

Министерство науки и высшего образования Российской Федерации

федеральное государственное бюджетное образовательное учреждение
высшего образования
«Санкт-Петербургский государственный технологический институт
(технический университет)»

Кафедра систем автоматизированного проектирования и управления

В.Ю. Плонский

Проектирование баз данных в СУБД MySQL

Лабораторный практикум

**Санкт-Петербург
2018**

ЦЕЛЬ РАБОТЫ

- 1. Изучение основ администрирования СУБД MySQL (создание объектов базы данных; работа со словарем данных; резервное копирование и восстановление).
- 2. Изучение подключения к базам данных MySQL через ODBC-интерфейс.
- 3. Изучение работы с запросами в СУБД MySQL.

Примечание: для работы в аудитории используется СУБД MySQL 5.1; среда разработки dbForge Studio for MySQL 8.0; СУБД LibreOffice Base 6.1.

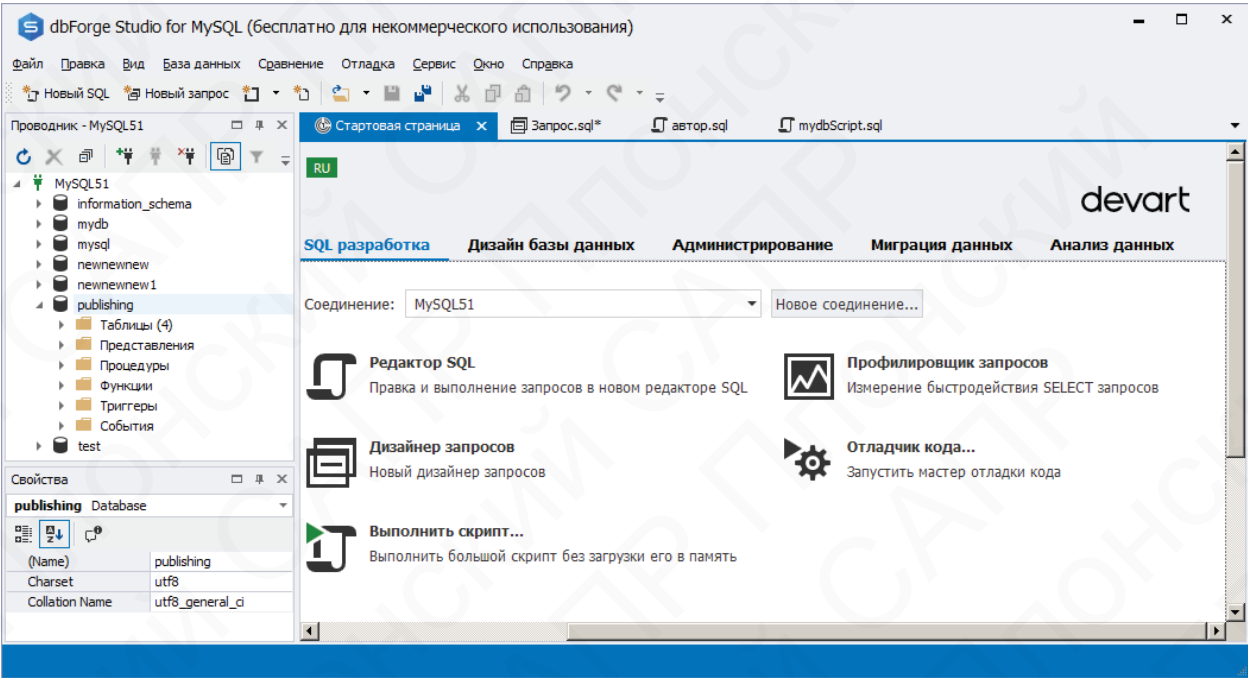


Рисунок 1. Основное окно dbForge Studio for MySQL

ПОРЯДОК ВЫПОЛНЕНИЯ

- 1. Конвертация модели данных для СУБД MySQL.
Необходимо предварительно провести конвертацию модели БД, разработанной в Toad Data Modeler в модель БД MySQL. Откройте существующую модель:

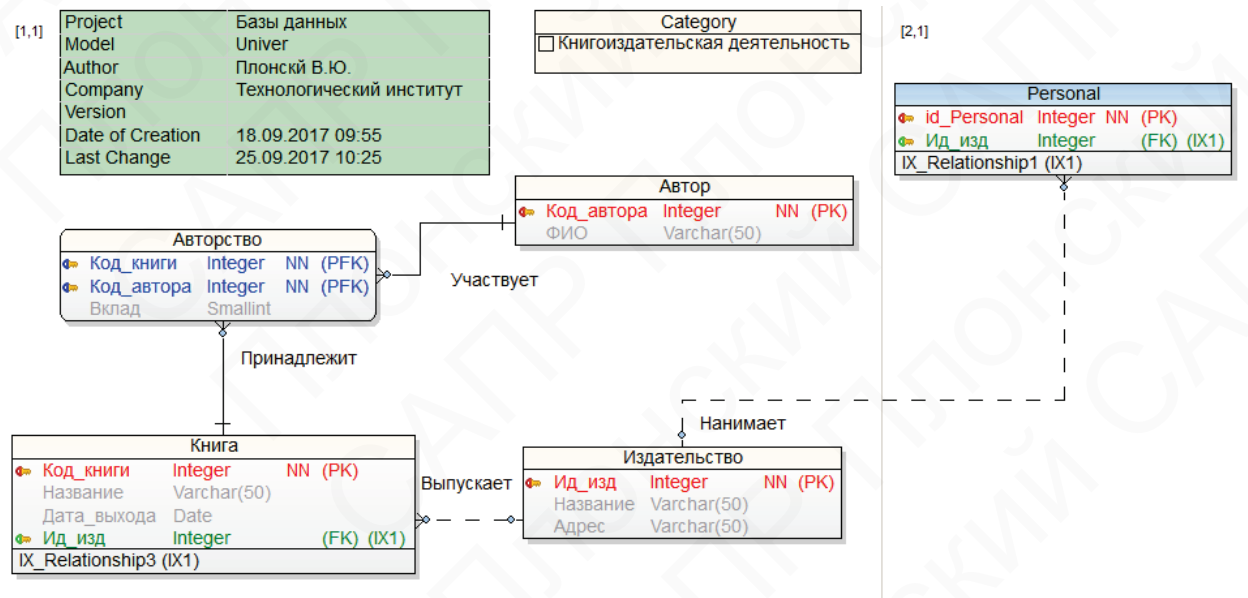


Рисунок 2. Исходная модель данных предметной области

Выберите в меню «Model» команду «Convert Model» и далее «Run». В диалоговом окне «Model Conversion» укажите СУБД (включая версию), для которой будет создана модель данных:

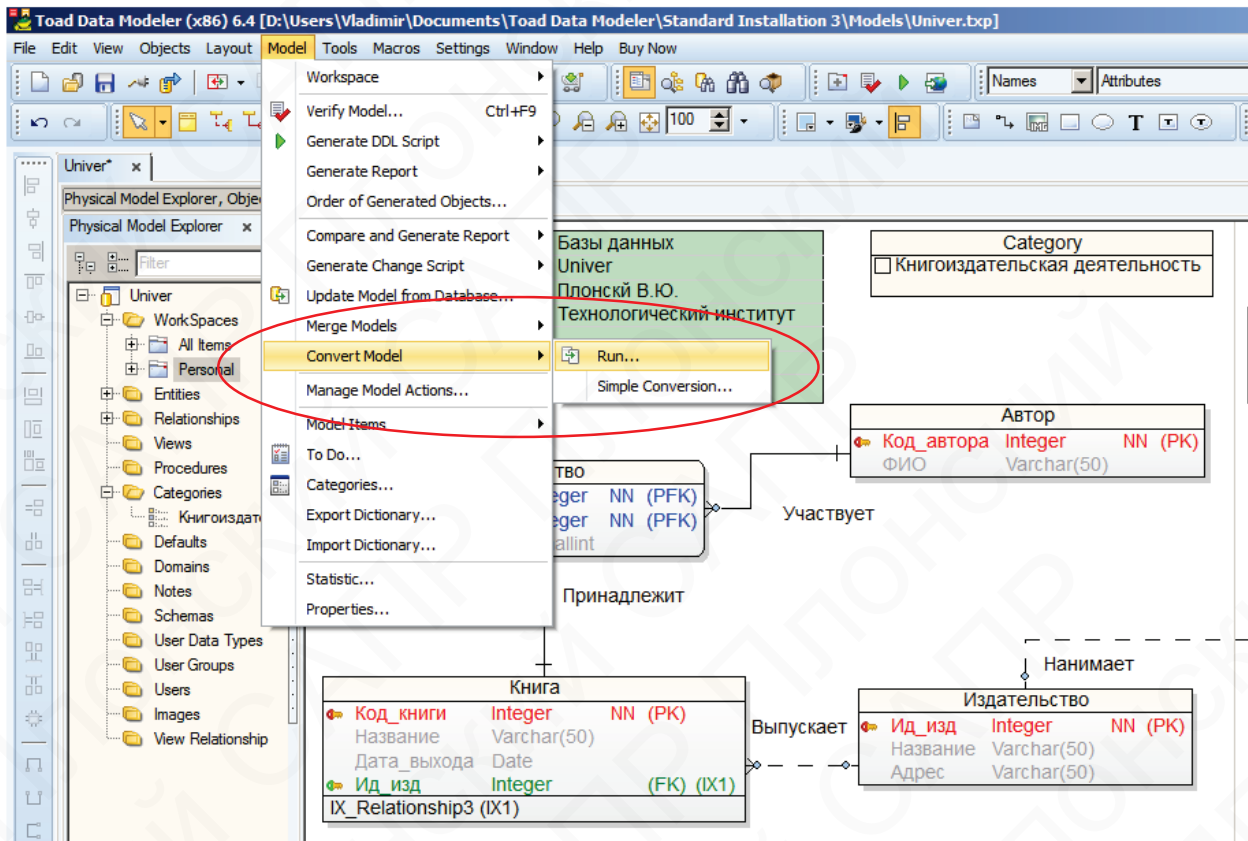


Рисунок 3. Конвертация модели данных

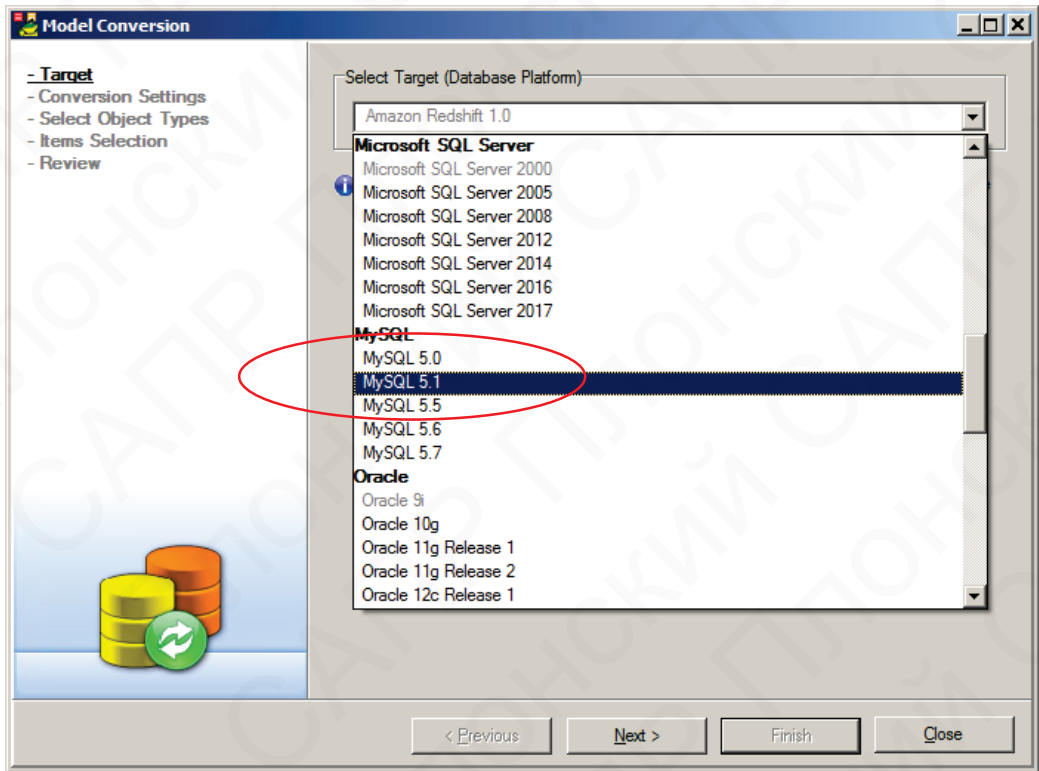


Рисунок 4. Выбор целевой СУБД

2. Создание базы данных MySQL.

2.1. Генерация скрипта для создания объектов базы данных.

Выберите в меню «Model», затем «Generate DDL Script» и «Run»:

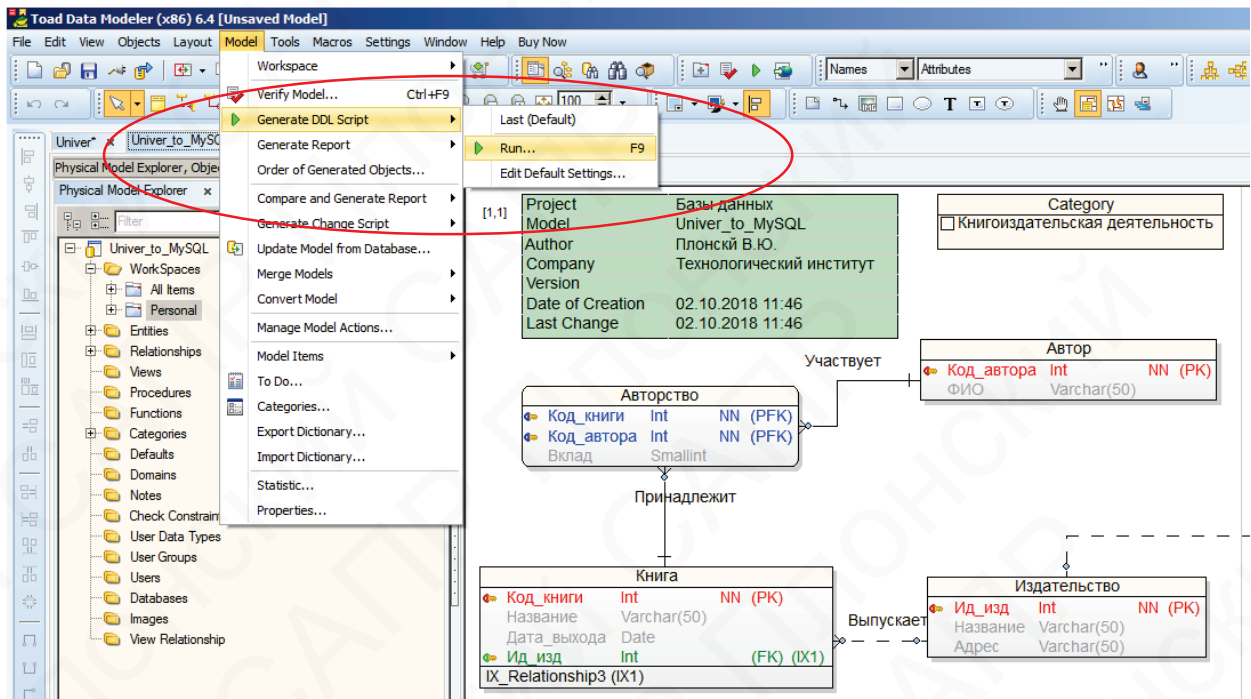


Рисунок 5. Запуск процесса создания DDL-скрипта

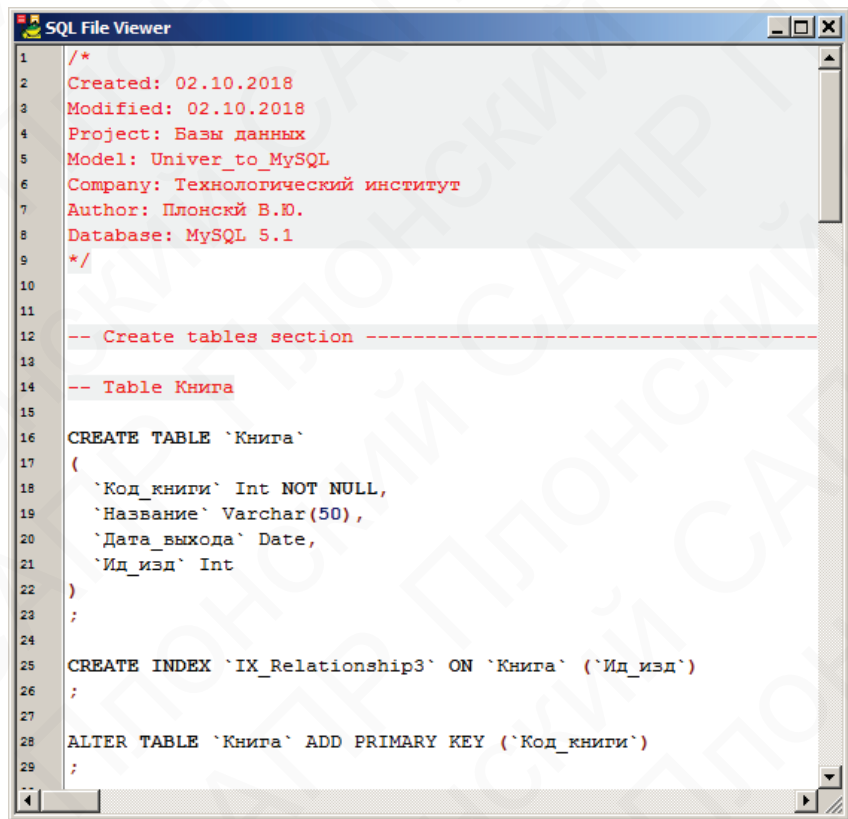


Рисунок 6. Результат – описание схемы БД на языке SQL

Примечание: обратите внимание, что полученный скрипт не содержит в себе команды для создания БД, а только команды для объектов БД (таблицы, ограничения, индексы).

2.2. Просмотр списка существующих баз данных.

На установленном сервере MySQL можно получить список БД следующим образом:
Вариант 1: запустить утилиту командной строки (разновидность клиента MySQL). Предполагается, что сервер уже запущен, например, в виде сервиса.

Примечание: для работы в аудитории логин и пароль уточните у преподавателя (по умолчанию пользователь – root, пароль – root).

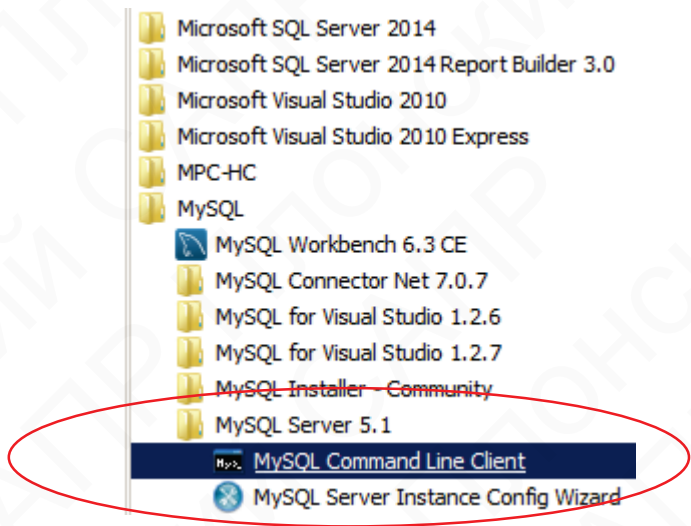


Рисунок 7. Запуск клиента командной строки MySQL

После ввода пароля для пользователя root@localhost можно выполнить команду:
mysql> SHOW DATABASES;

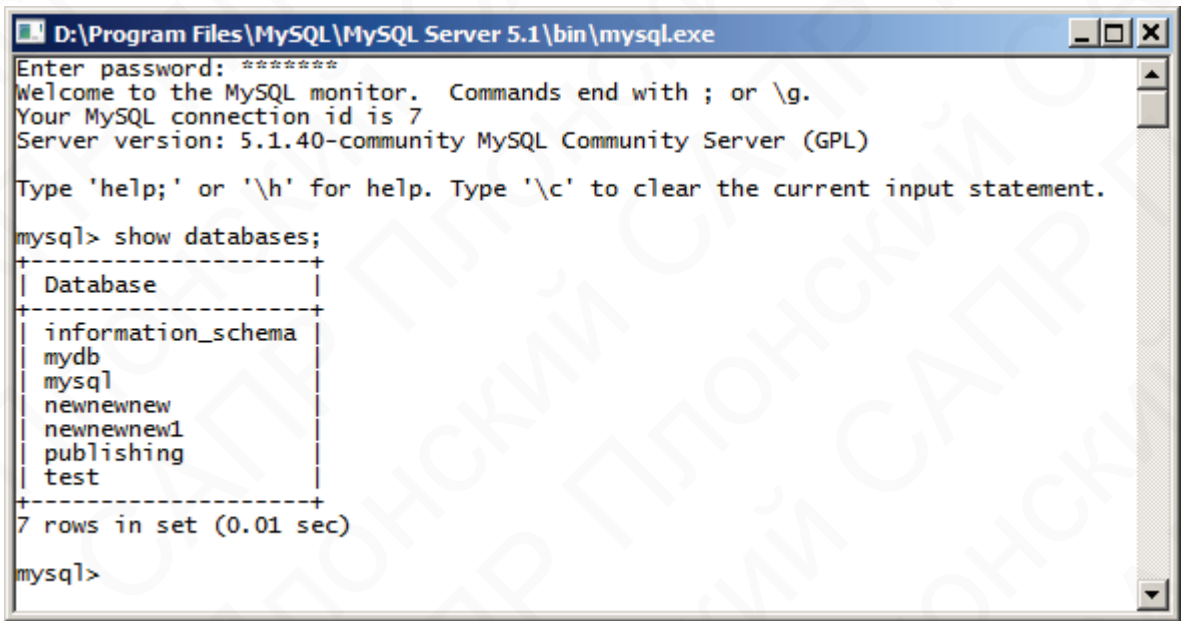


Рисунок 8. Запуск клиента командной строки MySQL

Вариант 2: список можно посмотреть в визуальных средах (таких как dbForge Studio for MySQL, MySQL Workbench).
Для этого надо предварительно открыть соединение, а в случае, если соединение ещё не создано, добавить его, указав параметры соединения с сервером:

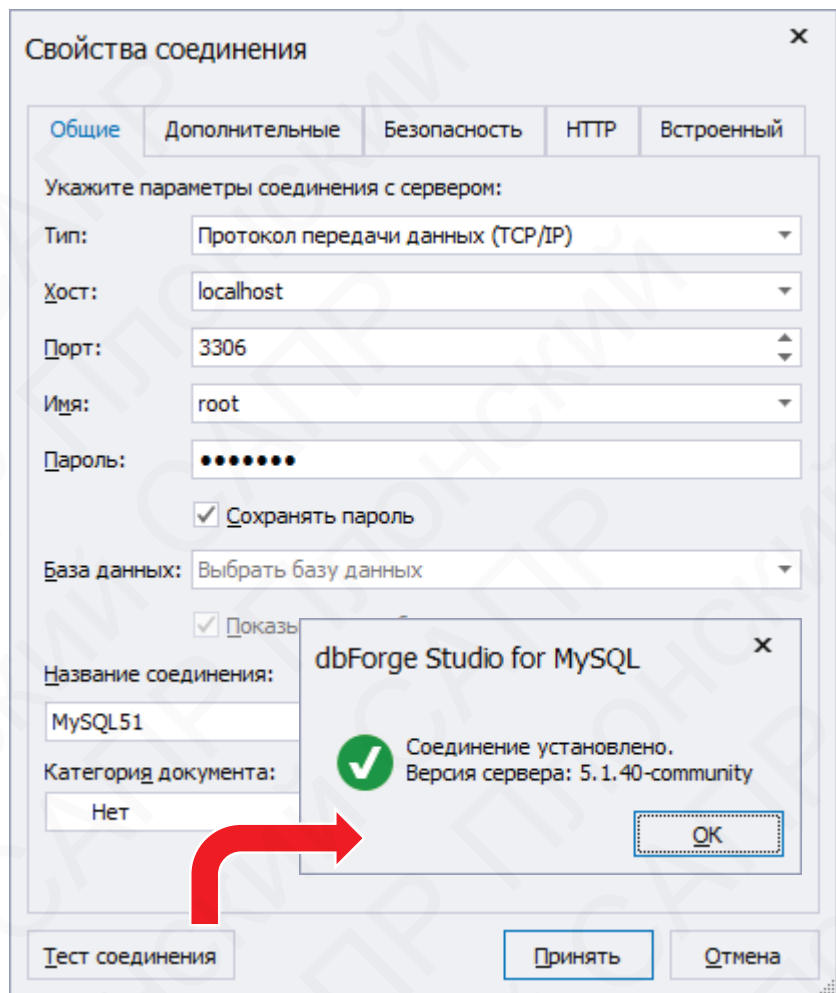


Рисунок 9. Запуск клиента командной строки MySQL

Обратите внимание на значения, указанные в свойствах соединения:

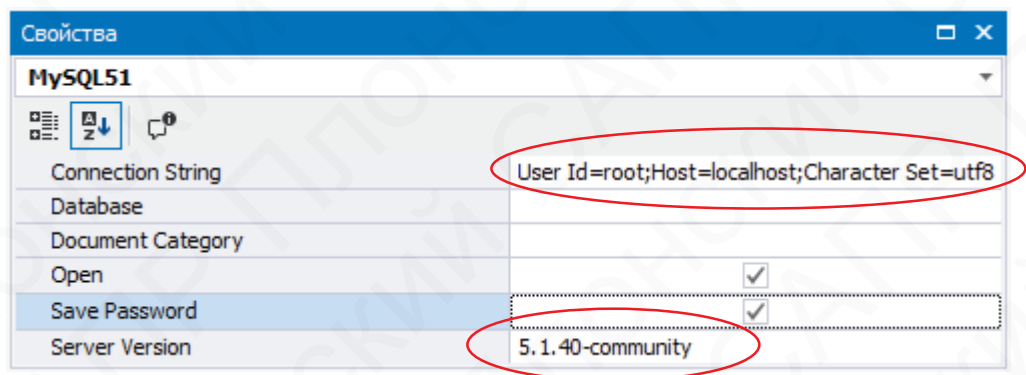


Рисунок 10. Окно свойства соединения

2.3. Создание базы данных.

