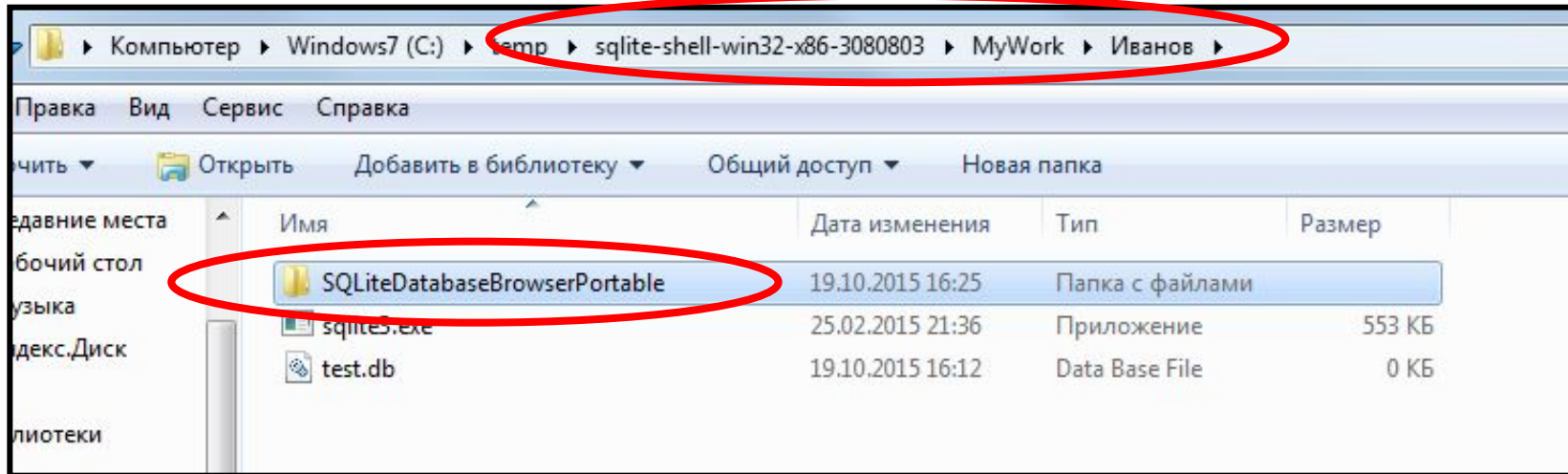




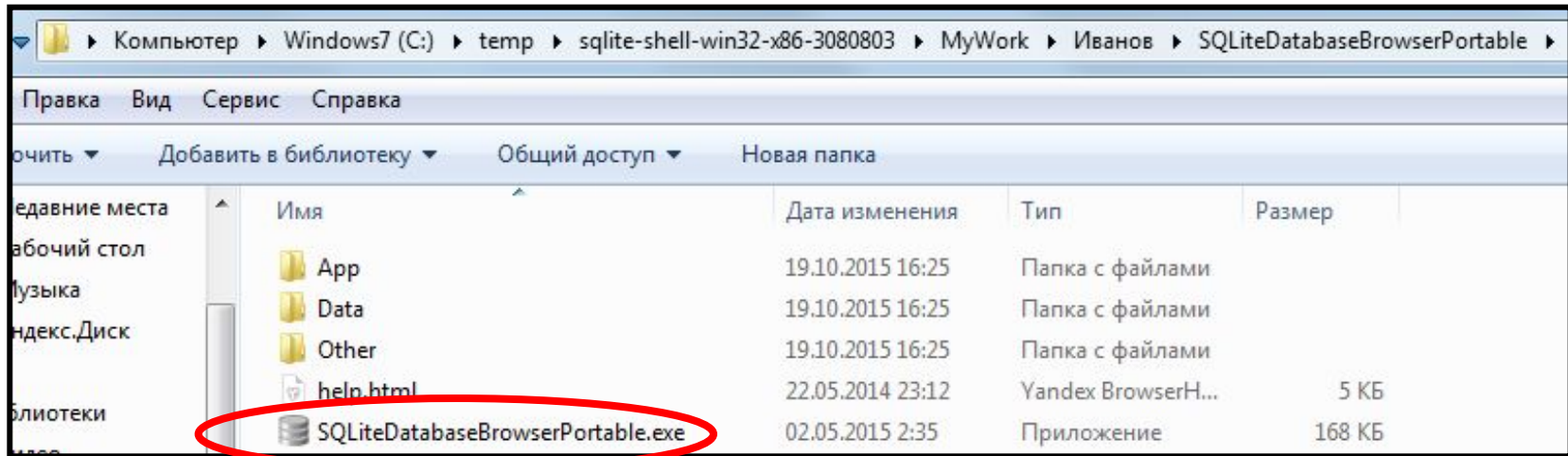
**SQLite менеджер**  
**Создание БД и таблиц**  
**DDL и DML запросы**

