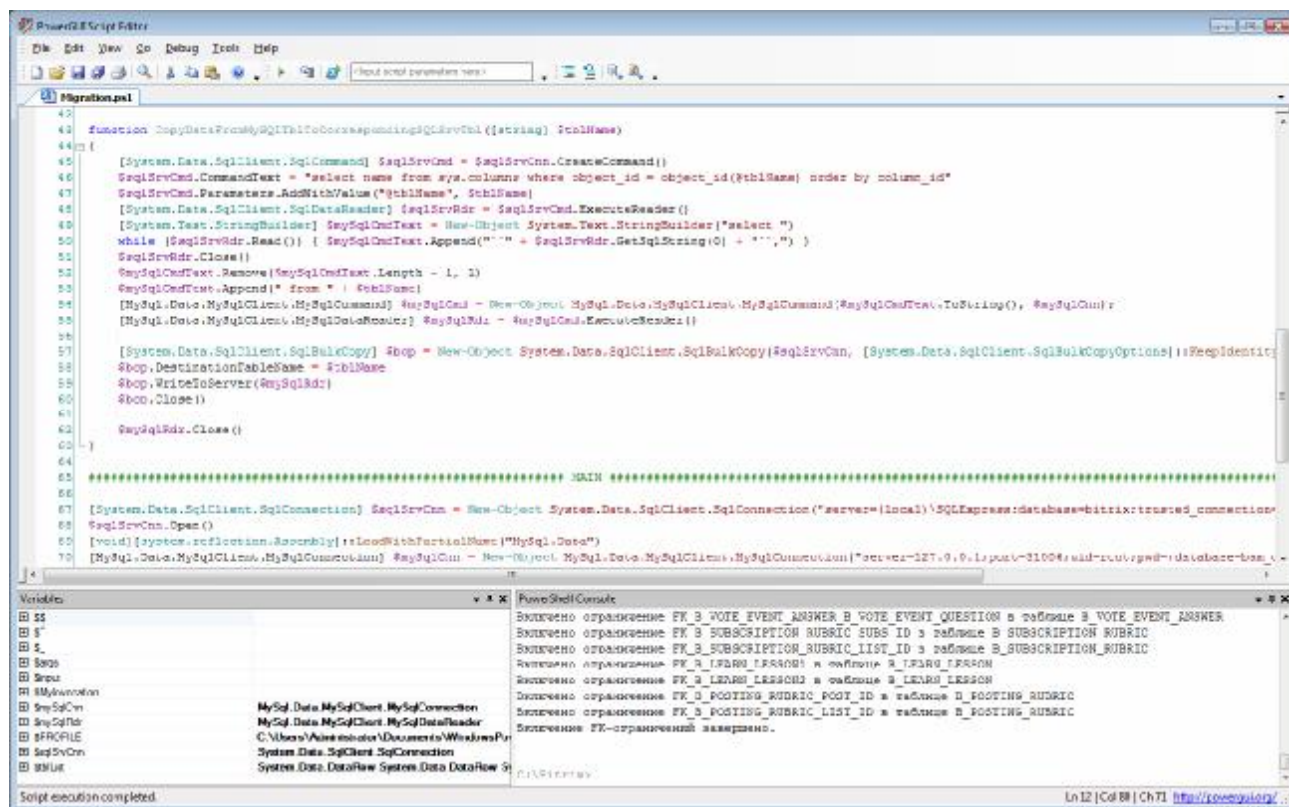


Рисунок 4.1 Перенос данных с помощью сценария PowerShell

- Ø Вновь запустить **Bitrix Environment** и открыть сайт.



 **Примечание.** При возникновении непредвиденной ситуации вернуться на исходную позицию можно, вернув переименованной папке с установкой «1С-Битрикс: Управление сайтом» на MySQL имя `www`.



Заключение

В «Примере» рассмотрена технология миграции сайт с базы MySQL на базу MS SQL. Если по его изучению у вас остались вопросы, обращайтесь на [форумы](#) компании 1С-Битрикс.