Есть несколько вариантов создания БД, описанные ниже. Выполните вариант 1 или вариант 3.

Вариант 1: выполнить команду CREATE DATABASE в клиенте командной строки MySQL:

```
mysql> CREATE DATABASE ИмяВашейБД CHARACTER SET utf8;
```


Примечание: при создании базы данных она автоматически не выбирается. Выбирать ее нужно отдельно. Сделать БД текущей можно с помощью следующей команды:

```
mysql> USE ИмяВашейБД
```

Создавать базу нужно только однажды, но выбирать ее приходится в каждом сеансе работы с MySQL. Делать это можно с помощью команды USE, представленной выше, или из командной строки при запуске MySQL. Для этого достаточно лишь ввести ее имя после параметров соединения, которые нужно вводить в любом случае:

```
shell> mysql -h Хост -u Пользователь -p ИмяВашейБД  
Enter password: *****
```

Вариант2: создать БД из визуальной среды (например, в dbForge Studio):

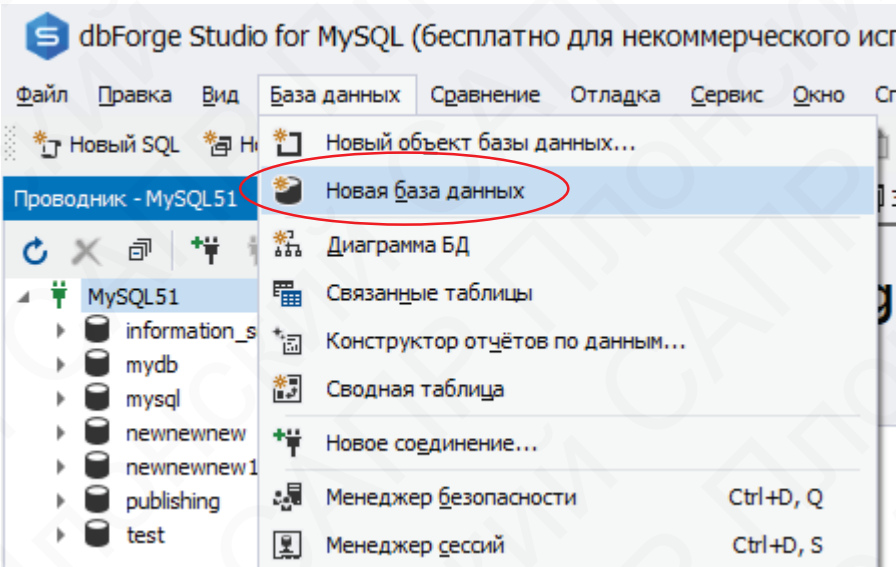


Рисунок 11. Добавление новой БД

Вариант 3: добавить следующую последовательность команд в полученный на предыдущем этапе SQL-скрипт:

```
CREATE DATABASE ИмяВашейБД CHARACTER SET utf8;  
USE ИмяВашейБД;
```

Примечание: найти файл со скриптом можно, посмотрев настройку «Location of SQL File» в Toad Data Modeler:

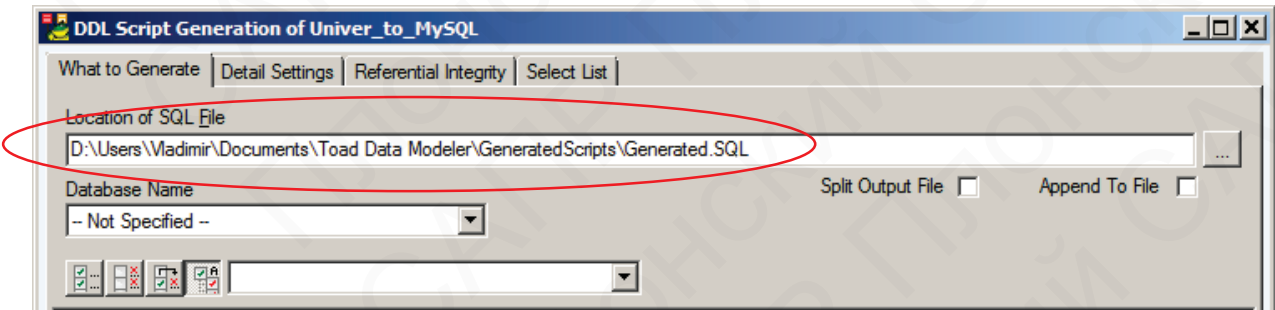


Рисунок 12. Расположение скрипта с командами DDL

```
Generated.SQL* x
1  /*
2  Created: 02.10.2018
3  Modified: 02.10.2018
4  Project: Базы данных
5  Model: Univer_to_MySQL
6  Company: Технологический институт
7  Author: Плонский В.Ю.
8  Database: MySQL 5.1
9  */
10
11  -- ЭТИ ДВЕ СТРОЧКИ ДОБАВЛЕНЫ ВРУЧНУЮ:
12  CREATE DATABASE Publishing_2018 CHARACTER SET utf8;
13  USE Publishing_2018;
14
15  -- Create tables section -----
16  -- Table Книга
17  CREATE TABLE `Книга`
18  (
19    `Код_книги` Int NOT NULL,
20    `Название` Varchar(50),
21    `Дата_выхода` Date,
22    `Ид_изд` Int
23  )
24  ;
```

Рисунок 13. Отредактированный DDL-скрипт

Загрузите dbForge Studio, на закладке «SQL разработка» выберите команду «Выполнить скрипт» (рисунок 14):

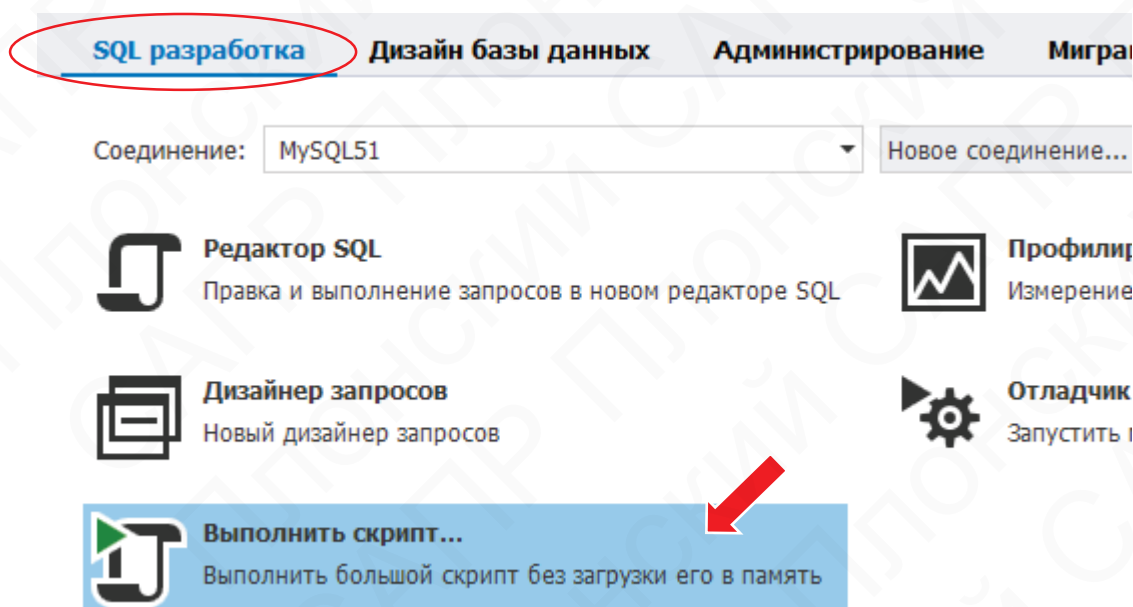


Рисунок 14. Запуск мастера выполнения скрипта

Укажите путь к файлу скрипта и нажмите кнопку «Выполнить» (рисунок 15):

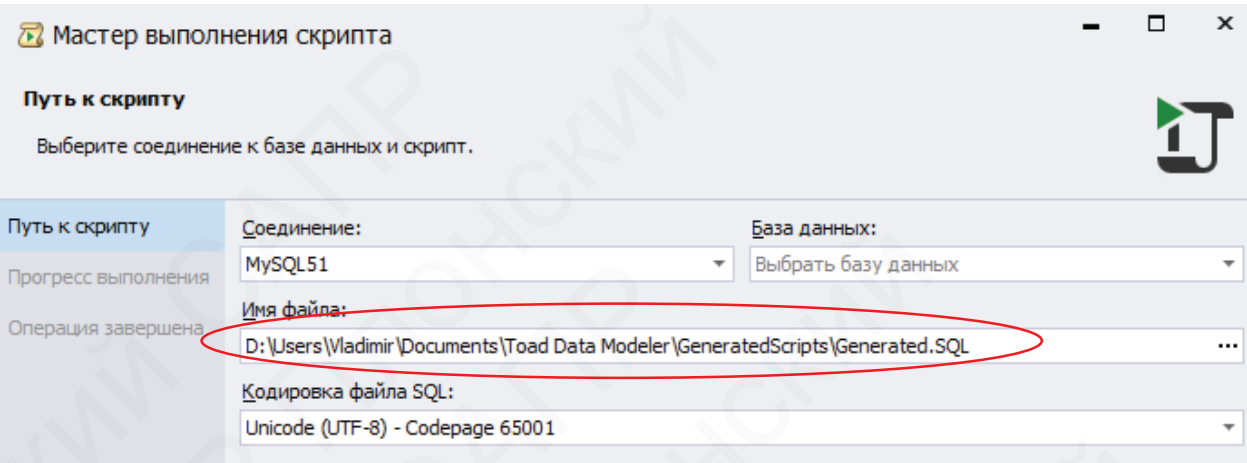


Рисунок 15. Настройка соединения и расположения файла командами SQL

Примечание: если есть ошибки, то будет предложено открыть скрипт для просмотра и редактирования. При этом скрипт будет загружен в «Редактор SQL». Скрипт можно исправить, сохранить и выполнить прямо в этом инструменте.

2.4. Проверка новой БД в списке баз MySQL.

Используя визуальную среду и консоль получите обновленный список БД:

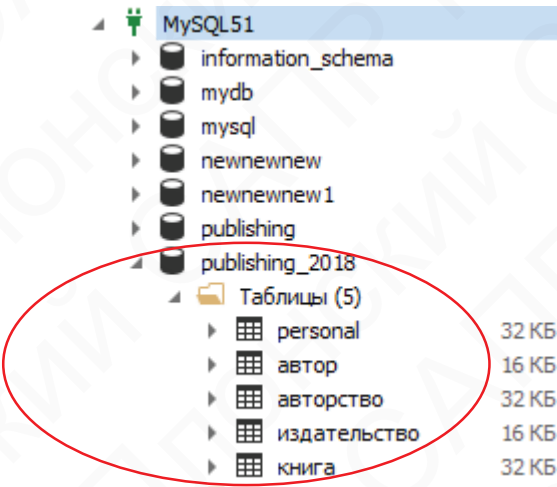


Рисунок 16. Список БД MySQL в dbForge Studio

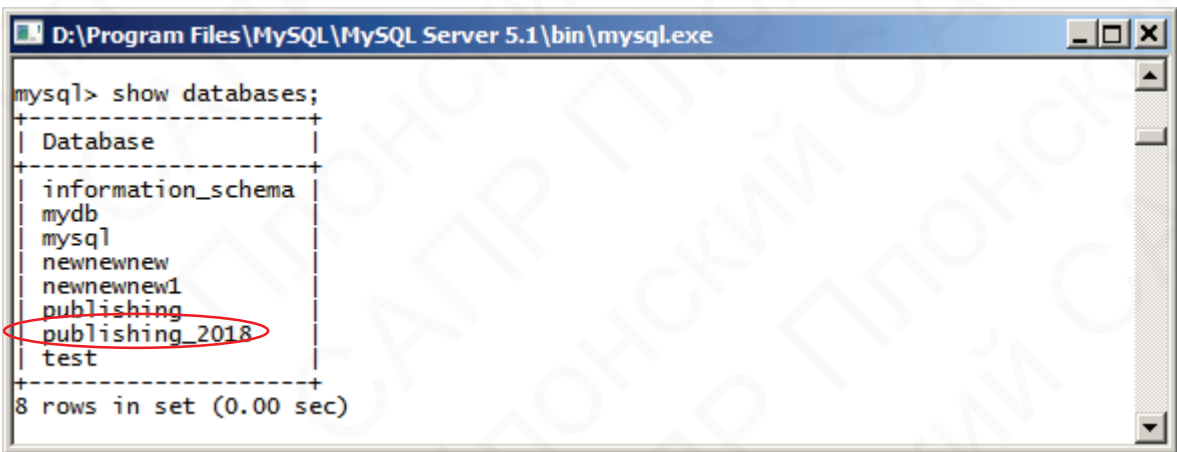


Рисунок 17. Список БД MySQL в клиенте командной строки

2.5. Создание диаграммы БД

Выберите в меню «База данных» команду «Диаграмма БД». В рабочую область перенесите таблицы текущей БД из окна Проводника. Сохраните диаграмму в файл (*.dbd).

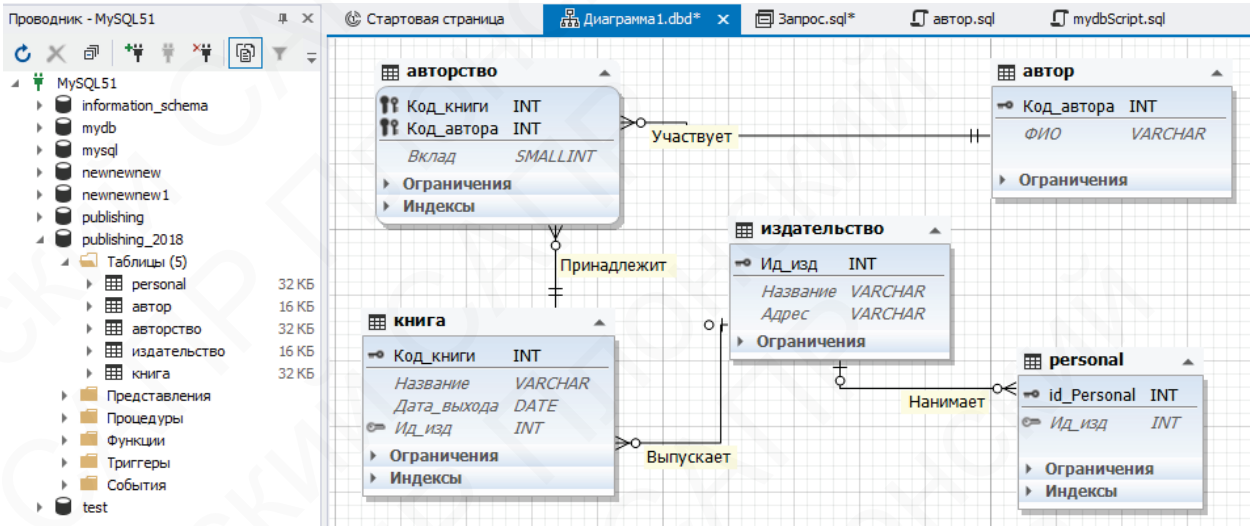


Рисунок 18. Диаграмма БД

3. Создание источника данных ODBC для БД MySQL.

Управление и настройка источников данных ODBC осуществляется с помощью специальной программы – «Администратор источников данных ODBC». Используя поиск («ODBC»), найдите и запустите эту программу. В окне «Администратор источников данных ODBC» на закладке «Пользовательский DSN» нажмите кнопку «Добавить»:

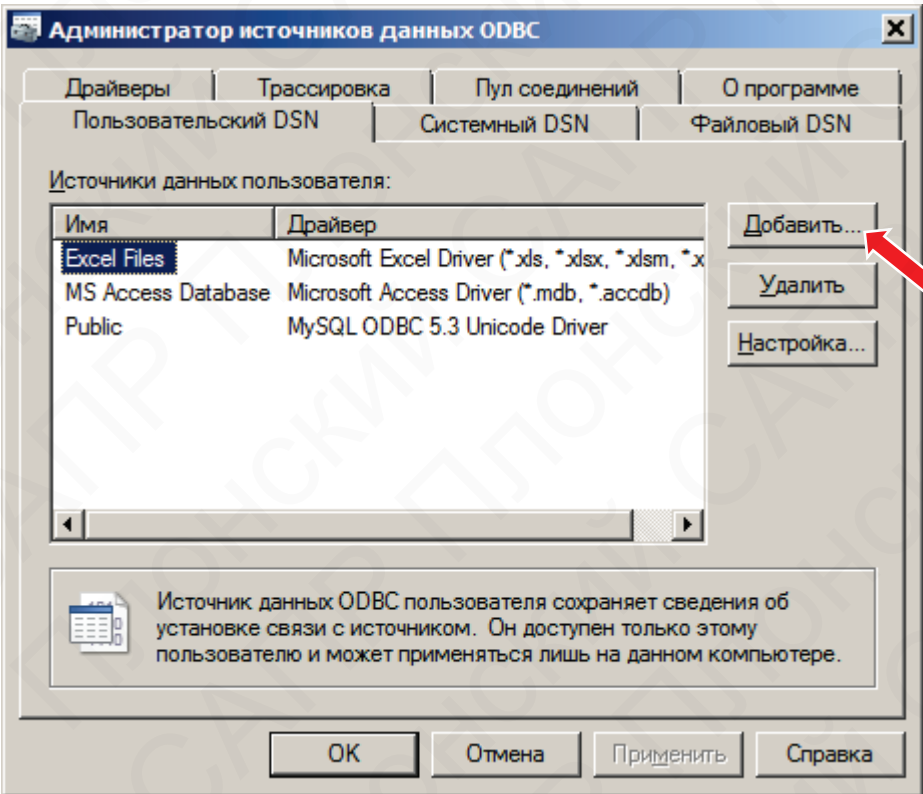


Рисунок 19. Список БД MySQL в клиенте командной строки

Примечание: ODBC представляет собой набор динамических библиотек DLL, которые обслуживают подключение и работу с конкретным типом базы данных. При запросе на подключение к заранее описанной БД "активируется" определенная DLL – драйвер этого типа БД. Обращение к БД происходит по имени источника данных ODBC (DSN – Data Source Name). DSN может быть пользовательским (находится в области видимости только одного пользователя, в сеансе которого был создан этот DSN) и системным (виден всем пользователям данного компьютера). Сведения о несистемных источниках данных хранятся в реестре:

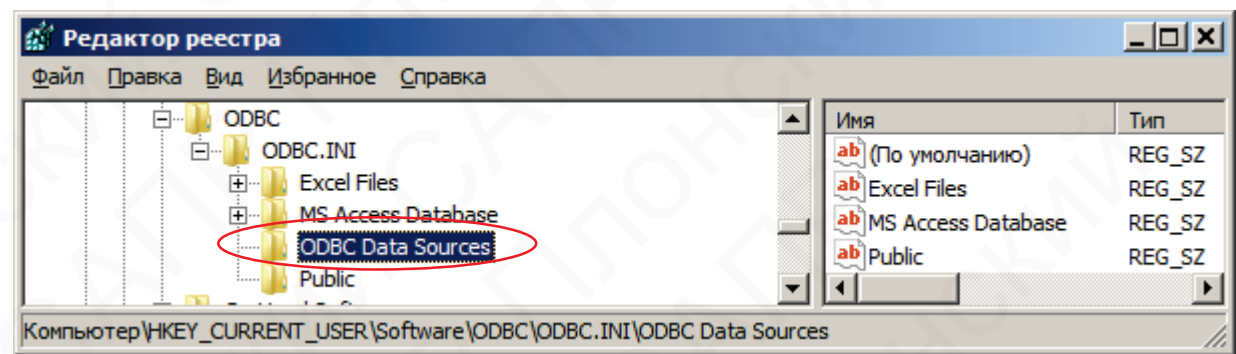


Рисунок 20. HKEY_CURRENT_USER\SOFTWARE\ODBC\ODBC.INI

Выберите драйвер MySQL ODBC:

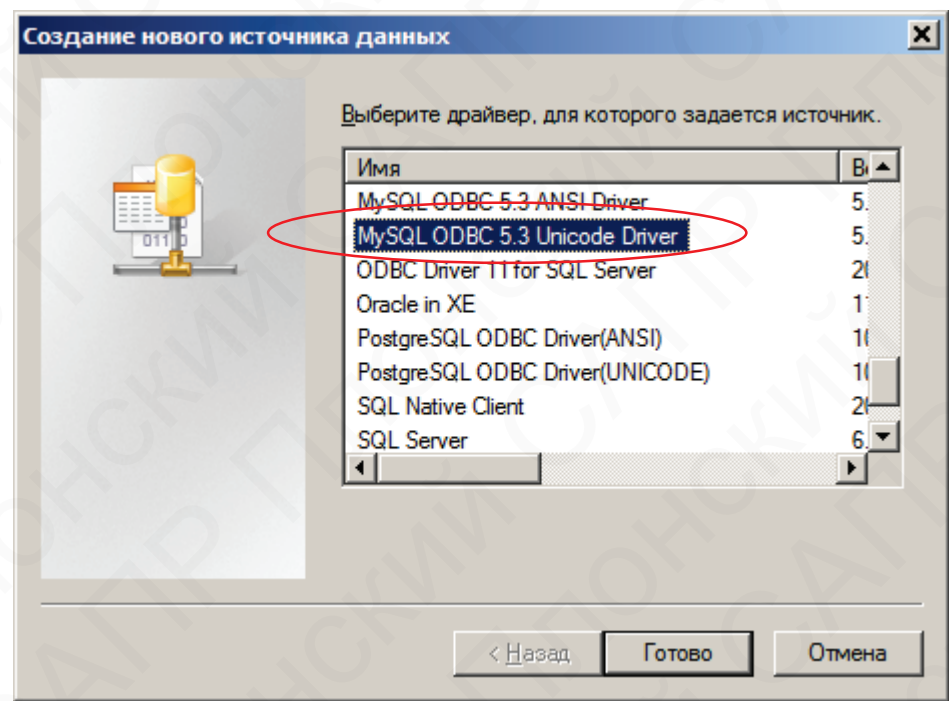


Рисунок 21. Выбор драйвера

Нажмите «Готово», заполните параметры соединения, укажите БД. Проведите тестирование нового соединения с источником данных:

Примечание: для работы в аудитории логин и пароль уточните у преподавателя (по умолчанию сервер – MYSQL, пользователь – root, пароль – root).