# Запуск менеджера SQLite



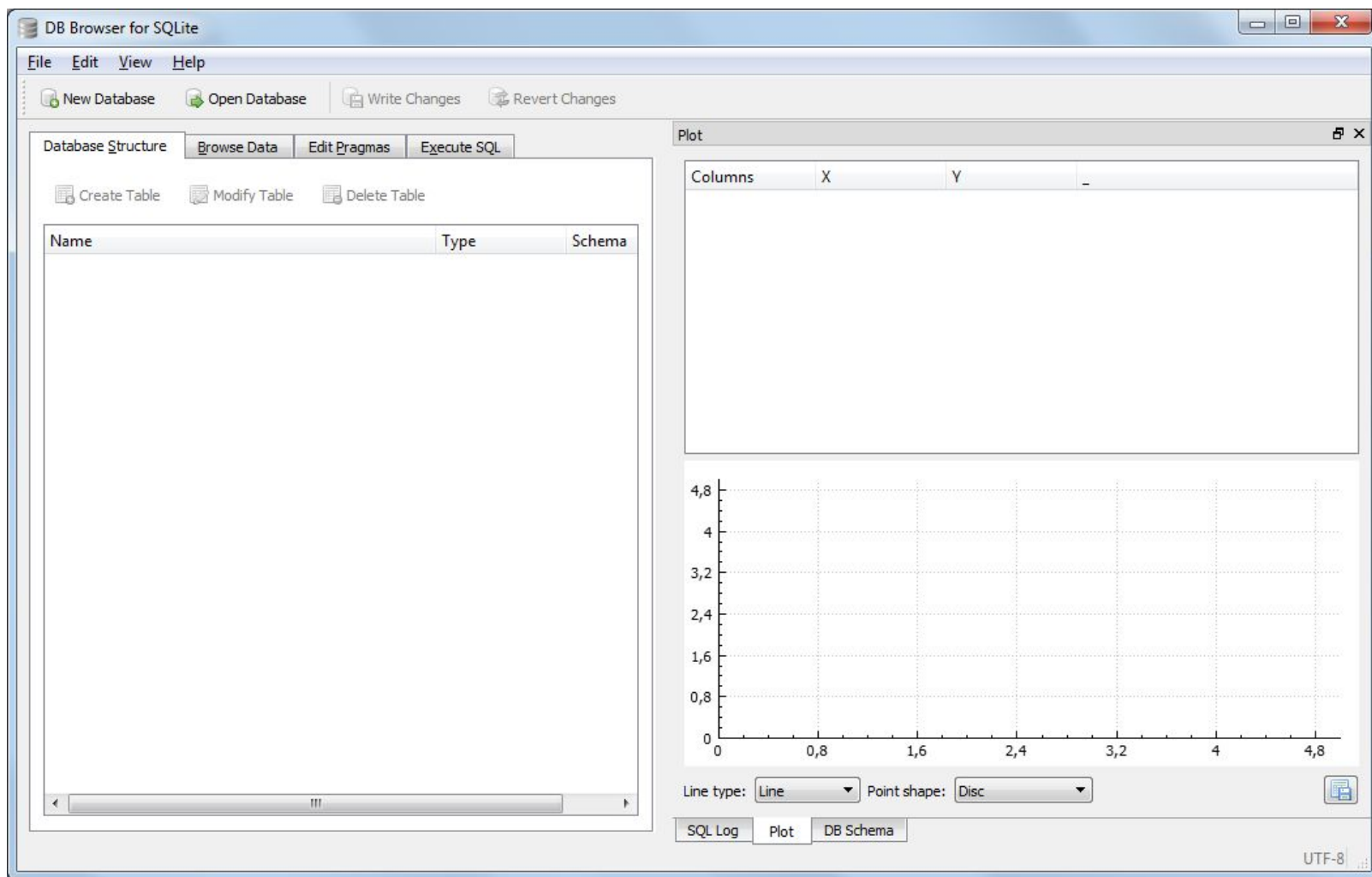
1. Откройте свою рабочую папку.
2. Выберите папку менеджера баз данных SQLite