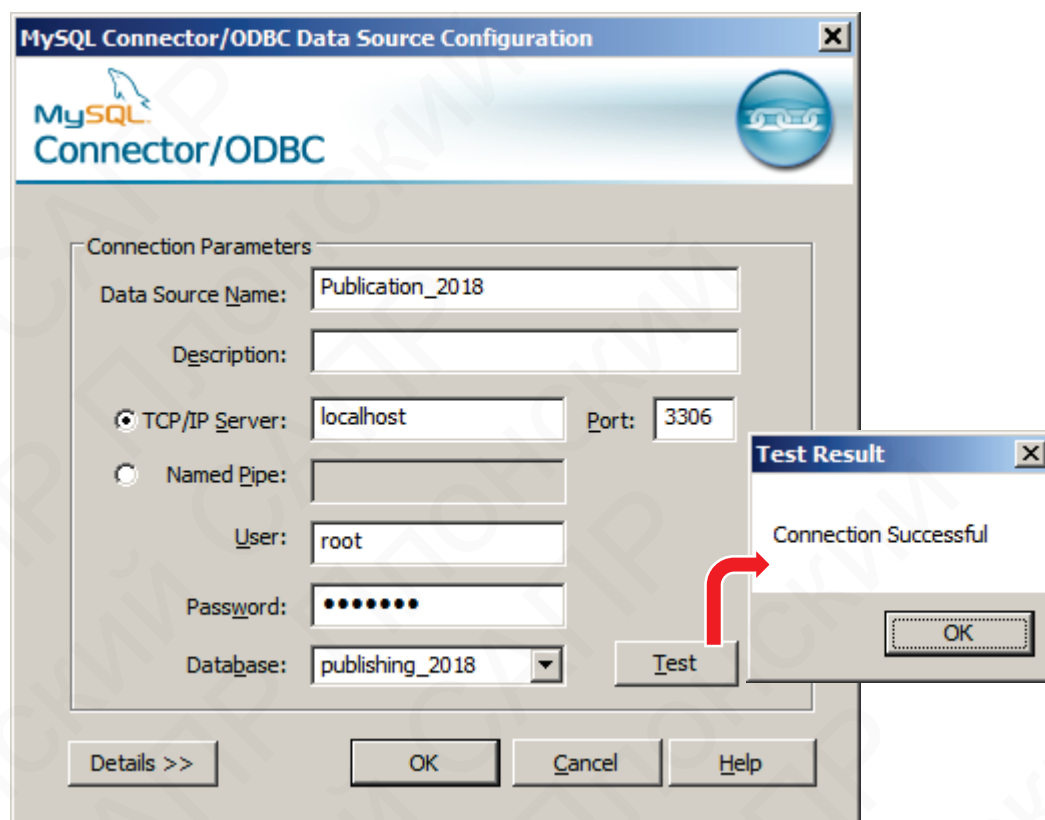


Рисунок 22. Конфигурирование источника данных

После подтверждения в списке «Источники данных пользователя» должна появиться новая строка.

4. Подключение к БД из внешнего приложения (LibreOffice Base).

4.1. Создание новой БД.

Запустите LibreOffice Base. Откроется окно «Мастер баз данных».

На первом шаге укажите вариант «Соединиться с существующей БД», выбрав значение «MySQL» (рисунок 23).

На втором шаге выберите подключения к БД (рисунок 24).

На третьем шаге выберите пользовательского источника данных (рисунок 25).

На четвертом шаге установите флаг «Требуется пароль», введите имя пользователя, выполните тестирование подключения (рисунок 26).

На последнем шаге оставьте значения настроек по умолчанию («Зарегистрировать БД» и «Открыть БД для редактирования»). Нажмите «Готово» и сохраните документ (*.odf).

Проверьте список таблиц в основном окне LibreOffice Base (рисунок 27).

Выберите в меню «Сервис» команду «Связи». Проверьте наличие всех таблиц, связей. Разместите таблицы для удобной работы со схемой БД (рисунок 28). Нажмите кнопку «Сохранить».

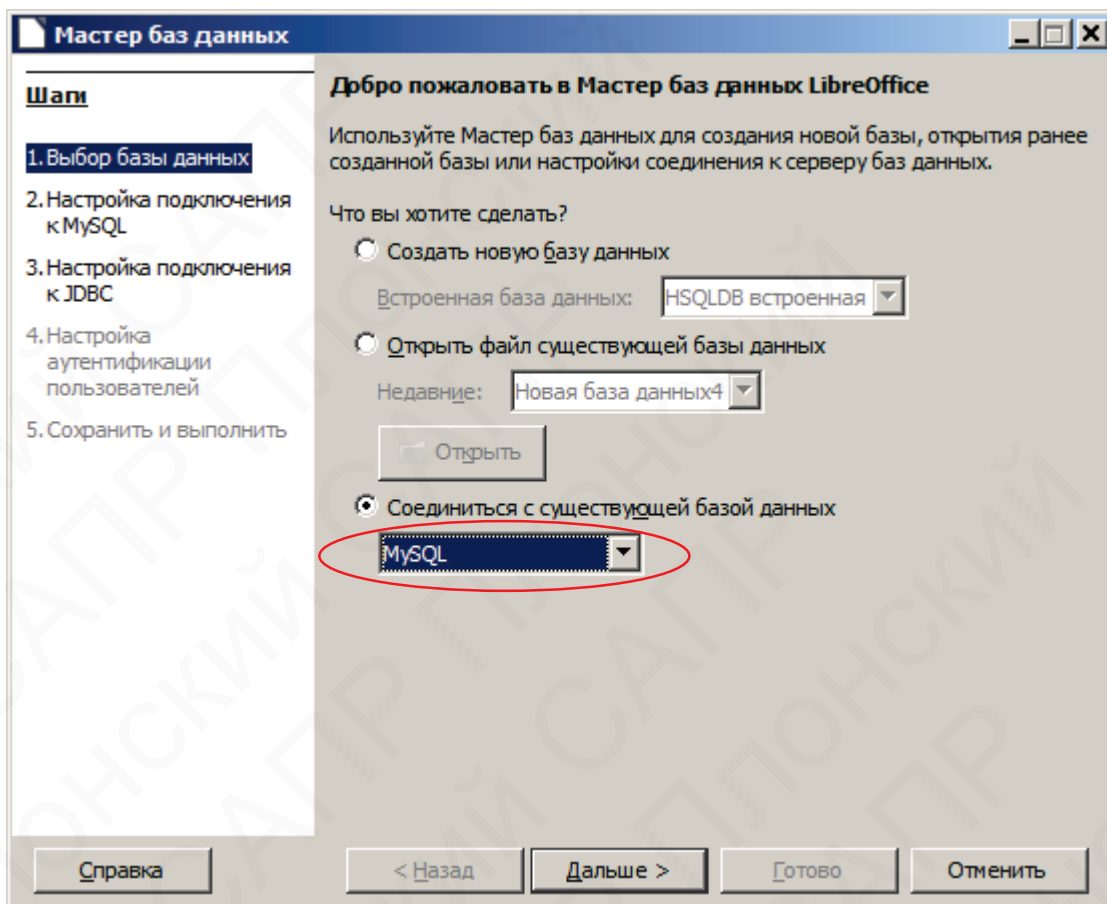


Рисунок 23. Выбор способа соединения с БД

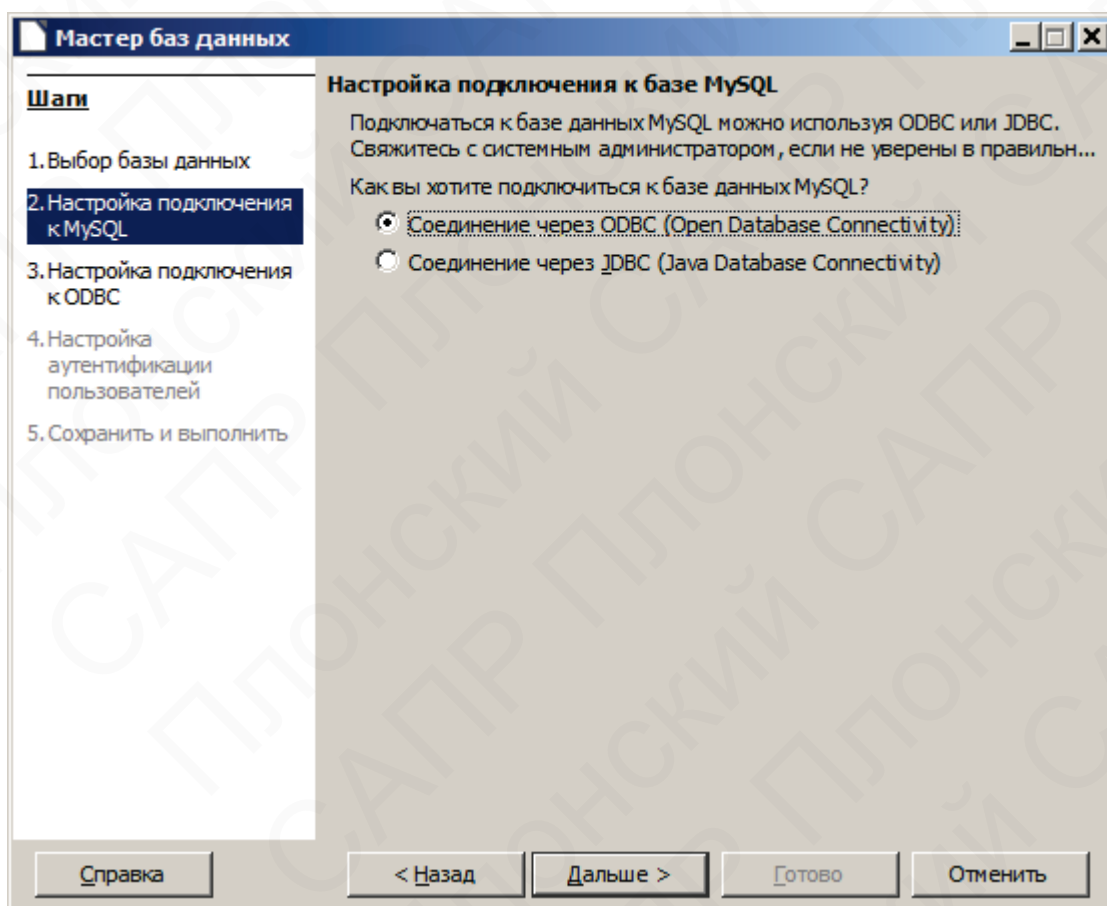


Рисунок 24. Выбор способа подключения

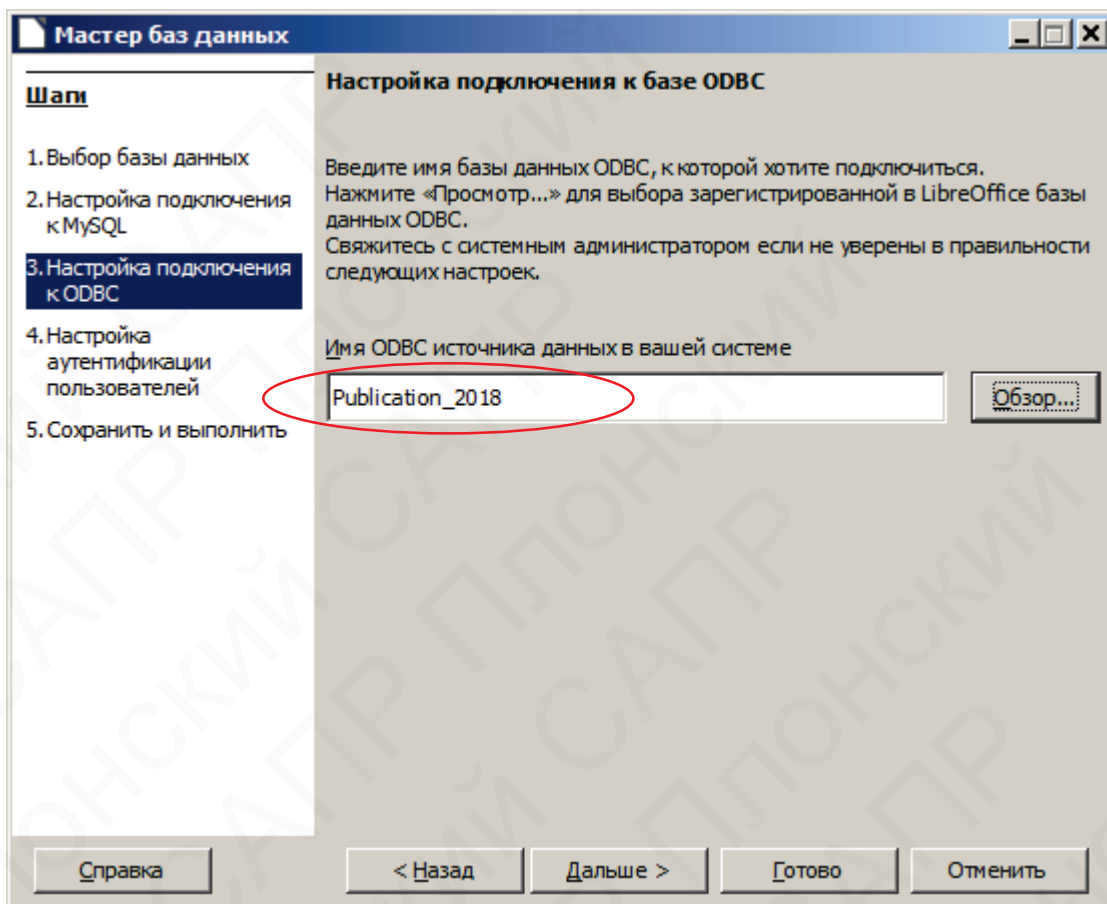


Рисунок 25. Выбор источника данных

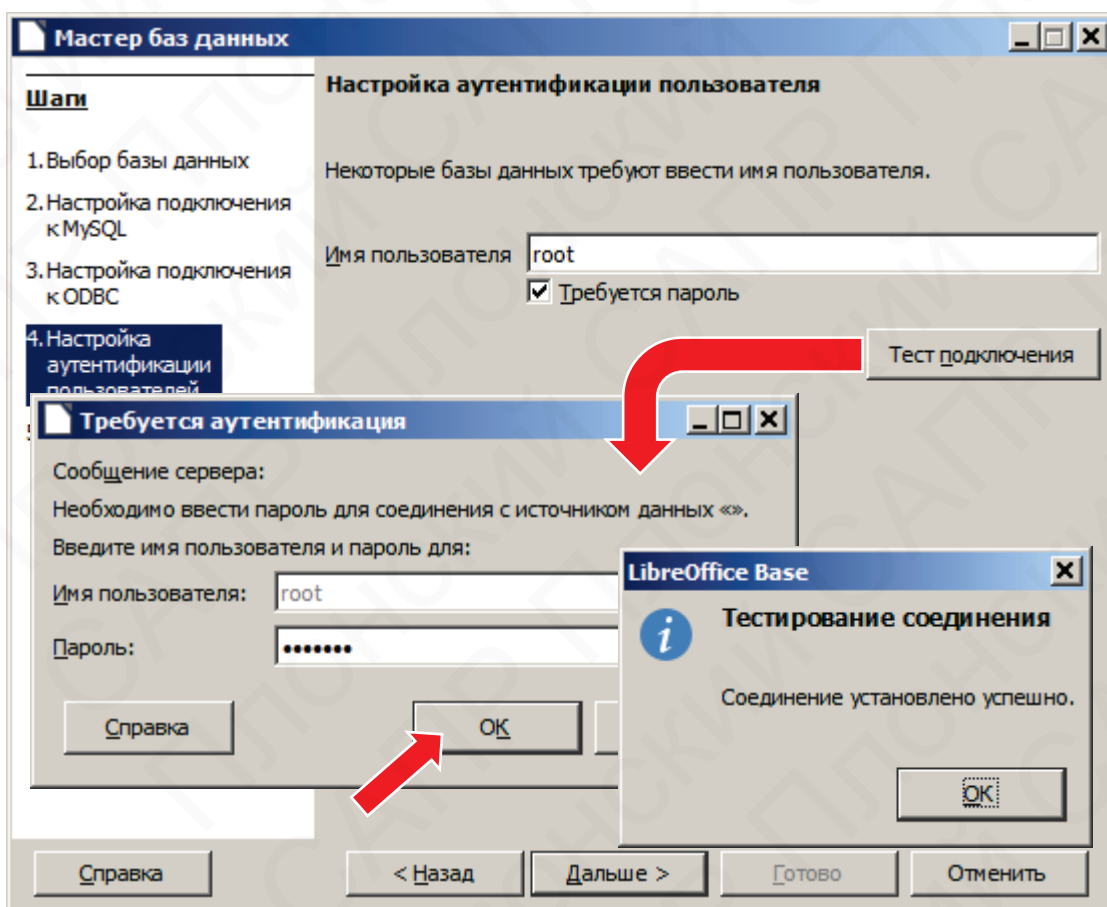


Рисунок 26. Настройка аутентификации

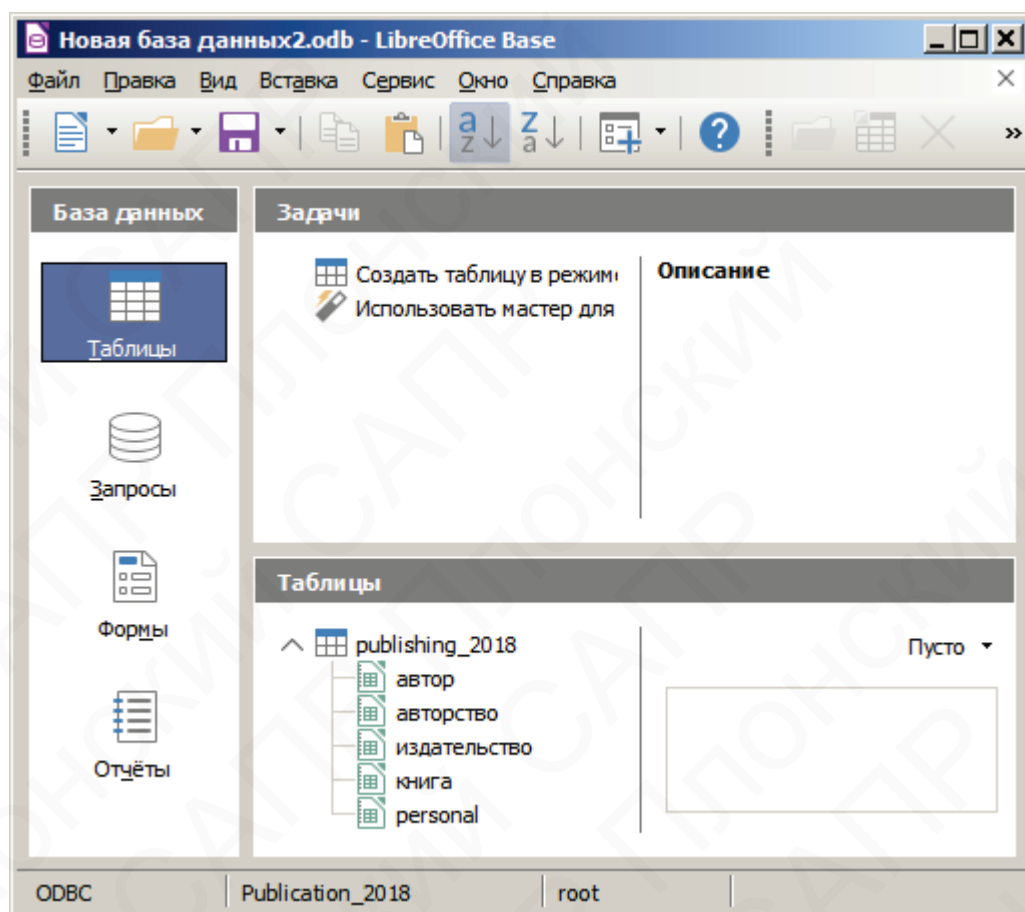


Рисунок 27. Таблицы БД

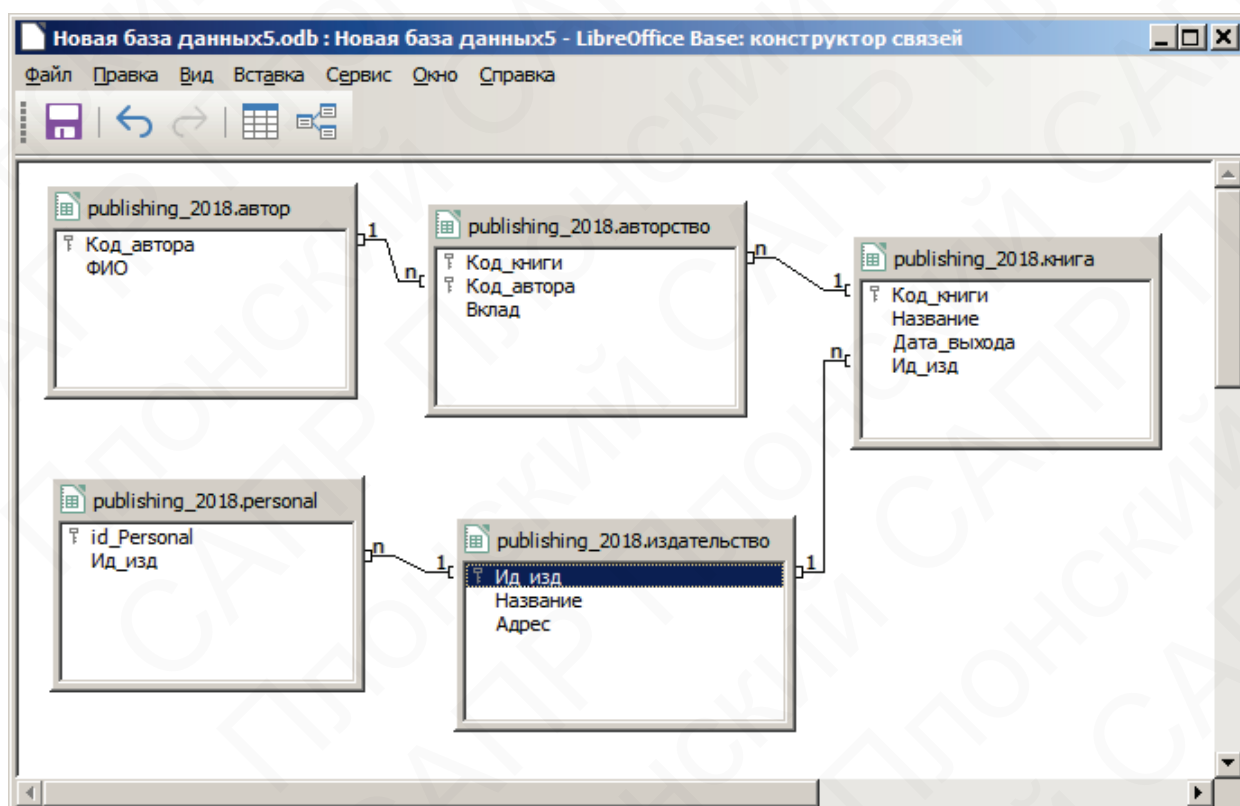
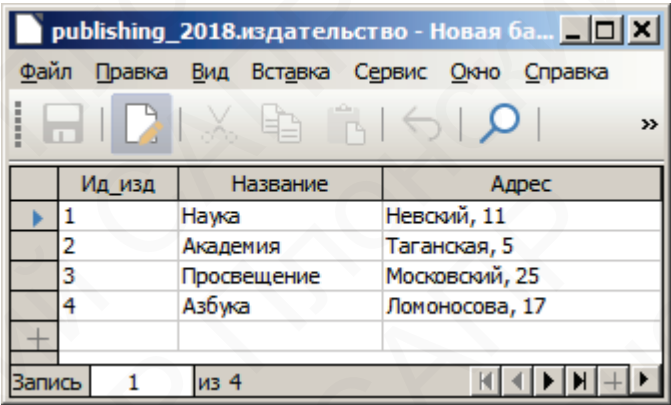


Рисунок 28. Схема БД

4.2. Ввод тестовых данных в БД

Введите тестовые данные во все таблицы. Проверьте заполнение таблиц в dbForge Studio (рисунок 33).

Примечание: вне зависимости от силы связи (сильные или слабые) первыми заполняются таблицы со стороны «один» на схеме данных. Затем вводятся данные на стороне «многие».

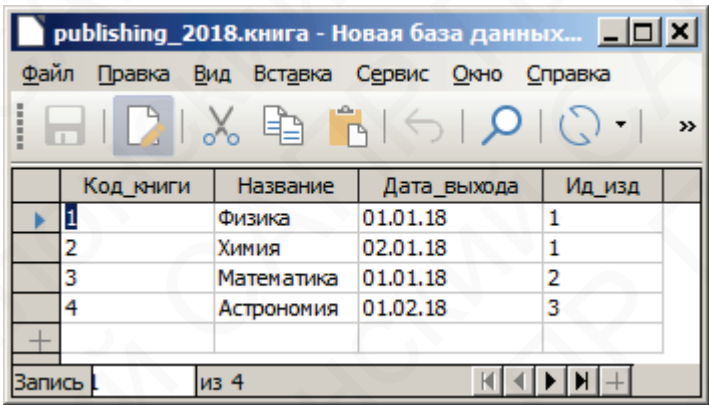


publishing_2018.издательство - Новая ба...

	Ид_изд	Название	Адрес
▶	1	Наука	Невский, 11
	2	Академия	Таганская, 5
	3	Просвещение	Московский, 25
	4	Азбука	Ломоносова, 17
+			

Запись 1 из 4

Рисунок 29. Таблица «ИЗДАТЕЛЬСТВО»

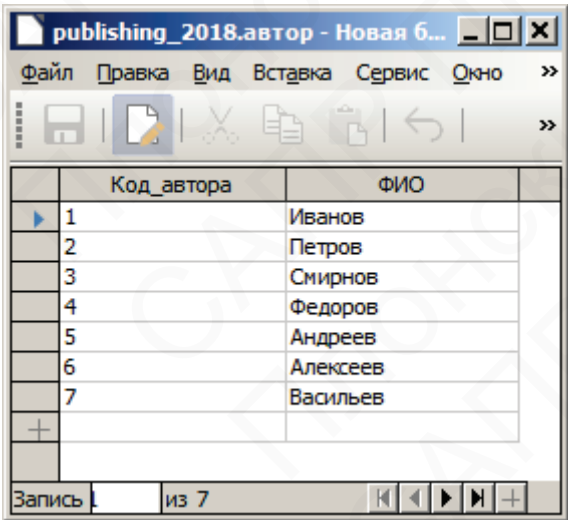


publishing_2018.книга - Новая база данных...

	Код_книги	Название	Дата_выхода	Ид_изд
▶	1	Физика	01.01.18	1
	2	Химия	02.01.18	1
	3	Математика	01.01.18	2
	4	Астрономия	01.02.18	3
+				

Запись 1 из 4

Рисунок 30. Таблица «Книга»

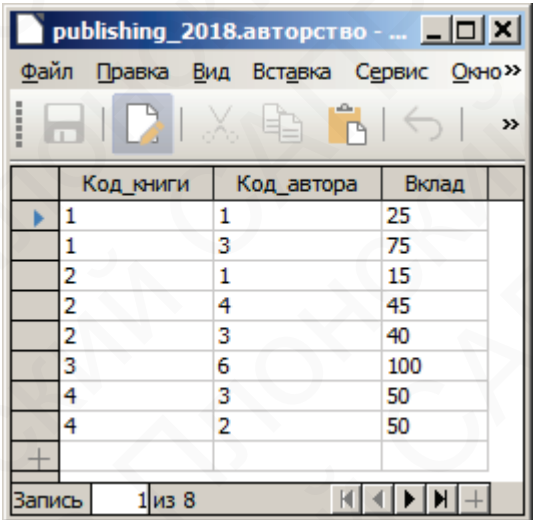


publishing_2018.автор - Новая б...

	Код_автора	ФИО
▶	1	Иванов
	2	Петров
	3	Смирнов
	4	Федоров
	5	Андреев
	6	Алексеев
	7	Васильев
+		

Запись 1 из 7

Рисунок 31. Таблица «АВТОР»



publishing_2018.авторство - ...

	Код_книги	Код_автора	Вклад
▶	1	1	25
	1	3	75
	2	1	15
	2	4	45
	2	3	40
	3	6	100
	4	3	50
	4	2	50
+			

Запись 1 из 8

Рисунок 32. Таблица «АВТОРСТВО»

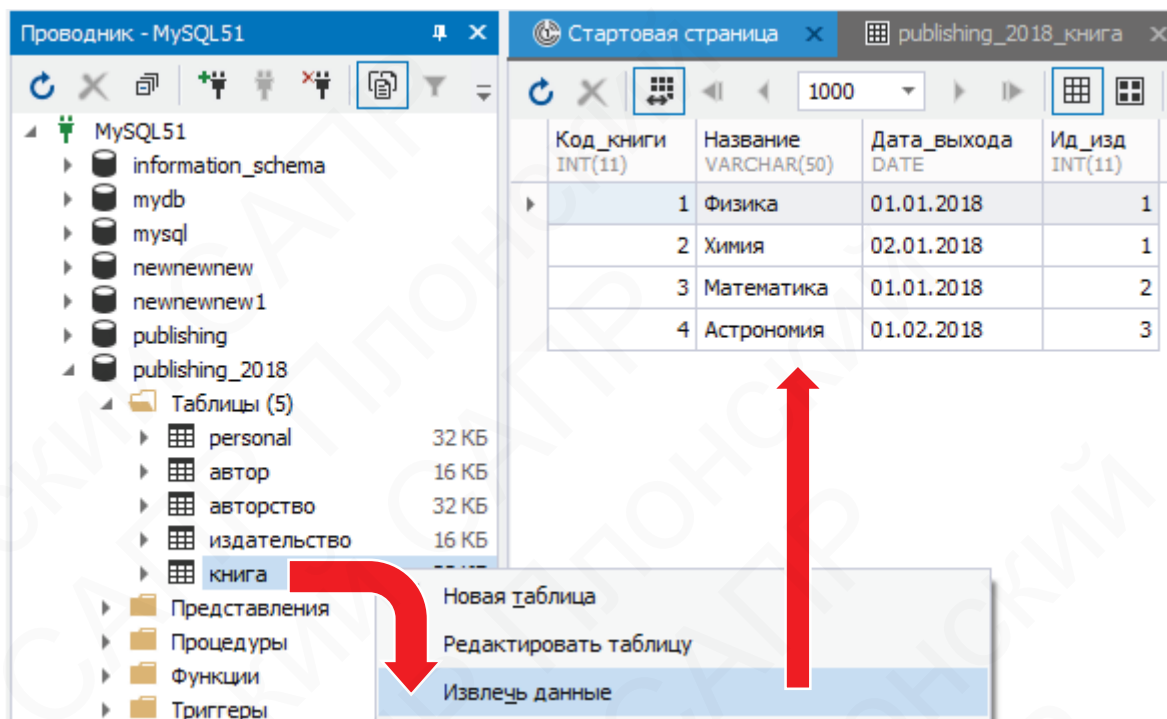


Рисунок 33. Таблица «Книга» в dbForge Studio

5. Резервное копирование и восстановление.

5.1. Расположение файлов БД

Узнайте дефолтный каталог расположения файлов БД вашего MySQL-сервера. Для этого загрузите клиент командной строки MySQL и выполните запрос:

```
mysql> SELECT @@datadir;
```

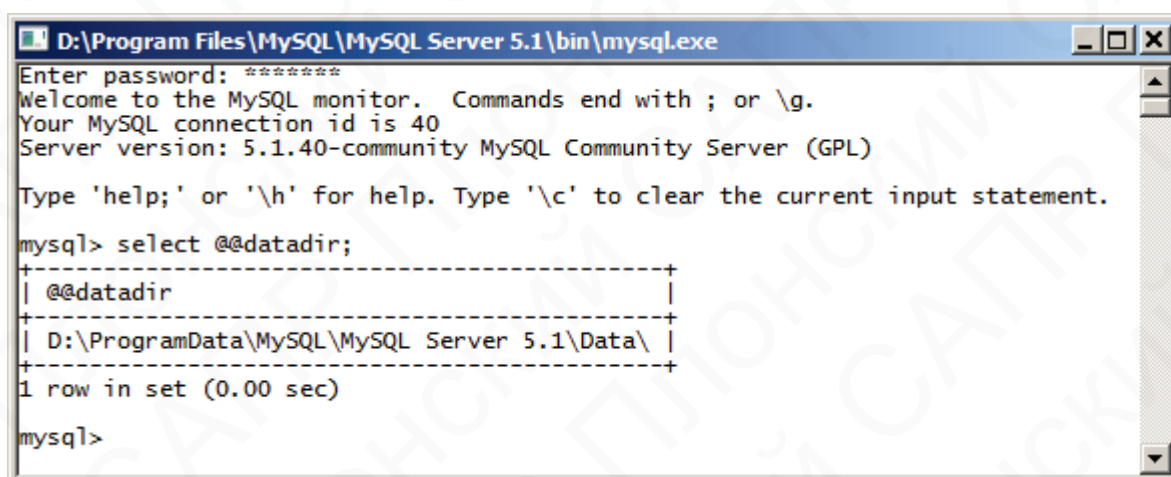


Рисунок 34. Местоположение каталога данных MySQL

Примечание: расположение данных также можно узнать в конфигурационных файлах установки сервера MySQL, посмотрев значение `datadir`. Для Windows это файл – `my.ini`; для Linux – `my.cnf`.

По этому пути будут находиться другие каталоги с именами, совпадающими с названиями баз данных. Каждая таблица имеет здесь отдельный файл.

5.2. Создание резервной копии в MYSQLDUMP

Для выполнения резервного копирования можно использовать возможности «MYSQLDUMP». Данная утилита позволяет получить дамп («моментальный снимок») содержимого базы данных или совокупности баз для создания резервной копии или пересылки данных на другой сервер баз данных (не обязательно MySQL-сервер). Изучите синтаксис этой утилиты, выполнив команду в интерпретаторе командной строки:

```
shell> MYSQLDUMP
```

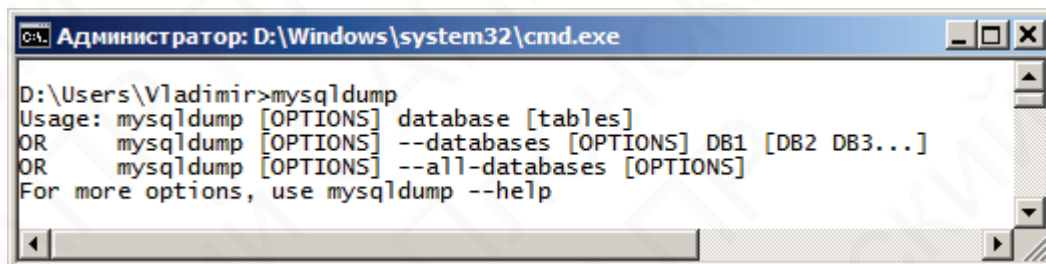


Рисунок 35. Синтаксис утилиты «mysqldump»

Примечание: возможно создание дампа только структуры (без данных), а также получение дампа только одной или нескольких таблиц.

Выполните резервное копирование вашей БД, перенаправив поток в SQL-файл:

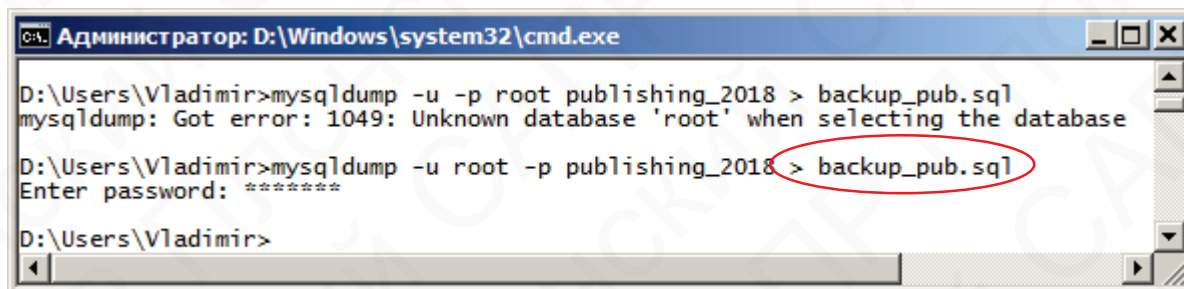


Рисунок 36. Резервное копирование

Дамп будет содержать набор команд SQL для создания и/или заполнения таблиц (рисунок 37).

Примечание: обратите внимание на наличие DDL-инструкций DROP TABLE и CREATE TABLE в дампе. Это позволяет перенести БД на новый сервер.

Загрузите клиент командной строки MySQL. Сделайте базу данных активной (используя команду «USE»). Посмотрите список таблиц (рисунок 37), выполнив команду:

```
mysql> SHOW TABLES;
```

Примечание: в списке таблиц может наблюдаться проблема с кодировкой русских символов (например, когда имена таблиц были набраны русскими символами). Команда «SET NAMES» (рисунок 37) устанавливает кодировку соединения (кодировку, в которой клиент обменивается данными с сервером):

```
mysql> SET NAMES cp866;
```

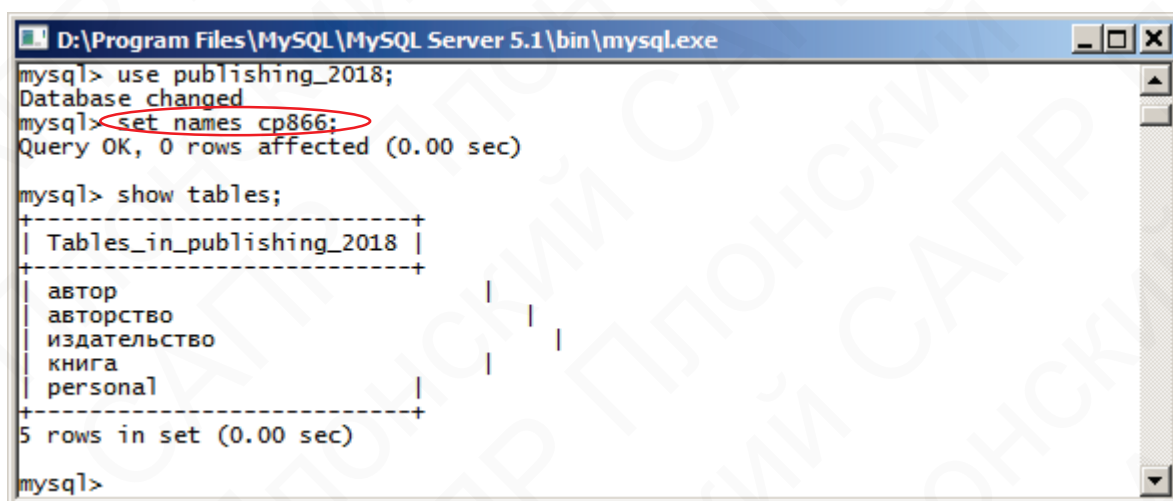
```
--
-- Table structure for table `автор`
--
DROP TABLE IF EXISTS `автор`;
/*!40101 SET @saved_cs_client = @@character_set_client */;
/*!40101 SET character_set_client = utf8 */;
CREATE TABLE `автор` (
  `Код_автора` int(11) NOT NULL,
  `ФИО` varchar(50) DEFAULT NULL,
  PRIMARY KEY (`Код_автора`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8;
/*!40101 SET character_set_client = @saved_cs_client */;

--
-- Dumping data for table `автор`
--

LOCK TABLES `автор` WRITE;
/*!40000 ALTER TABLE `автор` DISABLE KEYS */;
INSERT INTO `автор` VALUES (1,'Иванов'),(2,'Петров'),(3,'Смирнов');
/*!40000 ALTER TABLE `автор` ENABLE KEYS */;
UNLOCK TABLES;

--
-- Table structure for table `авторство`
```

Рисунок 37. Фрагмент дампа



```
D:\Program Files\MySQL\MySQL Server 5.1\bin\mysql.exe
mysql> use publishing_2018;
Database changed
mysql> set names cp866;
Query OK, 0 rows affected (0.00 sec)

mysql> show tables;
+-----+
| Tables_in_publishing_2018 |
+-----+
| автор                     |
| авторство                 |
| издательство              |
| книга                     |
| personal                  |
+-----+
5 rows in set (0.00 sec)

mysql>
```

Рисунок 38. Вывод списка таблиц

Выполните запрос на выборку к одной из таблиц. Измените или удалите строки таблицы, используя соответственно операторы UPDATE или DELETE.

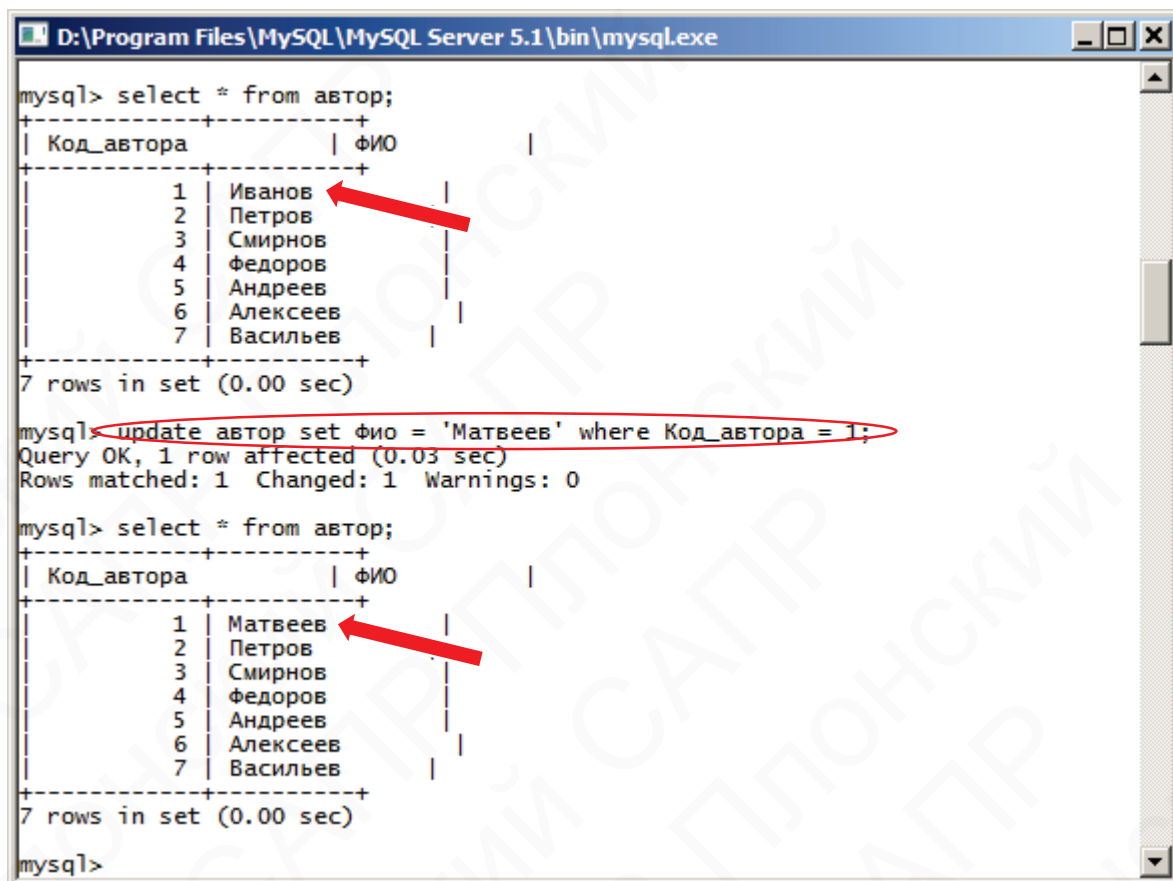


Рисунок 39. Данные таблицы до изменения

Примечание: если не распознаются русские символы в тексте запроса, проверьте параметры кодировок:

```
mysql> SHOW VARIABLES LIKE 'character%';
```

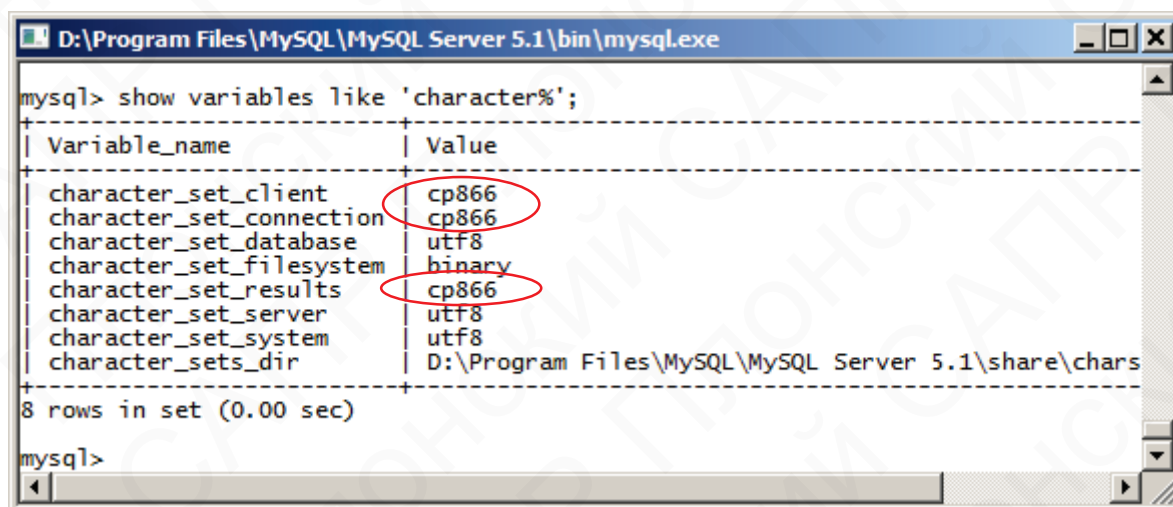


Рисунок 40. Текущие параметры кодировок MySQL

Параметры кодировки, используемые при коннекте клиента к серверу:
character_set_client – кодировка, в которой посылается запрос от клиента;
character_set_connection – кодировка, используемая для конвертации пришедшего запроса;
character_set_results – кодировка, в которую сервер должен перевести результат перед его отправкой клиенту.