# Запуск менеджера SQLite

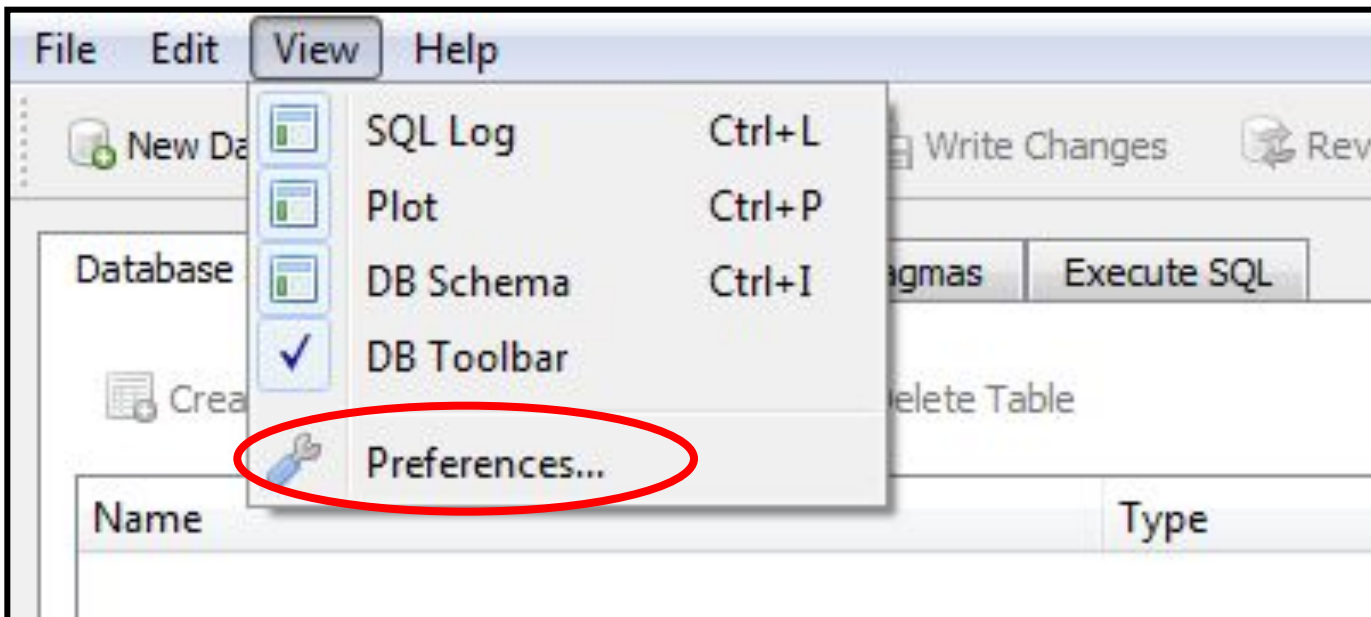


**Запустите программу менеджера**

# Окно программы

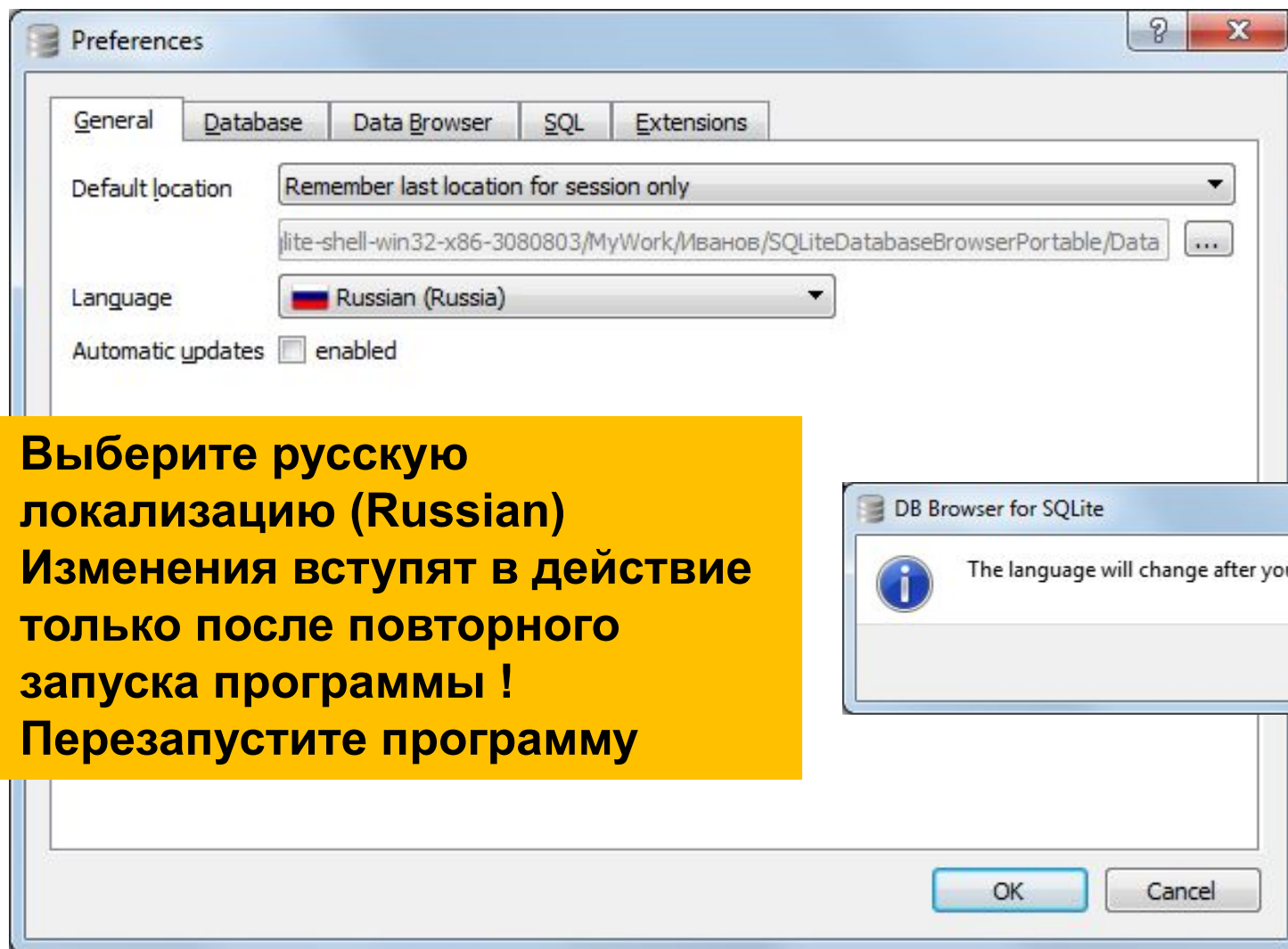