Эти три параметры можно установить одной командой:

```
mysql> SET NAMES cp866;
```

Можно настроить отдельно каждый параметр:

```
mysql> SET character_set_client = cp866;
```

Обратите внимание, что база данных, таблицы, колонки таблиц могут использовать свою кодировку. Для просмотра кодировок этих объектов модно использовать словарь данных, представленный базой данных INFORMATION_SCHEMA, хранящей метаданные о всех БД сервера MySQL:

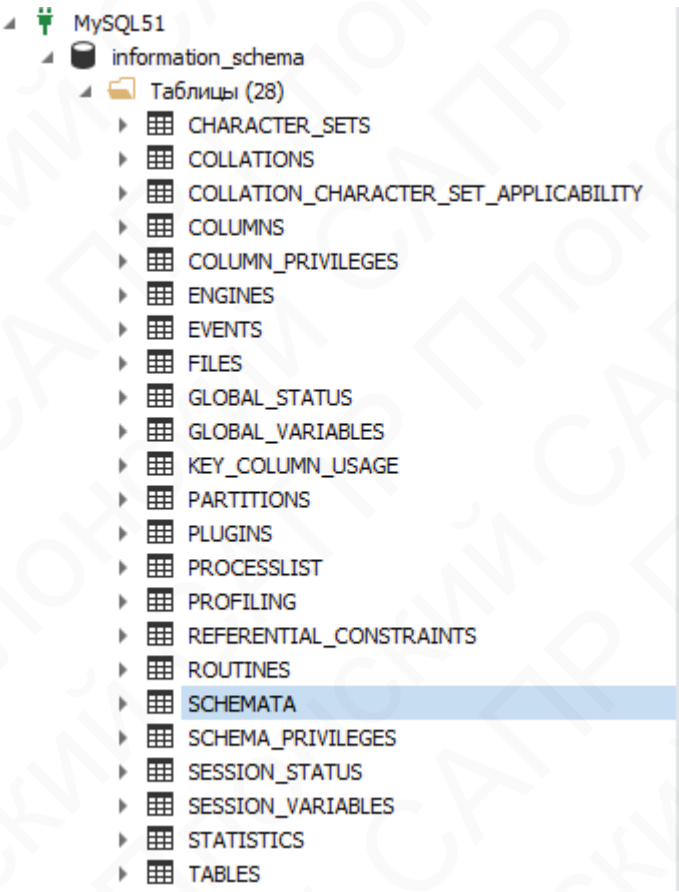


Рисунок 41. Некоторые таблицы словаря данных

CATALOG_NAME VARCHAR(512)	SCHEMA_NAME VARCHAR(64)	DEFAULT_CHARACTER_SET_NAME VARCHAR(32)	DEFAULT_COLLATION_NAME VARCHAR(32)	SQL_PATH VARCHAR(512)
(null)	information_schema	utf8	utf8_general_ci	(null)
(null)	chemical	utf8	utf8_general_ci	(null)
(null)	mydb	utf8	utf8_general_ci	(null)
(null)	mysql	utf8	utf8_general_ci	(null)
(null)	newnewnew	utf8	utf8_general_ci	(null)
(null)	newnewnew1	utf8	utf8_general_ci	(null)
(null)	publishing	utf8	utf8_general_ci	(null)
(null)	publishing_2018	utf8	utf8_general_ci	(null)
(null)	test	utf8	utf8_general_ci	(null)

Рисунок 42. Таблица SCHEMA базы данных INFORMATION_SCHEMA

Для просмотра кодировки конкретной БД также можно выполнить запрос:

```
mysql> SELECT default_character_set_name FROM information_schema WHERE  
schema_name = ИмяВашейБД;
```

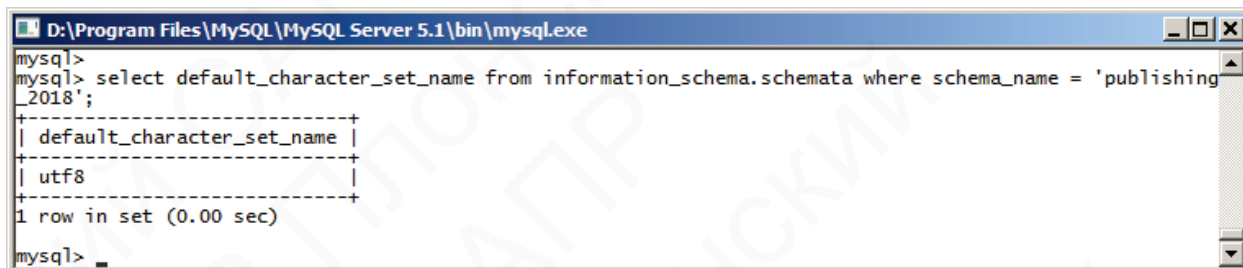


Рисунок 43. Кодировка базы данных

A screenshot of the MySQL command prompt showing the output of the 'SHOW TABLES' command. The output is a list of tables and their data types, with 'TABLE_COLLATION' highlighted. The text is as follows:
TABLES
+-----+
| Столбцы (21) |
+-----+
TABLE_CATALOG	VARCHAR(512)
TABLE_SCHEMA	VARCHAR(64), not null
TABLE_NAME	VARCHAR(64), not null
TABLE_TYPE	VARCHAR(64), not null
ENGINE	VARCHAR(64)
VERSION	BIGINT(21)
ROW_FORMAT	VARCHAR(10)
TABLE_ROWS	BIGINT(21)
AVG_ROW_LENGTH	BIGINT(21)
DATA_LENGTH	BIGINT(21)
MAX_DATA_LENGTH	BIGINT(21)
INDEX_LENGTH	BIGINT(21)
DATA_FREE	BIGINT(21)
AUTO_INCREMENT	BIGINT(21)
CREATE_TIME	DATETIME
UPDATE_TIME	DATETIME
CHECK_TIME	DATETIME
TABLE_COLLATION	VARCHAR(32)
CHECKSUM	BIGINT(21)
CREATE_OPTIONS	VARCHAR(255)
TABLE_COMMENT	VARCHAR(80), not null
+-----+

Рисунок 44. Метаданные – таблицы баз данных сервера MySQL

Для изменения кодировки БД (например, в UTF-8) можно выполнить запрос следующего вида (не преобразует существующие таблицы, устанавливает только значения по умолчанию для вновь создаваемых таблиц):

```
mysql> ALTER DATABASE ИмяБД CHARACTER SET utf8 COLLATE utf8_general_ci;
```

Для изменения кодировки таблицы можно использовать запрос:

```
mysql> ALTER TABLE ИмяТаблицы CONVERT TO CHARACTER SET utf8 COLLATE  
utf8_general_ci;
```

Columns	
Столбцы (19)	
TABLE_CATALOG	VARCHAR(512)
TABLE_SCHEMA	VARCHAR(64), not null
TABLE_NAME	VARCHAR(64), not null
COLUMN_NAME	VARCHAR(64), not null
ORDINAL_POSITION	BIGINT(21), not null
COLUMN_DEFAULT	LONGTEXT
IS_NULLABLE	VARCHAR(3), not null
DATA_TYPE	VARCHAR(64), not null
CHARACTER_MAXIMUM_LENGTH	BIGINT(21)
CHARACTER_OCTET_LENGTH	BIGINT(21)
NUMERIC_PRECISION	BIGINT(21)
NUMERIC_SCALE	BIGINT(21)
CHARACTER_SET_NAME	VARCHAR(32)
COLLATION_NAME	VARCHAR(32)
COLUMN_TYPE	LONGTEXT, not null
COLUMN_KEY	VARCHAR(3), not null
EXTRA	VARCHAR(27), not null
PRIVILEGES	VARCHAR(80), not null
COLUMN_COMMENT	VARCHAR(255), not null

Рисунок 45. Метаданные – столбцы таблиц баз данных сервера MySQL

Для изменения кодировки столбца таблицы можно применить запрос:

```
mysql> ALTER TABLE ИмяТаблицы MODIFY ИмяСтолбца ТипДанных
CHARACTER SET utf8 COLLATE utf8_general_ci;
```

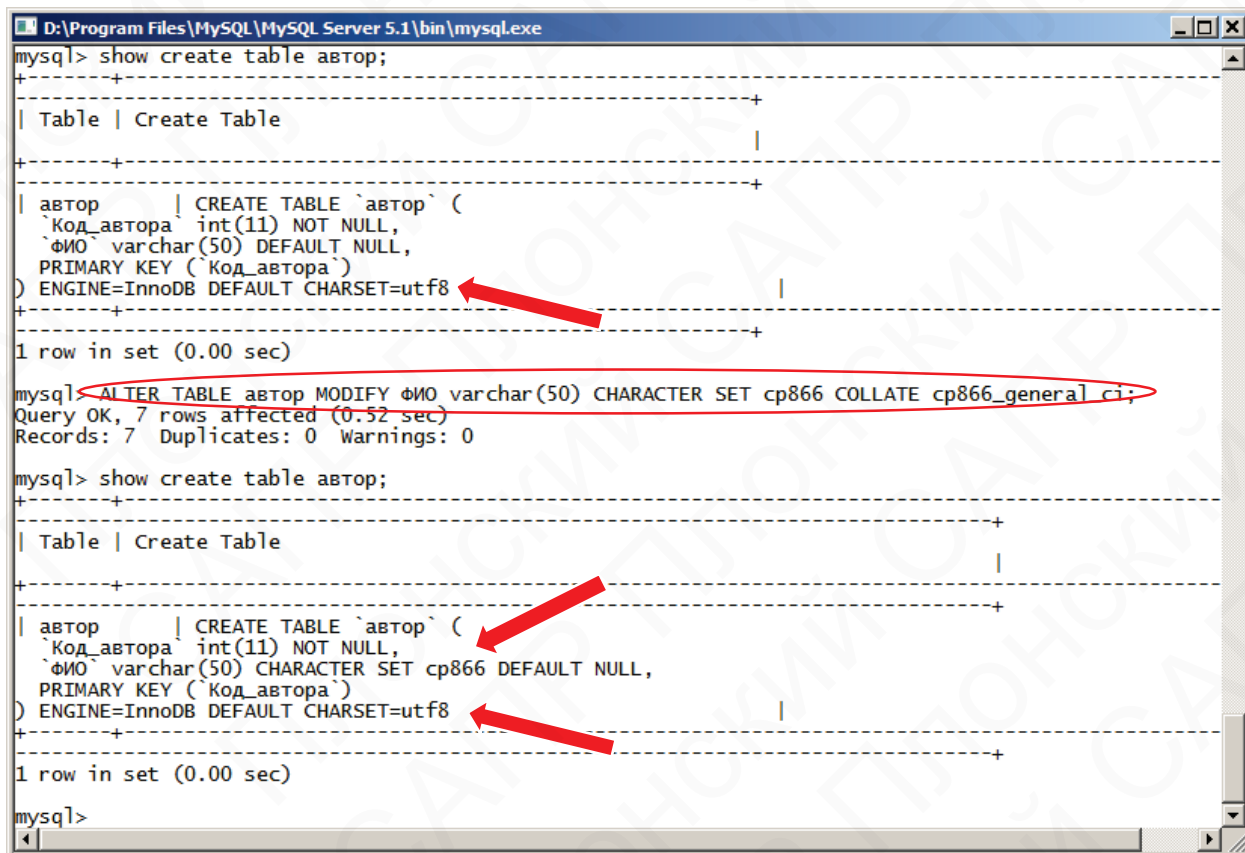


Рисунок 46. Метаданные – разная кодировка для столбца и таблицы

Не забудьте восстановить исходную кодировку столбца для продолжения работы.

Восстановление БД из SQL-дампа выполняется командой «SOURCE»:

mysql> SOURCE ИмяФайлаСоСценариемSQL;

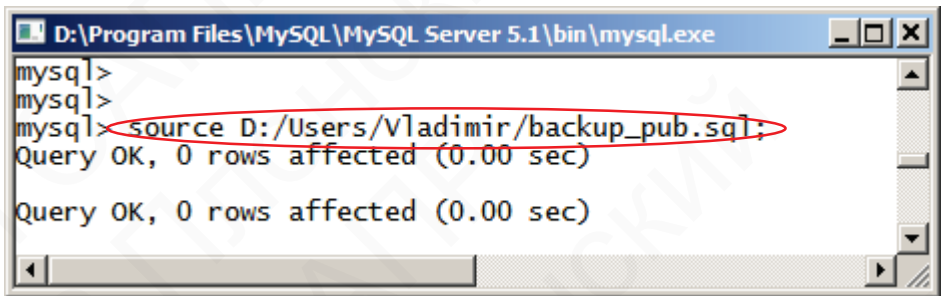


Рисунок 47. Восстановление БД

После восстановления из дампа содержимое таблицы должно соответствовать моменту создания резервной копии (сравните данные на рисунках 38 и 47):

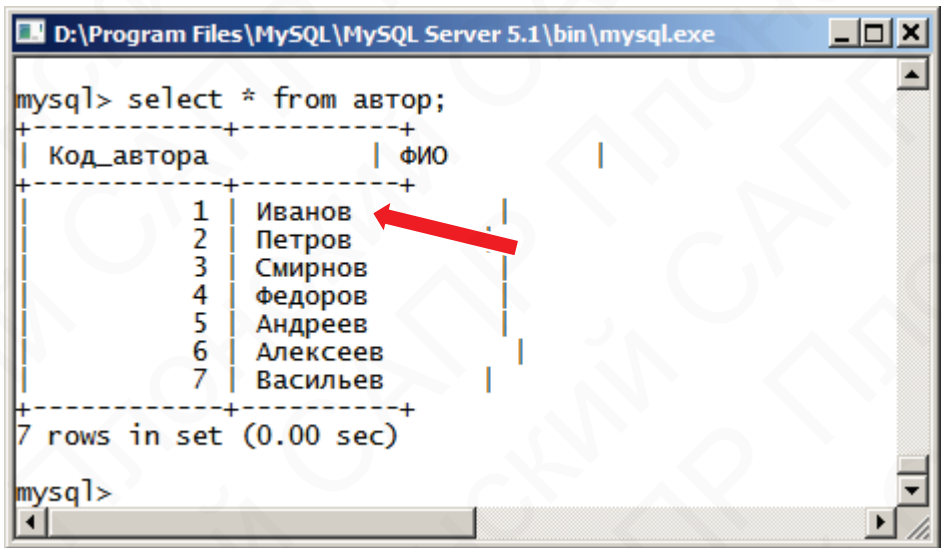


Рисунок 48. Данные таблицы после восстановления

5.3. Создание резервной копии в dbForge Studio

Перейдите на вкладку «Администрирование» и выполните команду «Создать резервную копию БД...» (рисунок 49).

Примечание: такая же команда доступна на вкладке «Миграция данных».

На первом шаге мастера резервирования укажите требуемую БД, расположение резервной копии (рисунок 50).

Примечание: обратите внимание, что на первом шаге можно создать архив. Такая же возможность существует и у программы «MYSQLDUMP».

На втором шаге (вкладка «Содержимое») уточните объекты для резервного копирования. Дополнительные настройки устанавливаются (при необходимости) на третьем шаге (вкладка «Опции»).

Вызовите команду «Выполнить», установите флажок «Открыть скрипт» и завершите работу с мастером резервирования. Разберите содержимое SQL-скрипта (рисунок 51).

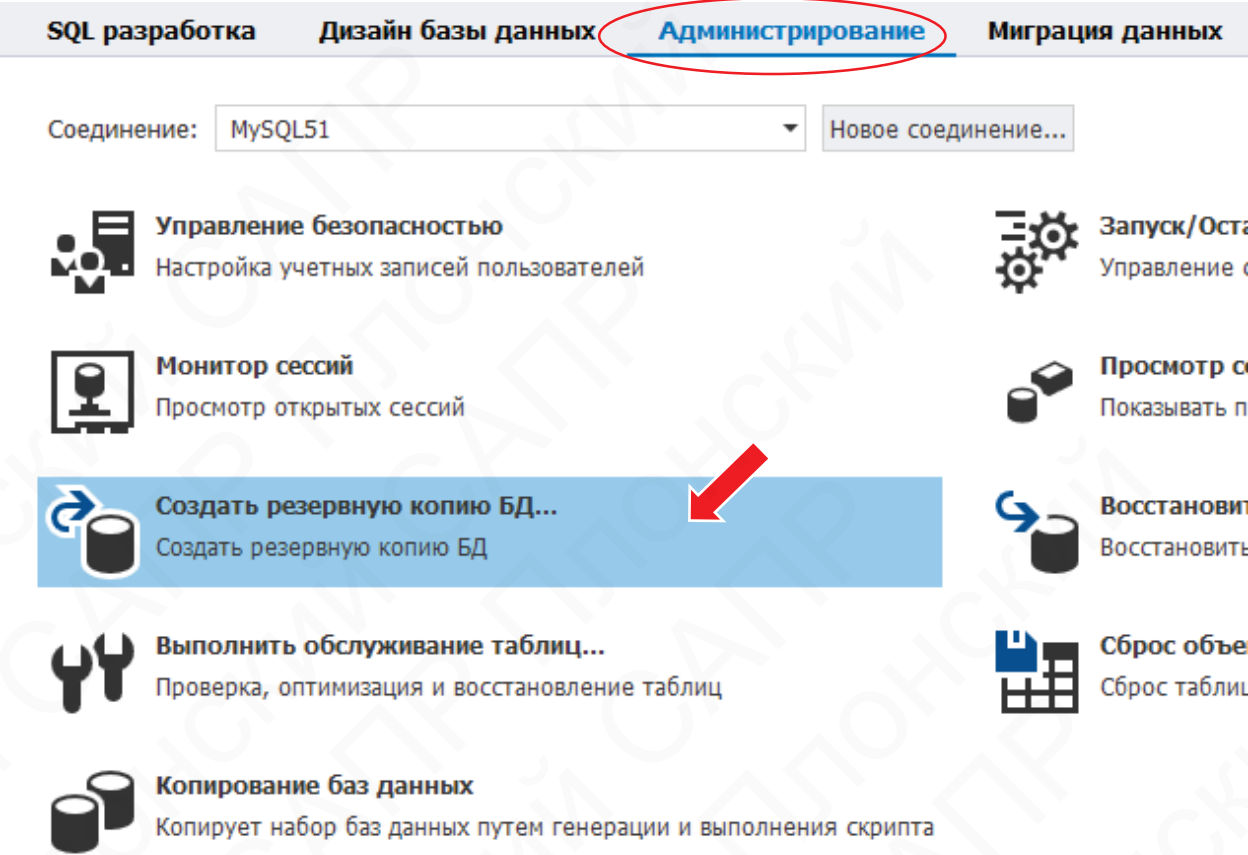


Рисунок 49. Команда создания резервной копии БД

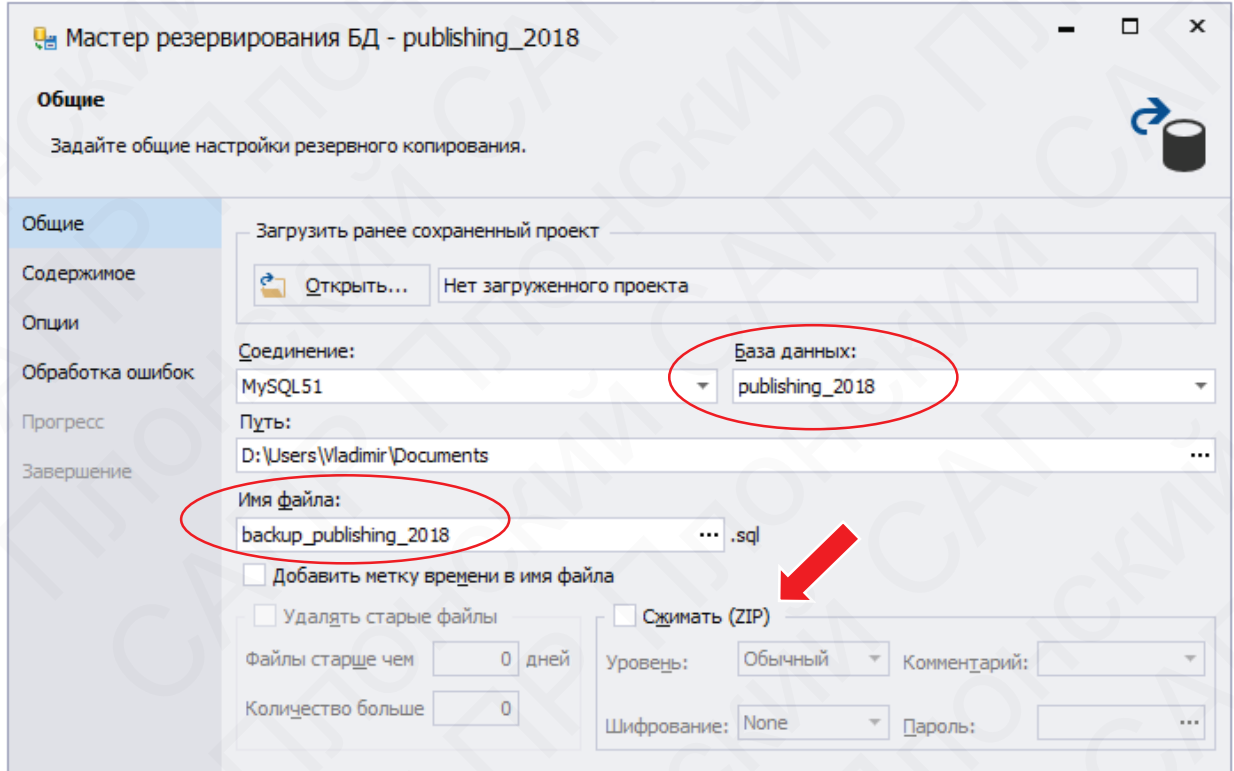


Рисунок 50. Данные таблицы после восстановления


```

--
-- Скрипт сгенерирован Devart dbForge Studio for MySQL, Версия 8.0.80.0
-- Домашняя страница продукта: http://www.devart.com/ru/dbforge/mysql/studio
-- Дата скрипта: 07.10.2018 19:08:31
-- Версия сервера: 5.1.40-community
-- Версия клиента: 4.1
--

--
-- Отключение внешних ключей
--
/*!40014 SET @OLD_FOREIGN_KEY_CHECKS=@@FOREIGN_KEY_CHECKS, FOREIGN_KEY_CHECKS=0 */;

--
-- Установить режим SQL (SQL mode)
--
/*!40101 SET @OLD_SQL_MODE=@@SQL_MODE, SQL_MODE='NO_AUTO_VALUE_ON_ZERO' */;

--
-- Установка кодировки, с использованием которой клиент будет посылать запросы на сервер
--
SET NAMES 'utf8';

--
-- Установка базы данных по умолчанию
--
USE publishing_2018;

--
-- Удалить таблицу `авторство`
--
DROP TABLE IF EXISTS авторство;

```

Рисунок 51. Фрагмент скрипта резервной копии

Откройте одну из таблиц БД и внесите в неё изменения (рисунок 52). Для этого при выполнении команды «Извлечь данные» необходимо включить возможность редактирования.

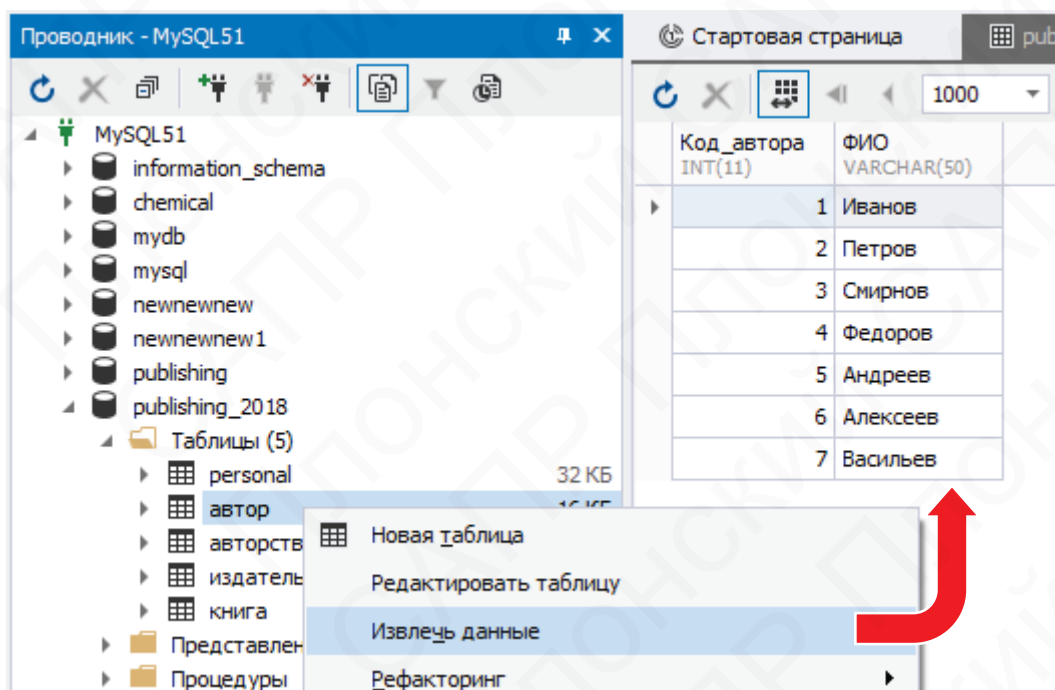


Рисунок 52. Открытие таблицы БД

На вкладке «Администрирование» выполните команду «Восстановить БД...» (рисунок 53). Укажите файл с резервной копией (рисунок 54) и приведите БД в исходное состояние (до внесения изменений в таблицу).

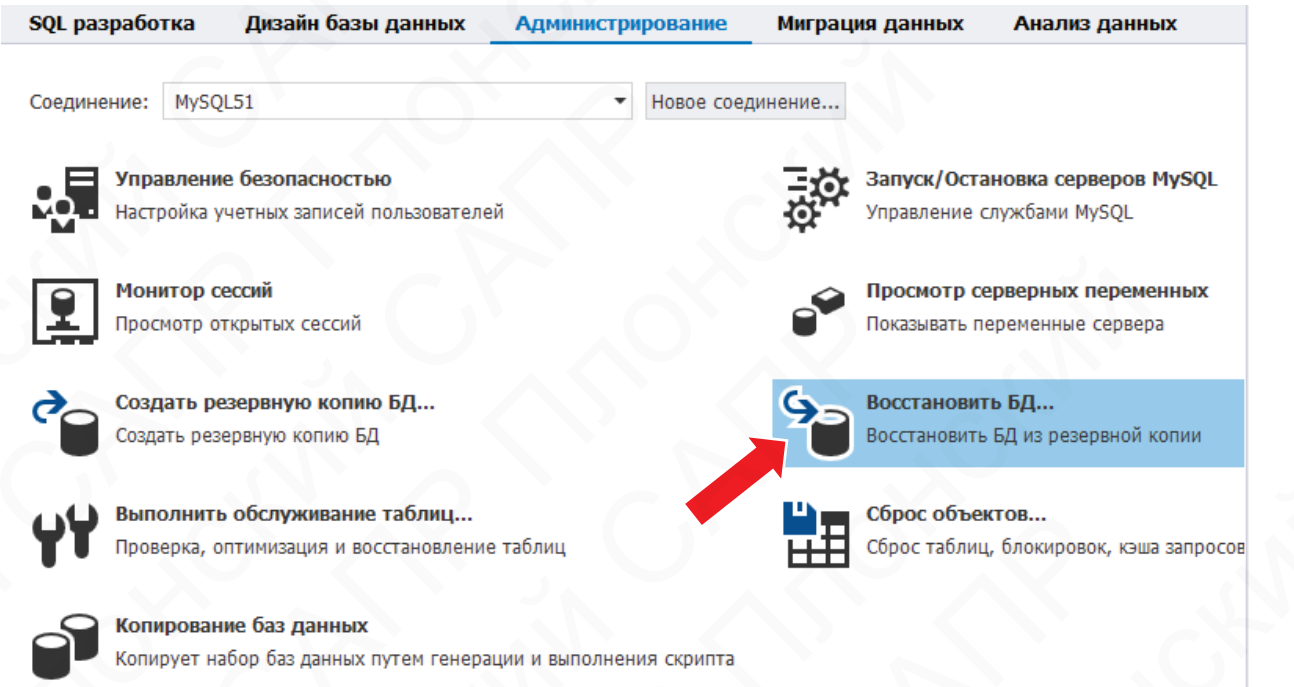


Рисунок 53. Восстановление из резервной копии

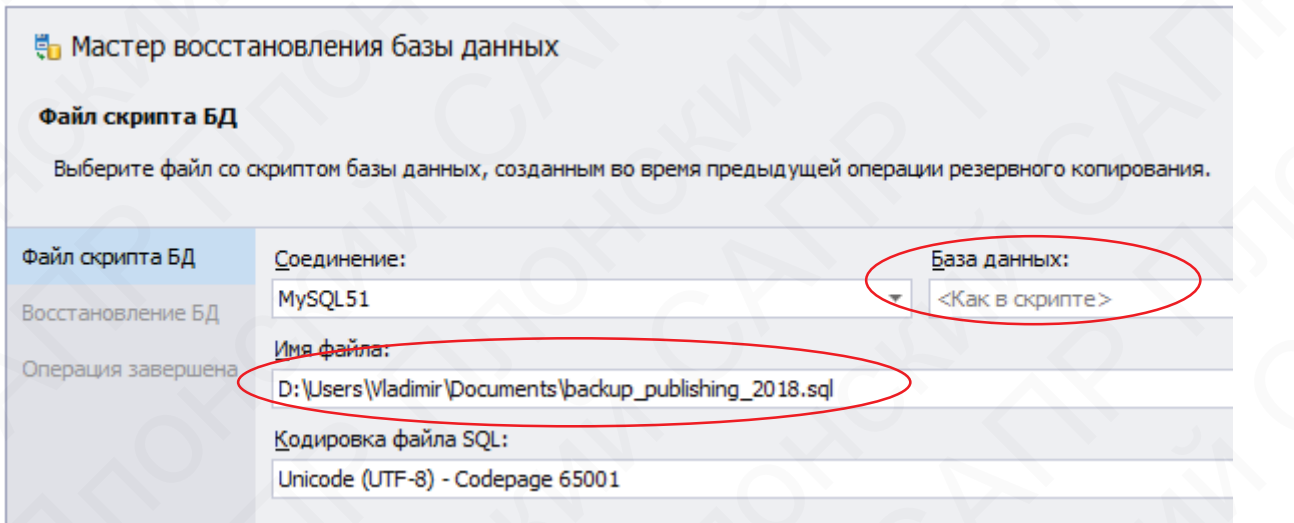


Рисунок 54. Выбор файла с резервной копией

6. Создание запросов к СУБД MYSQL

6.1 Добавление ограничения внешнего ключа

Добавьте в БД новую таблицу – некоторый классификатор (например, города, единицы измерения, ценовые категории или другие, подходящие для вашей предметной области). Создайте в нескольких таблицах столбцы, которые будут в дальнейшем внешними ключами к первичному ключу новой таблицы.

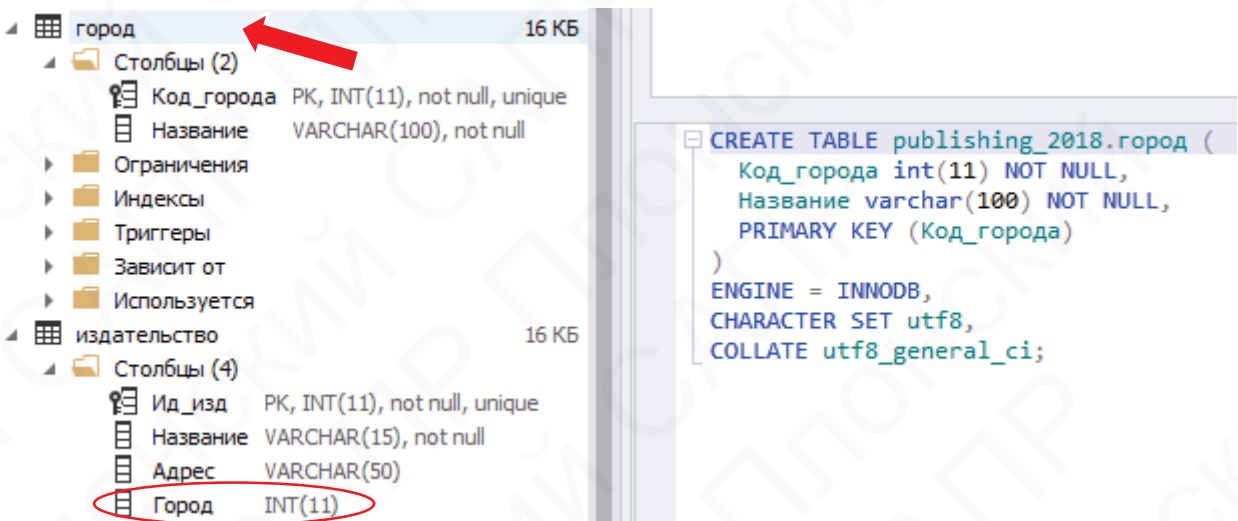


Рисунок 55. Новая таблица и DDL-код для её создания

Откройте диаграмму БД и добавьте на неё новую таблицу. В панели инструментов найдите и выберите команду «Новая связь»:

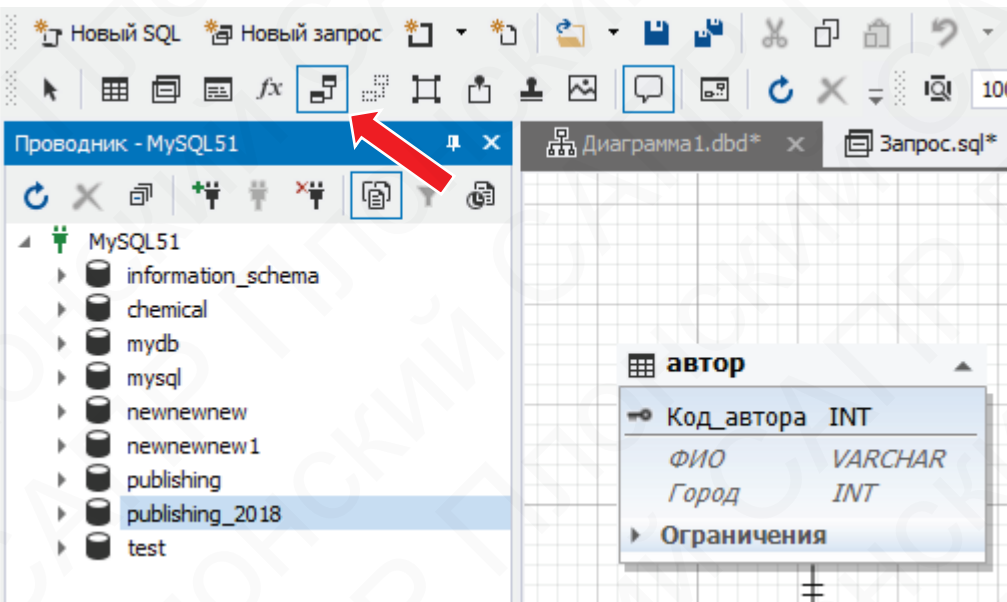


Рисунок 56. Команда «Новая связь»

На диаграмме перетащите поле из дочерней таблицы на первичный ключ новой (родительской таблицы) таблицы таблиц. В окне «Параметры внешнего ключа» выполните команду «Скрипт изменений». Откройте окно с текстом запроса и просмотрите его синтаксис (рисунок 58). Найдите в панели инструментов и запустите команду «Выполнить».

Параметры внешнего ключа

Общие Дополнительно

Имя внешнего ключа: Расположено

Ссылаться на базу данных: publishing_2018

Таблица

Имя: издательство

☐ Ид_изд <INT(11)> (ПК)
☐ Название <VARCHAR(15)>
☐ Адрес <VARCHAR(50)>
☒ Город <INT(11)>

Столбцы ключа: Город

Ссылаться на таблицу

Имя: город

☒ Код_города <INT(11)> (ПК)
☐ Название <VARCHAR(100)>

Ссылаться на столбцы: Код_города

☒ Показывать типы столбцов

Скрипт изменений Применить изменения Отмена

Рисунок 57. Команда «Новая связь»

```

--
-- Установка базы данных по умолчанию
--
USE publishing_2018;

--
-- Создать внешний ключ
--
ALTER TABLE издательство
ADD CONSTRAINT FK_издательство_город_Код_города FOREIGN KEY (Город)
REFERENCES город(Код_города) ON DELETE NO ACTION ON UPDATE NO ACTION;

```

Рисунок 58. Запрос на создание связи между таблицами

Примечание: изучите протокол работы запроса. Если в окне вывода есть сообщения об ошибках «errno: 150» или «errno: 121» (рисунок 59), одной из причин может быть отсутствие индекса в родительской таблице (по первичному ключу). Проверьте, что родительская таблица содержит первичный ключ, в запросе ограничение создается для дочерней таблицы, типы данных первичного и внешнего ключа совпадают. Поля в дочерней таблице тоже должны быть проиндексированы, но если нет подходящего индекса, MySQL автоматически его создаст при добавлении внешнего ключа.

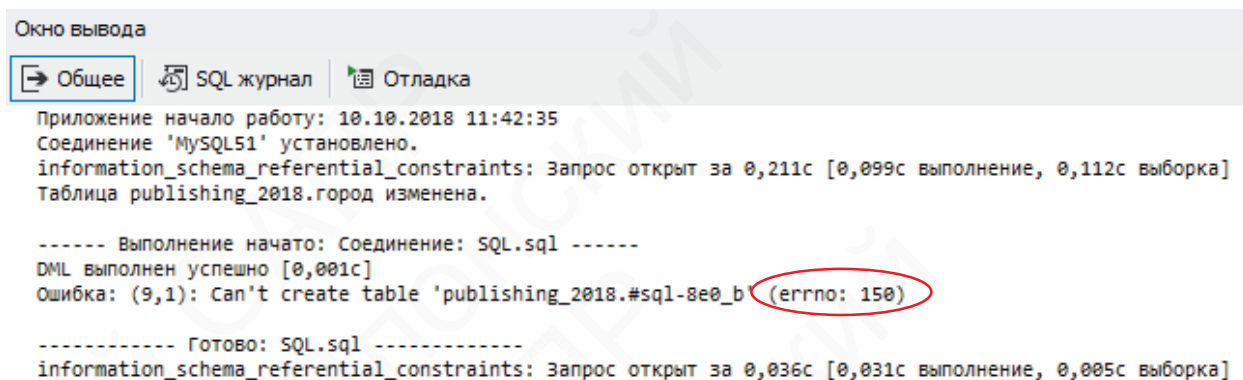


Рисунок 59. Протокол работы скрипта SQL (с ошибками)

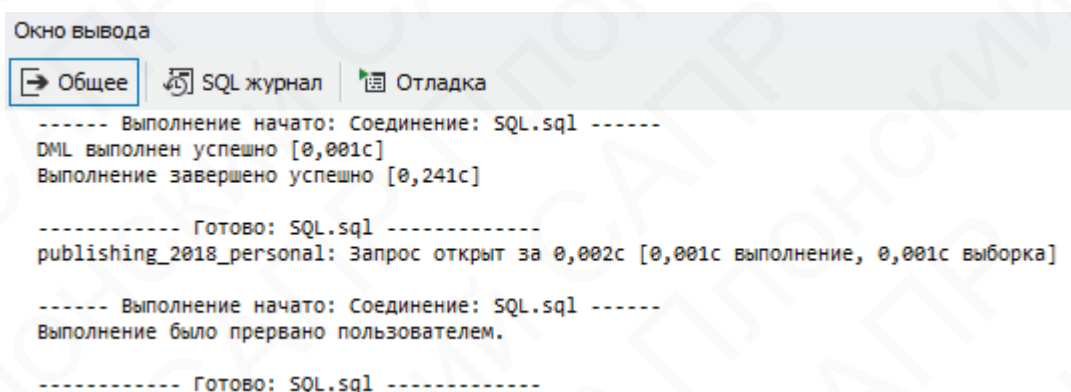


Рисунок 60. Протокол работы скрипта SQL (успешное выполнение)

Обновите диаграмму, на ней должна появиться новая связь:

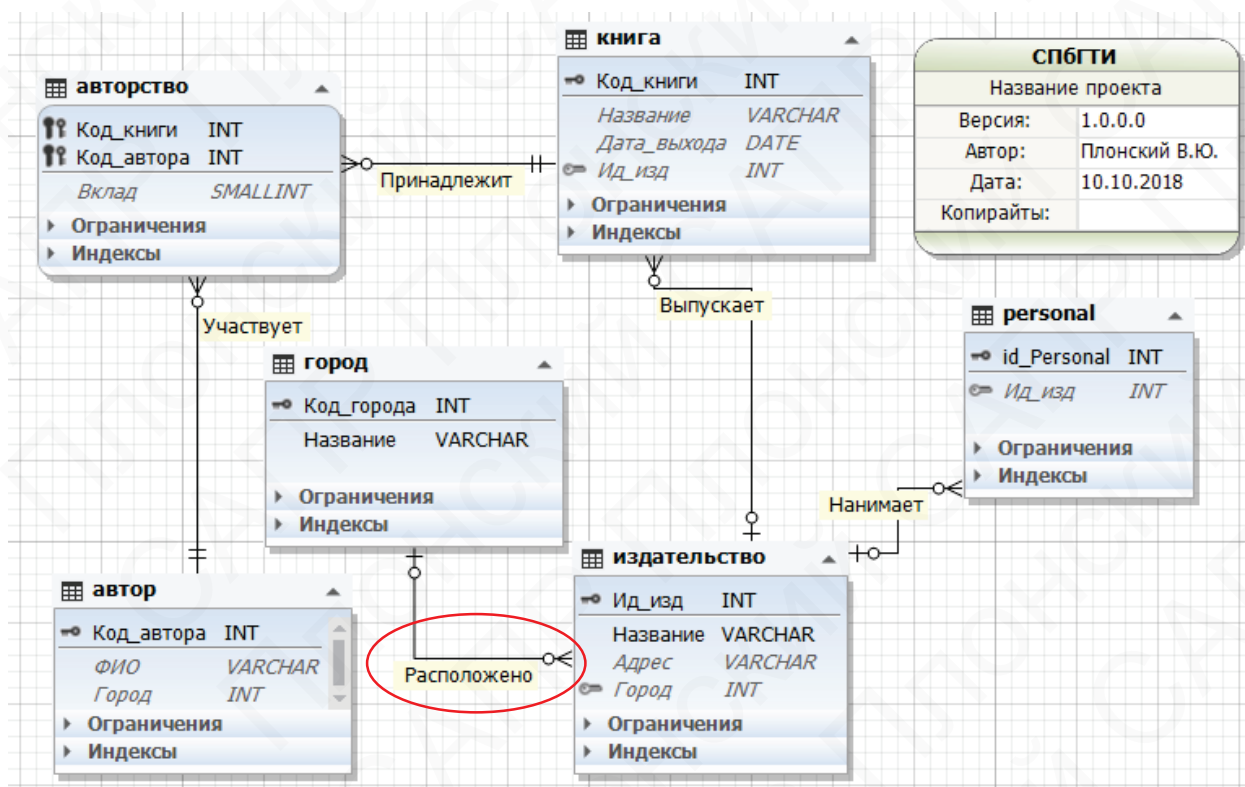


Рисунок 61. Протокол работы скрипта SQL (успешное выполнение)

Проверьте описание связи в словаре данных (база данных INFORMATION_SCHEMA, таблица REFERENTIAL_CONSTRAINTS):

CONSTRAINT_SCHEMA	CONSTRAINT_NAME	UNIQUE_CONSTRAINT_SCHEMA	UNIQUE_CONSTRAINT_NAME	MATCH_OPTION	UPDATE_RULE	DELETE_RULE	TABLE_NAME	REFERENCED_TABLE_NAME
publishing	Принадлежит	publishing	PRIMARY	NONE	NO ACTION	NO ACTION	авторство	книга
publishing	Участвует	publishing	PRIMARY	NONE	NO ACTION	NO ACTION	авторство	автор
publishing	Выпускает	publishing	PRIMARY	NONE	NO ACTION	NO ACTION	книга	издательство
publishing_2018	Принадлежит	publishing_2018	PRIMARY	NONE	NO ACTION	NO ACTION	авторство	книга
publishing_2018	Участвует	publishing_2018	PRIMARY	NONE	NO ACTION	NO ACTION	авторство	автор
publishing_2018	Расположено	publishing_2018	PRIMARY	NONE	NO ACTION	NO ACTION	издательство	город
publishing_2018	Выпускает	publishing_2018	PRIMARY	NONE	NO ACTION	NO ACTION	книга	издательство
publishing_2018	Нанимает	publishing_2018	PRIMARY	NONE	NO ACTION	NO ACTION	personal	издательство

Рисунок 62. Описание связи в словаре данных

6.2 Добавление столбца в таблицу

В редакторе SQL подготовьте запрос, добавляющий один или несколько столбцов в таблицу (рисунок 63). После выполнения запроса новую структуру таблицы можно посмотреть с помощью команды «SHOW COLUMNS» или в окне проводника dbForge Studio (рисунок 64).