# Локализация программы



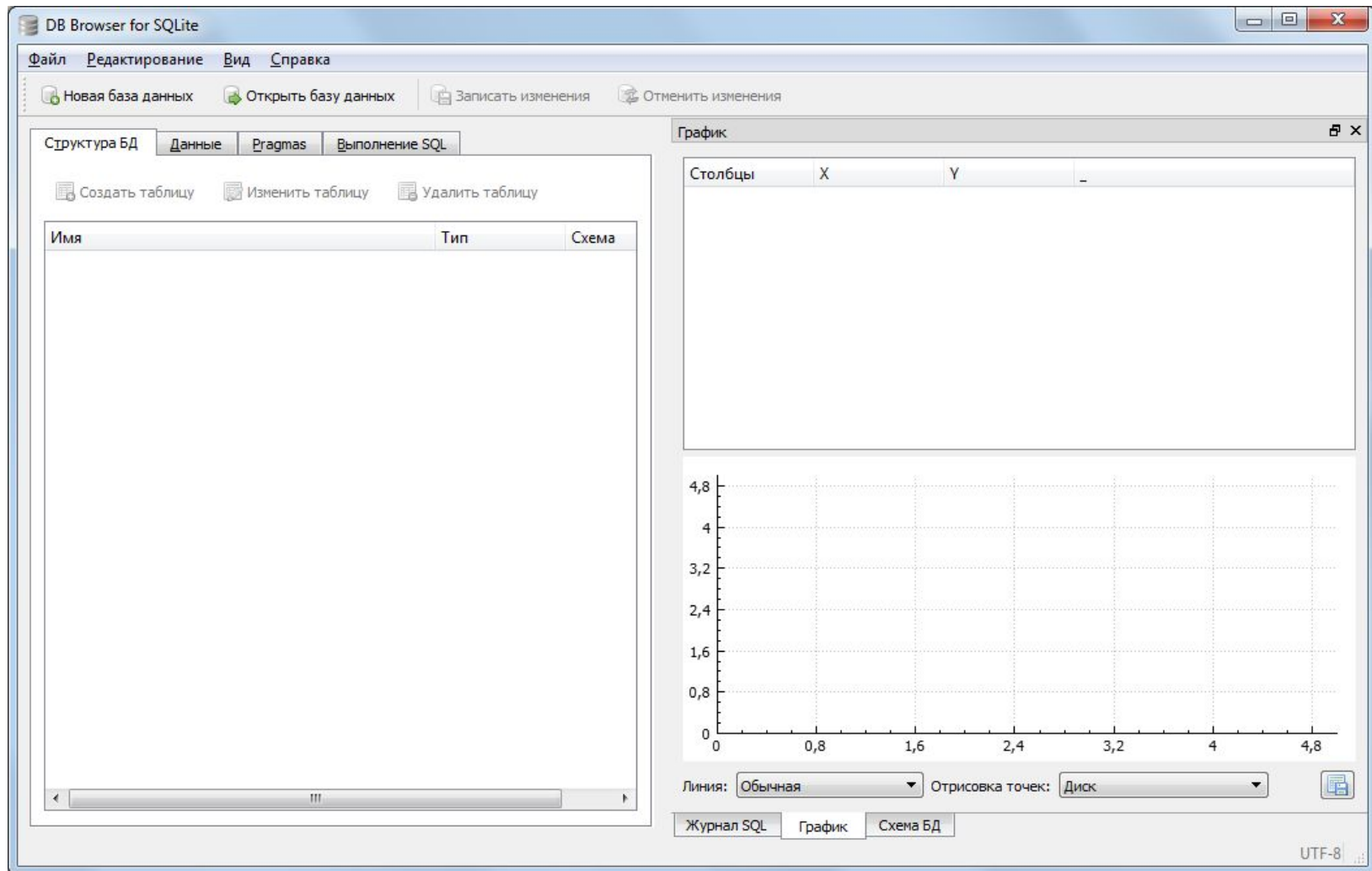
**Выберите пункт меню View и команду Preferences**

# Локализация программы

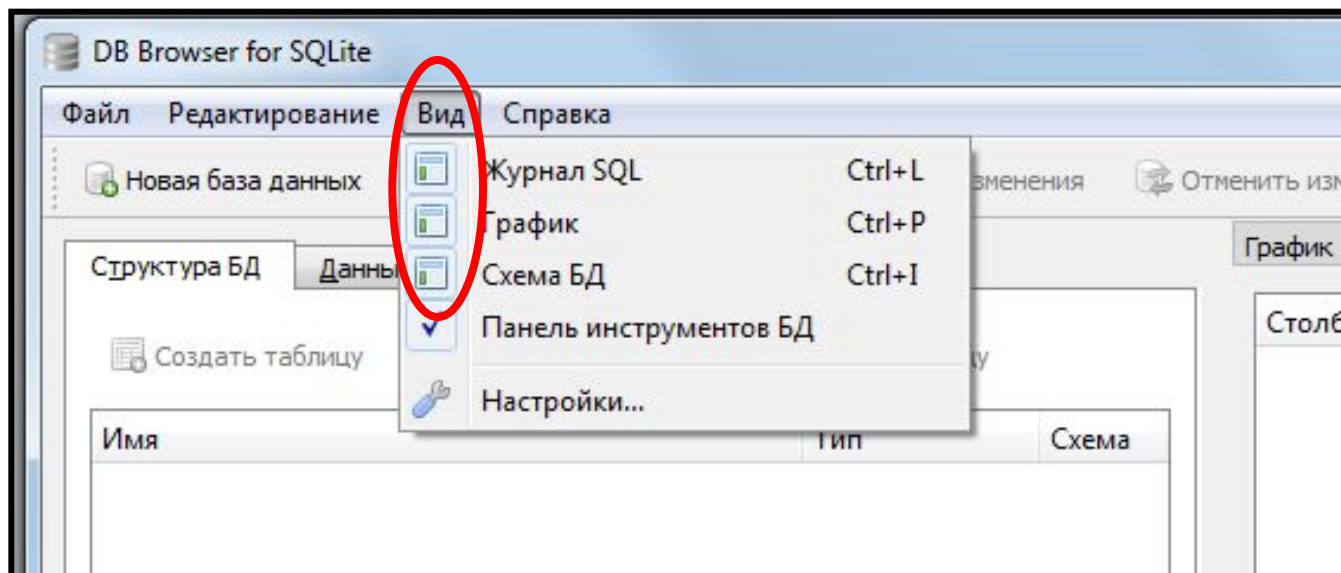


1. Выберите русскую локализацию (Russian)
2. Изменения вступят в действие только после повторного запуска программы !
3. Перезапустите программу