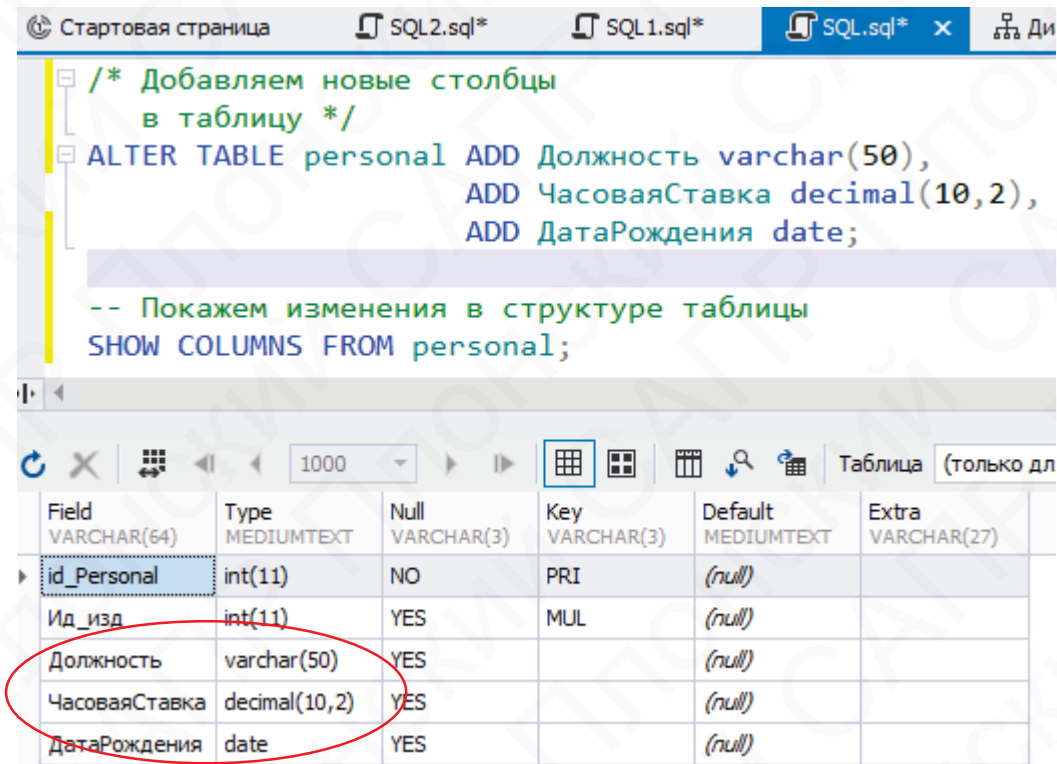


Рисунок 63. Запрос на добавление данных

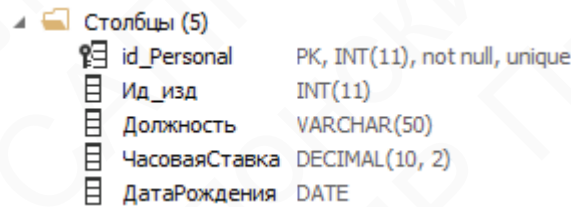


Рисунок 64. Запрос на добавление данных

6.3 Изменение определения столбца

Подготовьте запрос, изменяющий тип данных и ограничение столбца:

Стартовая страница

SQL1.sql* x

SQL.sql*

Диаграмма1.dbd*

```
/* Изменяем тип данных столбца и
   добавляем ограничение NOT NULL */
ALTER TABLE personal MODIFY Должность char(20) NOT NULL;

-- Покажем изменения в структуре таблицы
SHOW COLUMNS FROM personal;
```

1000

Таблица (только для чтения)

Field	Type	Null	Key	Default	Extra
id_Personal	int(11)	NO	PRI	(null)	
Ид_изд	int(11)	YES	MUL	(null)	
Должность	char(20)	NO		(null)	
ЧасоваяСтавка	decimal(10,2)	YES		(null)	
ДатаРождения	date	YES		(null)	

Рисунок 65. Запрос на изменение существующего столбца таблицы

Примечание: если необходимо изменить имя столбца, то вместо команды «ALTER TABLE ... MODIFY» можно использовать «ALTER TABLE ... CHANGE».

6.4 Удаление столбца таблицы

В редакторе SQL подготовьте запрос, удаляющий один или несколько столбцов в таблице:

Стартовая страница

SQL2.sql* x

SQL1.sql*

SQL.sql*

```
ALTER TABLE personal DROP Должность,
                        DROP ЧасоваяСтавка,
                        DROP ДатаРождения;

-- Покажем изменения в структуре таблицы
SHOW COLUMNS FROM personal;
```

1000

Таблица (только для чтения)

Field	Type	Null	Key	Default	Extra
id_Personal	int(11)	NO	PRI	(null)	
Ид_изд	int(11)	YES	MUL	(null)	

Рисунок 66. Запрос на удаление столбца таблицы

6.5 Параметрический запрос на выборку с условием

Сформулируйте запрос (например, вывести все издательства, расположенные в заданном пользователем городе).

Откройте дизайнер запросов и на рабочую область вкладки «Выборка» перетащите необходимые таблицы из проводника. Укажите требуемые поля для вывода (рисунок 67).

Примечание: для включения в запрос поля достаточно установить флажок для поля-метки с названием поля в области окна, отображающего структуру выбранных таблиц. Другой вариант формирования запроса заключается в выборе полей из раскрывающегося списка в колонке «Столбец». Так обычно в запрос добавляются составные выражения, содержащие функции или представляющие собой выражение из нескольких полей.

При необходимости уточните псевдонимы полей (именно они будут отображаться при выводе) и сортировку результата запроса

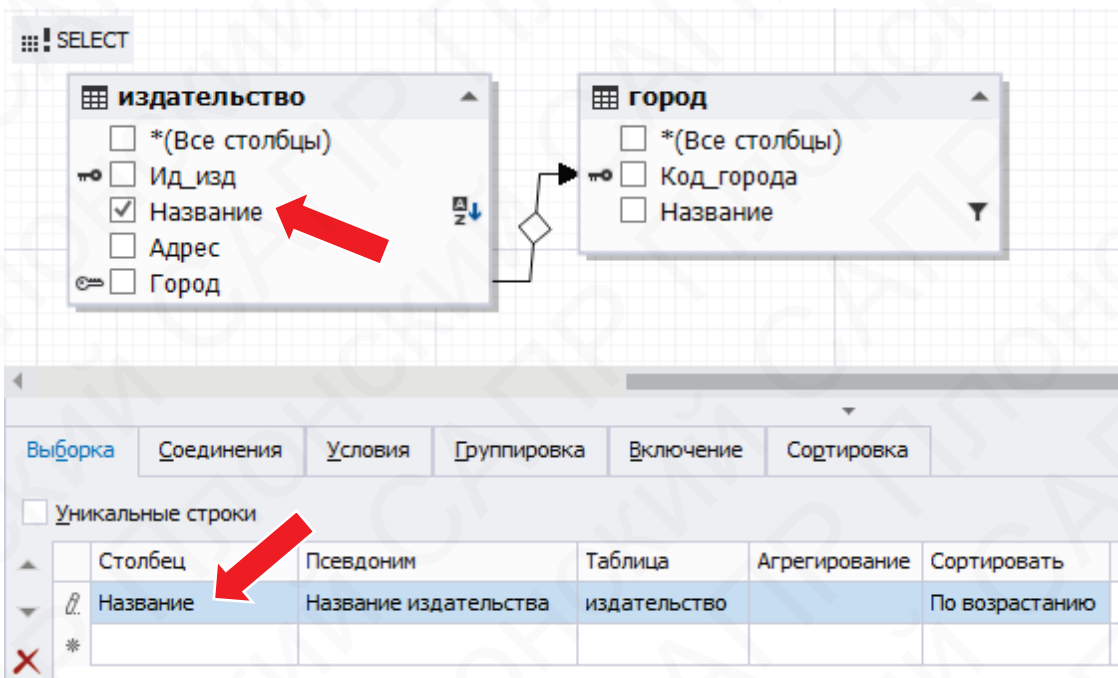


Рисунок 67. Поля вывода в запросе

На вкладке «Условия» добавьте необходимое количество условий, как минимум одно из которых должно включать параметр:

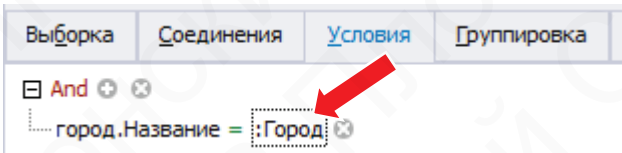


Рисунок 68. Параметр в условии запроса

Примечание: параметр вводится, указанием символа «:» перед идентификатором. Оператор условия можно заменить, выбрав другой из выпадающего списка. При нажатии символа логической операции появляется возможность конструировать сложные многоуровневые условия:

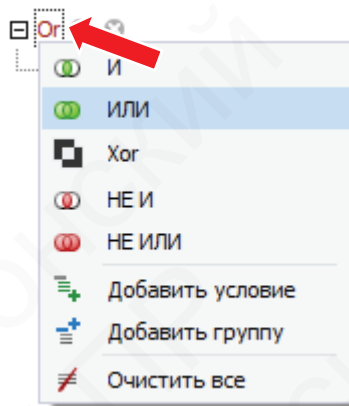


Рисунок 69. Конструирование сложного условия

Просмотрите текст запроса (рисунок 70). Вызовите команду «Выполнить» и заполните значения параметров запроса. Подтвердите ввод и проверьте соответствие полученного результата требованиям из формулировки запроса.

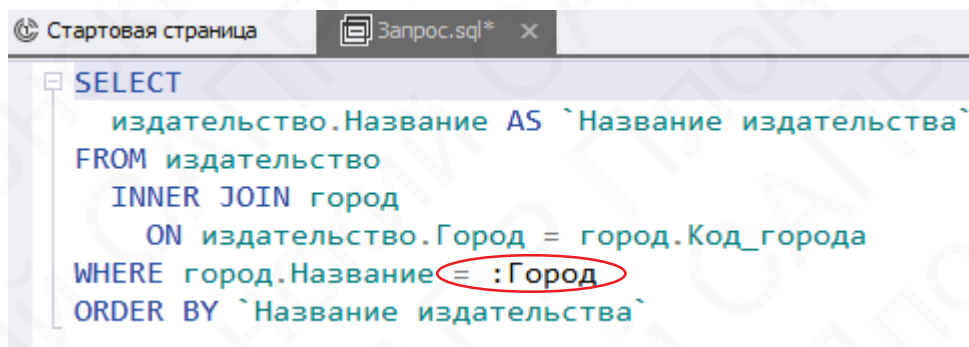


Рисунок 70. Текст запроса

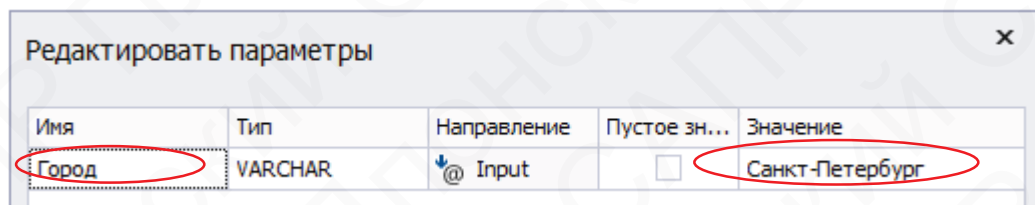


Рисунок 71. Ввод значений параметров

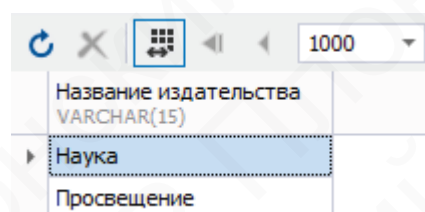


Рисунок 72. Результат запроса

6.6 Группировка записей и расчет агрегатных функций

Действие группировки дает возможность объединить одинаковые по какому-то признаку записи таблицы в группы и применять к ним вычисления с помощью разных функций (SUM, AVG, MAX, MIN, COUNT и др.).

Сформулируйте запрос (например, вывести авторов с указанием среднего вклада каждого автора в написанные им книги).

Откройте дизайнер запросов и на рабочую область вкладки «Выборка» перетащите необходимые таблицы из проводника. Укажите требуемые поля для вывода. На вкладке «Выборка» в столбце «Агрегирование» выберите подходящую агрегатную функцию (рисунок 73).

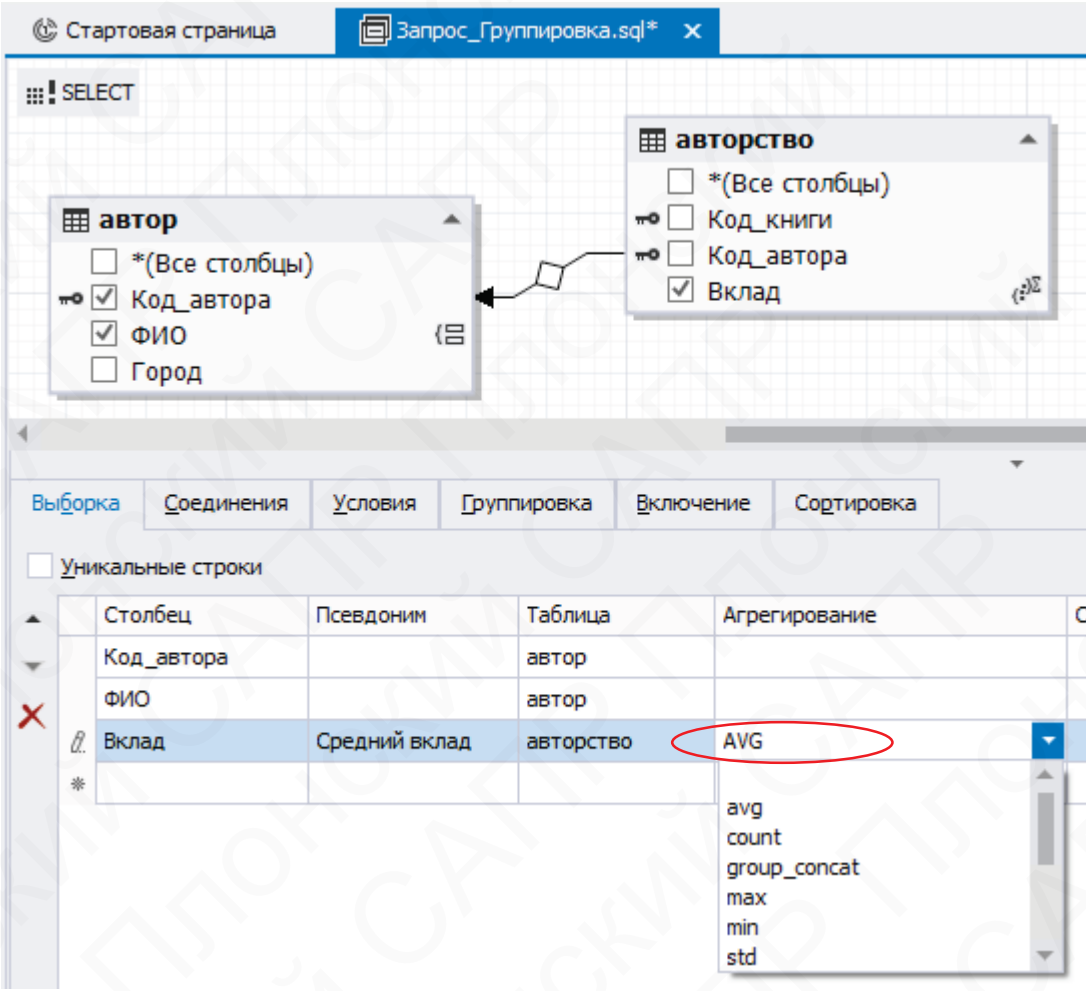


Рисунок 73. Запрос с группировкой

Примечание: по полям, для которых не указана агрегатная функция на вкладке «Выборка» будет установлена группировка (рисунок 74). Изменить группировку записей можно на вкладке «Группировка». В СУБД MYSQL не требуется включать все неагрегированные столбцы в предложение GROUP BY. В некоторых других СУБД необходимо точно указать, что делать с объединенными строками, где для некоторых столбцов есть разные значения – включать в предложение GROUP BY или использовать в агрегатной функции.

```
SELECT
    автор.Код_автора,
    автор.ФИО,
    AVG(авторство.Вклад) AS `Средний вклад`
FROM авторство
INNER JOIN автор
    ON авторство.Код_автора = автор.Код_автора
GROUP BY
    автор.Код_автора,
    автор.ФИО
```

Рисунок 74. Запрос с группировкой

Код_автора INT(11)	ФИО VARCHAR(50)	Средний вклад DECIMAL(9, 4)
1	Иванов	20,0000
2	Петров	50,0000
3	Смирнов	55,0000
4	Федоров	45,0000
6	Алексеев	100,0000

Рисунок 75. Результат запроса с группировкой

Переформулируйте запрос, добавив в него условие, ограничивающее результат, возвращаемый группировкой (например, вывести авторов с указанием среднего вклада каждого автора в написанные им книги в том случае, если минимальное участие автора в группе не меньше определенного значения).

Примечание: перемещение условия из предложения HAVING в предложение WHERE (рисунок 76) в общем случае даст другой результат (рисунки 77, 78).

```

/* HAVING vs WHERE
16.10.2018. Плонский В.Ю.
*/
-- Запрос с ограничением на группу
SELECT
    автор.Код_автора,
    автор.ФИО,
    AVG(авторство.Вклад) AS `Средний вклад`
FROM авторство
    INNER JOIN автор
        ON авторство.Код_автора = автор.Код_автора
GROUP BY автор.Код_автора,
    автор.ФИО
HAVING MIN(авторство.Вклад) >= 20;

-- Запрос с ограничением на исходные данные
SELECT
    автор.Код_автора,
    автор.ФИО,
    AVG(авторство.Вклад) AS `Средний вклад`
FROM авторство
    INNER JOIN автор
        ON авторство.Код_автора = автор.Код_автора
WHERE авторство.Вклад >= 20
GROUP BY автор.Код_автора,
    автор.ФИО

```

Рисунок 76. Условие в разных предложениях запроса

Примечание: предложение WHERE не допускает использования агрегатных функций. Предложение HAVING можно использовать без агрегирующих функций, накладывая условия на столбцы из предложения GROUP BY (рисунок 79).

автор автор(1)

Код_автора INT(11)	ФИО VARCHAR(50)	Средний вклад DECIMAL(9, 4)
2	Петров	50,0000
3	Смирнов	55,0000
4	Федоров	45,0000
6	Алексеев	100,0000

Рисунок 77. Условие в HAVING

автор автор(1)

Код_автора INT(11)	ФИО VARCHAR(50)	Средний вклад DECIMAL(9, 4)
1	Иванов	25,0000
2	Петров	50,0000
3	Смирнов	55,0000
4	Федоров	45,0000
6	Алексеев	100,0000

Рисунок 78. Условие в WHERE

```
/* Отсутствие функций в HAVING
Плонский В.Ю. Для учебных целей.
*/
-- Запрос с ограничением на значение группы
SELECT
    автор.Код_автора,
    автор.ФИО,
    AVG(авторство.Вклад) AS `Средний вклад`
FROM авторство
    INNER JOIN автор
        ON авторство.Код_автора = автор.Код_автора
GROUP BY автор.Код_автора,
    автор.ФИО
HAVING автор.ФИО LIKE 'Иван%';
```

Рисунок 79. Условие в разных предложениях запроса

6.7 Группировка с пустыми значениями (NULL)

Отредактируйте данные какой-либо из таблиц таким образом, чтобы в будущем поле группировки появились NULL-значения. Для некоторых записей, по которым будет рассчитываться агрегатная функция также введите совпадающие и пустые значения (рисунок 80):

1000

Код_книги INT(11)	Название VARCHAR(50)	Дата_выхода DATE	Ид_изд INT(11)
1	Физика	01.01.2018	1
2	Химия	01.01.2018	1
3	Математика	(null)	2
4	Астрономия	(null)	(null)
5	Базы данных	03.10.2018	(null)

Рисунок 80. Запрос с группировкой

Сформулируйте запрос, в котором должна в разных вариантах применяться функция подсчета количества записей в группе COUNT (например, считая отдельным выпуском каждую дату выхода, вывести для каждого издательства количество выпусков, количество различных выпусков и общее количество книг).


```

SELECT
    книга.Ид_изд,
    COUNT(книга.Дата_выхода) AS `Всего выпусков`,
    COUNT(DISTINCT книга.Дата_выхода) `Разных выпусков`,
    COUNT(*) AS `Всего книг`
FROM книга
GROUP BY книга.Ид_изд;

```

Рисунок 81. Текст запроса с группировкой (функция COUNT)

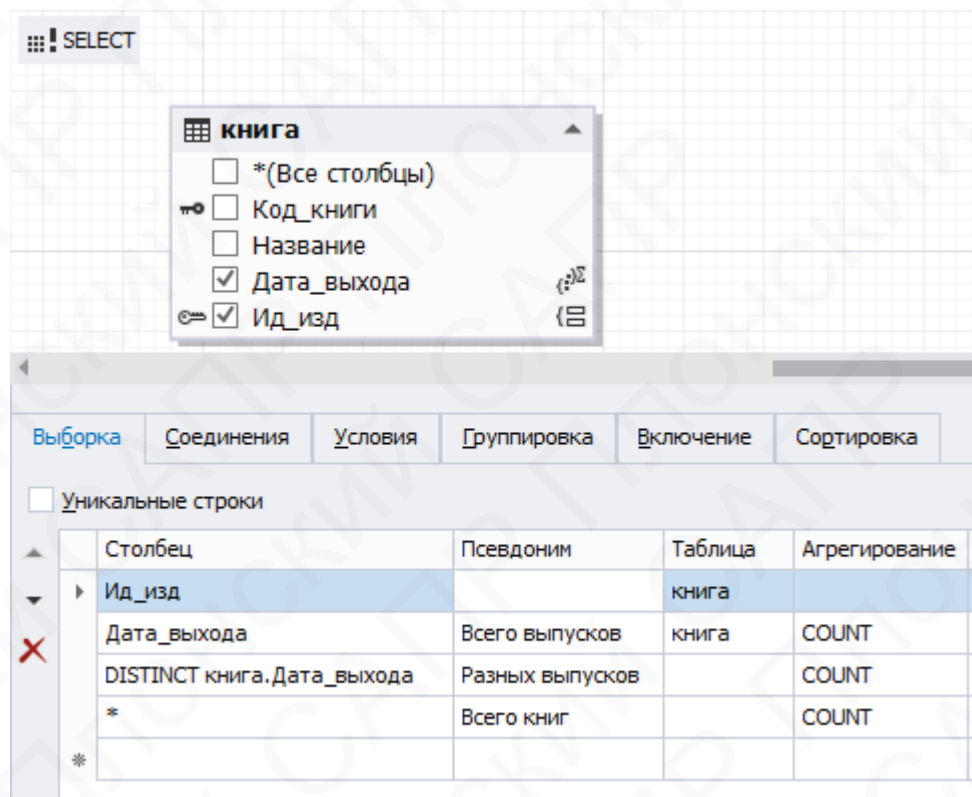


Рисунок 82. Запрос с группировкой (функция COUNT)

Проанализируйте результат запроса, сделайте вывод о том, как СУБД интерпретирует NULL-значения при агрегировании записей с функцией COUNT.

Ид_изд INT(11)	Всего выпусков BIGINT(21)	Разных выпусков BIGINT(21)	Всего книг BIGINT(21)
(null)	1	1	2
1	2	1	2
2	0	0	1

Рисунок 83. Результат запроса (функция COUNT)

6.8 Запросы по нескольким таблицам

Очень редко бывают ситуации, когда данные, которые необходимо получить, находятся в одной таблице. Чаще всего они находятся в различных таблицах, но должны быть получены вместе. Связь между таблицами устанавливается с помощью условия связи.

Различают несколько видов соединения таблиц: внутреннее, левое внешнее, правое внешнее, полное внешнее.

Сформулируйте запрос, в котором собрана информация из нескольких таблиц (например, вывести информацию о книгах с данными о выпустивших их издательствах).

Откройте дизайнер запросов и на рабочую область вкладки «Выборка» перетащите необходимые таблицы из проводника. Укажите требуемые поля для вывода. На вкладке «Выборка» в столбце «Псевдоним» уточните названия столбцов для вывода, если имена некоторых исходных полей таблиц совпадают (рисунок 84). На вкладке «Соединение» посмотрите вид соединения и условие связи (рисунок 85).

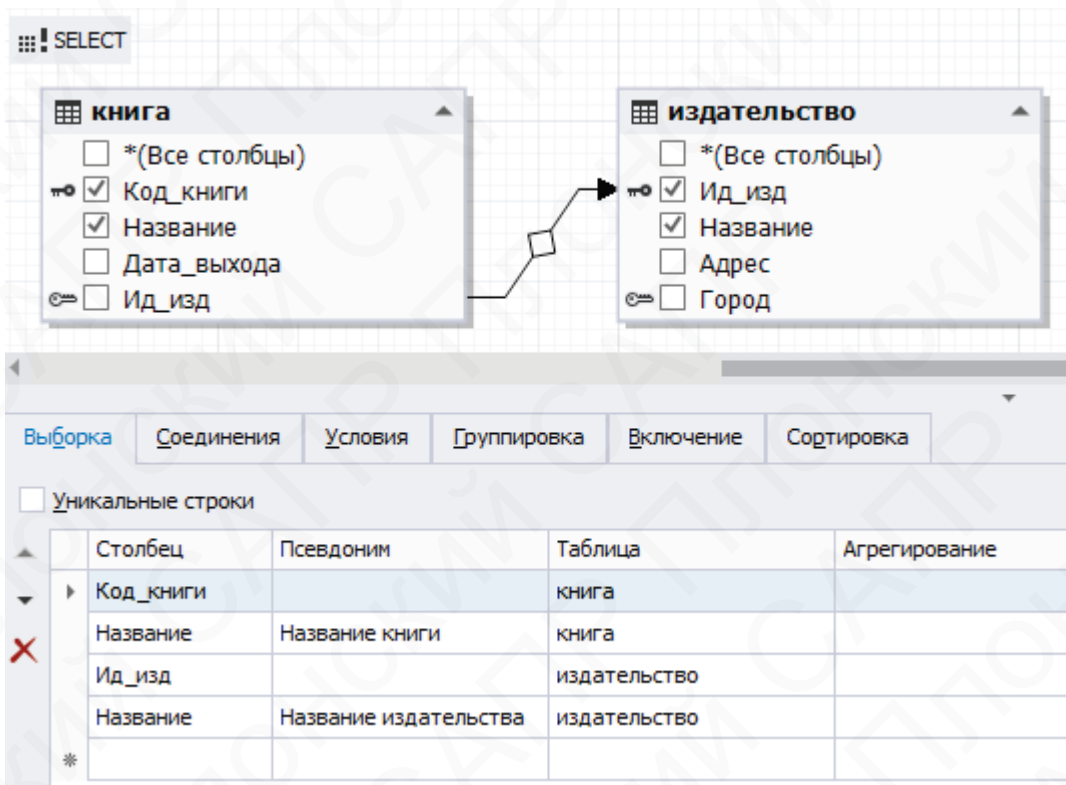


Рисунок 84. Многотабличный запрос

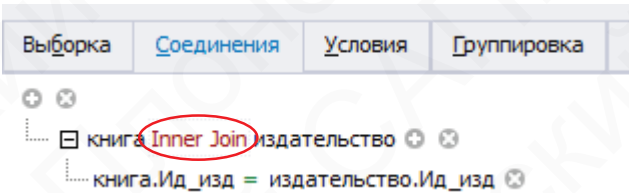


Рисунок 85. Внутреннее соединение

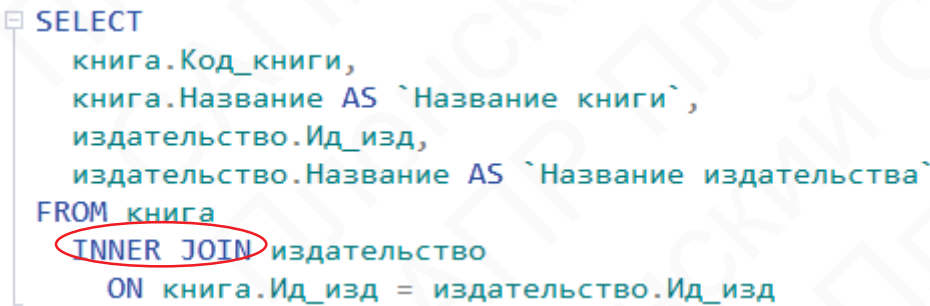
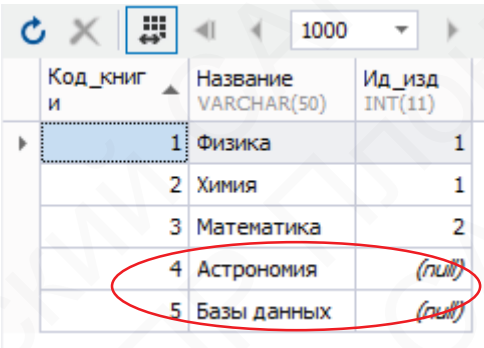


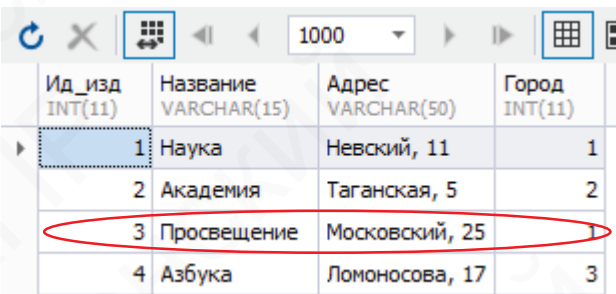
Рисунок 86. Внутреннее соединение (текст запроса)

Примечание: в результат выполнения запроса войдут только данные записей из обеих таблиц, для которых выполняется условие соединения. На рисунках выделены записи, не удовлетворяющие условию соединения в каждой таблице (строки с кодами 4 и 5 в таблице №1 «Книга» и строка с кодом 3 в таблице №2 «Издательство»):



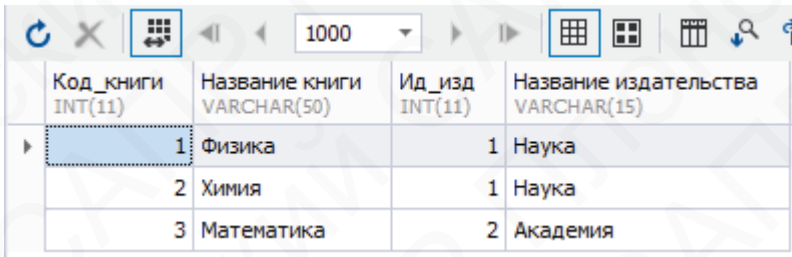
Код_книги	Название	Ид_изд
1	Физика	1
2	Химия	1
3	Математика	2
4	Астрономия	(null)
5	Базы данных	(null)

Рисунок 87. Таблица №1 «Книга»



Ид_изд	Название	Адрес	Город
1	Наука	Невский, 11	1
2	Академия	Таганская, 5	2
3	Просвещение	Московский, 25	1
4	Азбука	Ломоносова, 17	3

Рисунок 88. Таблица №2 «Издательство»



Код_книги	Название книги	Ид_изд	Название издательства
1	Физика	1	Наука
2	Химия	1	Наука
3	Математика	2	Академия

Рисунок 89. Результат внутреннего соединения

Переформулируйте исходный запрос, добавив в него данные записей из первой таблицы, даже если для них не выполняется условие соединения (например, вывести информацию о всех книгах с данными о выпустивших их издательствах, даже если для книги не указано, какое издательство её выпустило). Откорректируйте на вкладке «Соединение» вид соединения, просмотрите как изменился текст запроса:

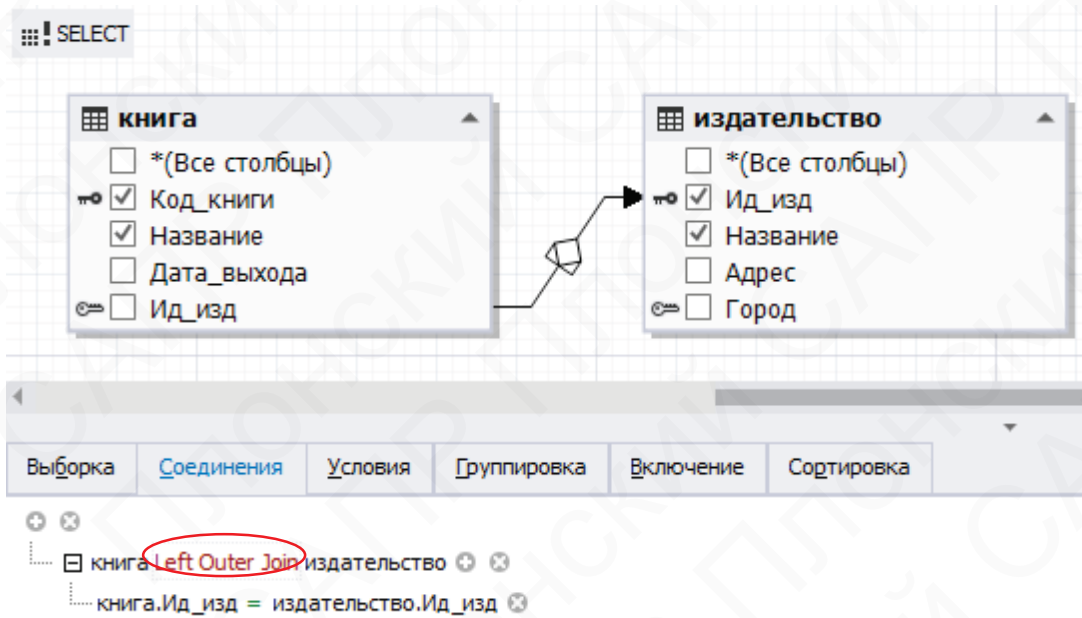


Рисунок 90. Левое внешнее соединение

```

SELECT
    книга.Код_книги,
    книга.Название AS `Название книги`,
    издательство.Ид_изд,
    издательство.Название AS `Название издательства`
FROM книга
LEFT OUTER JOIN издательство
ON книга.Ид_изд = издательство.Ид_изд

```

Рисунок 91. Левое внешнее соединение (текст запроса)

Примечание: в результат выполнения запроса войдут все данные из таблицы №1 (в общем случае многократно), причем для тех записей результата, для которых выполняется условие соединения, в полях, куда помещаются данные из таблицы №2, будут стоять значения, а для которых не выполняется, будет стоять null (рисунок 92).

Код_книги INT(11)	Название книги VARCHAR(50)	Ид_изд INT(11)	Название издательства VARCHAR(15)
1	Физика	1	Наука
2	Химия	1	Наука
3	Математика	2	Академия
4	Астрономия	(null)	(null)
5	Базы данных	(null)	(null)

Рисунок 92. Результат левого внешнего соединения

Переформулируйте исходный запрос, добавив в него данные записей из второй таблицы, даже если для них не выполняется условие соединения (например, вывести информацию о всех издательствах с данными о выпущенных книгах, даже если сведений о книгах данного издательства нет). Откорректируйте на вкладке «Соединение» вид соединения, просмотрите как изменился текст запроса:

```

SELECT
    книга.Код_книги,
    книга.Название AS `Название книги`,
    издательство.Ид_изд,
    издательство.Название AS `Название издательства`
FROM книга
RIGHT OUTER JOIN издательство
ON книга.Ид_изд = издательство.Ид_изд

```

Рисунок 93. Правое внешнее соединение (текст запроса)

Примечание: в результат выполнения запроса войдут все данные из таблицы №2 (в общем случае многократно), причем для тех записей результата, для которых выполняется условие соединения, в полях, куда помещаются данные из таблицы №1, будут стоять значения, а для которых не выполняется, будет стоять null (рисунок 94).

Код_книги INT(11)	Название книги VARCHAR(50)	Ид_изд INT(11)	Название издательства VARCHAR(15)
1	Физика	1	Наука
2	Химия	1	Наука
3	Математика	2	Академия
(null)	(null)	3	Просвещение
(null)	(null)	4	Азбука

Рисунок 94. Результат правого внешнего соединения

Примечание: полное внешнее соединение (FULL OUTER JOIN) не поддерживается СУБД MySQL. Эквивалентный результат можно получить операцией объединения левого и правого внешнего соединения.

```

/* Полное внешнее соединение.
-- Плонский В.Ю. Для учебных целей.
*/
SELECT
    книга.Код_книги,
    книга.Название AS `Название книги`,
    издательство.Ид_изд,
    издательство.Название AS `Название издательства`
FROM книга
LEFT OUTER JOIN издательство
ON книга.Ид_изд = издательство.Ид_изд
UNION
SELECT
    книга.Код_книги,
    книга.Название AS `Название книги`,
    издательство.Ид_изд,
    издательство.Название AS `Название издательства`
FROM книга
RIGHT OUTER JOIN издательство
ON книга.Ид_изд = издательство.Ид_изд

```

Рисунок 95. Аналог полного внешнего соединения (текст запроса)

Код_книги INT(11)	Название книги VARCHAR(50)	Ид_изд INT(11)	Название издательства VARCHAR(15)
1	Физика	1	Наука
2	Химия	1	Наука
3	Математика	2	Академия
4	Астрономия	(null)	(null)
5	Базы данных	(null)	(null)
(null)	(null)	3	Просвещение
(null)	(null)	4	Азбука

Рисунок 96. Результат полного внешнего соединения

Примечание: в результат запроса войдут как записи, для которых выполнялось условие соединения, так и записи, полученные из не вошедших данных обеих таблиц. В любом случае, в результате использования полного внешнего соединения не получается полная комбинация значений – добавляются записи, не удовлетворяющие условию, а не все возможные их комбинации.

6.9 Запрос на вставку записей из другой таблицы

Сформулируйте запрос, в котором необходимо добавить записи в таблицу на основании данных другой таблицы или нескольких таблиц (например, вставить в таблицу информацию об издательствах, которые не выпускают книг).

Создайте копию подходящей таблицы, выбрав её в проводнике и вызвав контекстную команду «Дублировать объект...». В диалоговом окне не устанавливайте флажок «Копировать данные».

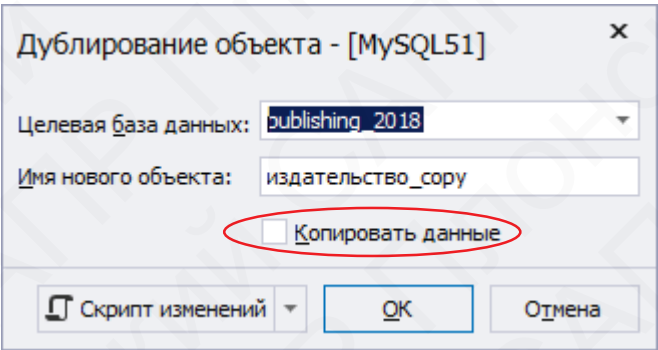


Рисунок 97. Дублирование объекта

Если по смыслу вашей задачи, необходимы дополнительные столбцы или, наоборот, некоторые столбцы являются лишними, измените структуру таблицы-копии. Подготовьте запрос, применив один из вариантов оператора INSERT с SELECT.

```
/* Создание копии таблицы.
-- Плонский В.Ю. Для учебных целей.
*/

INSERT INTO издательство_сору (Ид_изд, Название)

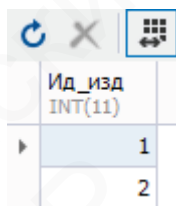
SELECT
    Ид_изд, Название
FROM издательство
WHERE Ид_изд NOT IN
    (SELECT DISTINCT Ид_изд FROM книга WHERE Ид_изд IS NOT NULL);
```

Рисунок 98. Вставка в таблицу из другой таблицы

Примечание: структуры таблиц должны быть одинаковыми. Более надежно, когда в тексте запроса сопоставлены имена столбцов таблицы вставки с таблицами, содержащими данные.

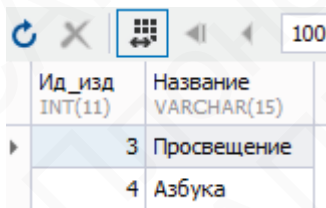
Примечание: в примере, на рисунке 95, в запросе, извлекающем данные из источника, используется вложенный подзапрос, возвращающий уникальные идентификаторы издательств, выпускающих книги. Незаполненные значения внешнего ключа таблицы «Книга» исключаются из набора результатов по условию проверки на пустые значения («... Ид_изд IS NOT NULL...»).

Примечание: можно посмотреть результат подзапроса, можно выделить только его в окне редактора и выполнить выделенный фрагмент:



Ид_изд INT(11)
1
2

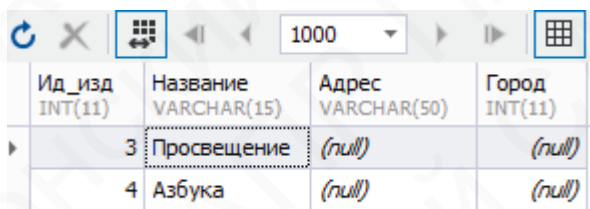
Рисунок 99. Результат внутреннего подзапроса



Ид_изд INT(11)	Название VARCHAR(15)
3	Просвещение
4	Азбука

Рисунок 100. Результат запроса к источнику данных

Выполните запрос и просмотрите данные таблицы-копии:



Ид_изд INT(11)	Название VARCHAR(15)	Адрес VARCHAR(50)	Город INT(11)
3	Просвещение	(null)	(null)
4	Азбука	(null)	(null)

Рисунок 101. Результат вставки записей

6.10 Запрос на вставку новой записи в таблицу с помощью хранимой процедуры

Базовый синтаксис оператора INSERT позволяет вводить новые записи в таблицу одну за другой:

INSERT INTO *ИмяТаблицы* (столбец1, столбец2, ...) VALUES (значение1, значение2, ...);

Примечание: изменение порядка столбцов не влияет на запрос INSERT, пока правильные значения были сопоставлены с правильными столбцами. По умолчанию MySQL будет вставлять значения NULL в столбцы, пропущенные в запросе INSERT. Если поставляются значения для всех столбцов в таблице, то можно столбцы не указывать.

Сформулируйте запрос, в котором необходимо добавить запись в таблицу на основании известных значений (например, вставить в таблицу информацию о новом издательстве). Используйте таблицу-копию для этой задачи:

INSERT INTO издательство_сору (Ид_изд, Название) VALUES (3, «Архангельск»)

Обратите внимание, что запрос на вставку может нарушить правило целостности данных и, соответственно, ошибку выполнения запроса, так как необходимо обеспечить уникальность значения первичного ключа для новой записи. Например, на рисунке 101 видно, что запись со значением ключа 3 уже существует.

В проводнике добавьте новую процедуру и определите её параметры:

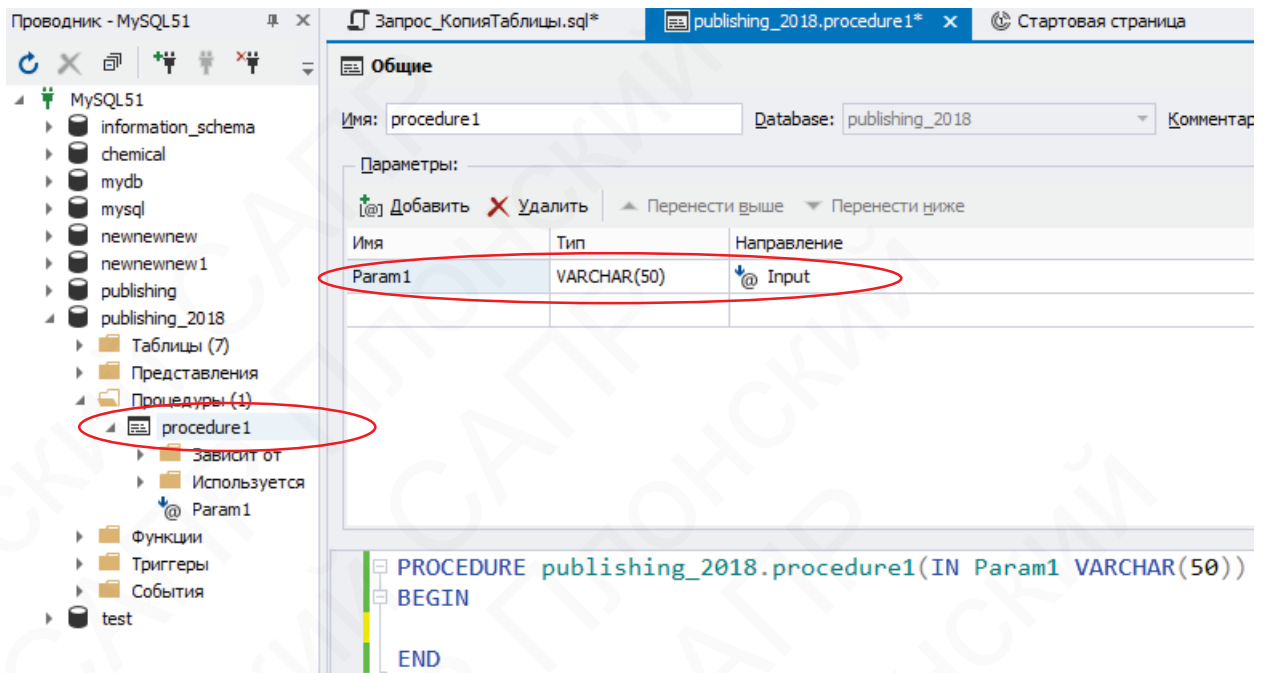


Рисунок 102. Хранимая процедура

Отредактируйте текст процедуры, выполняющий операцию вставки записи в таблицу. Откомпилируйте процедуру, выполнив команду «Компилировать» из меню «Отладка»:

```
PROCEDURE publishing_2018.procedure1(IN Param1 VARCHAR(50))
BEGIN
    DECLARE MaxId int;
    SET MaxId = (SELECT MAX(Ид_изд) FROM publishing_2018.издательство_сору);

    INSERT INTO publishing_2018.издательство_сору (Ид_изд, Название)
        VALUES (1, Param1)
    ON DUPLICATE KEY UPDATE Ид_изд=MaxId + 1;
END
```

Рисунок 103. Текст хранимой процедуры

Примечание: оператор DECLARE позволяет описать новую переменную процедуры. Оператор SET позволяет присвоить значение переменной процедуры.

В редакторе SQL подготовьте и выполните запросы:

CALL procedure1 ("Строительство");

CALL procedure1 ("Научная мысль");

Ид_изд INT(11)	Название VARCHAR(15)	Адрес VARCHAR(50)	Город INT(11)
1	Строительство	(null)	(null)
3	Просвещение	(null)	(null)
4	Азбука	(null)	(null)
5	Научная мысль	(null)	(null)

Рисунок 104. Результат работы хранимой процедуры