# Локализация программы



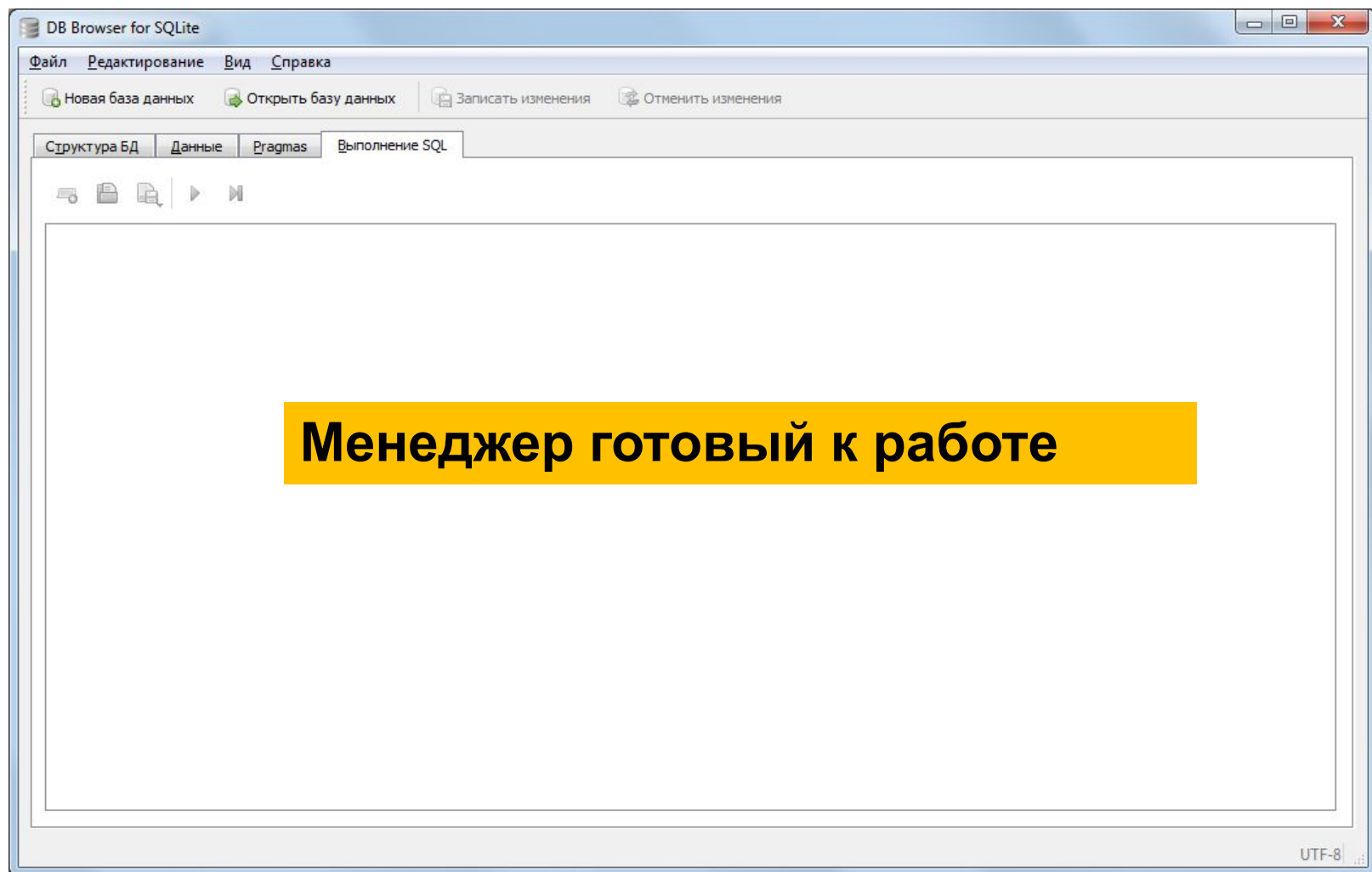
# Настройка интерфейса



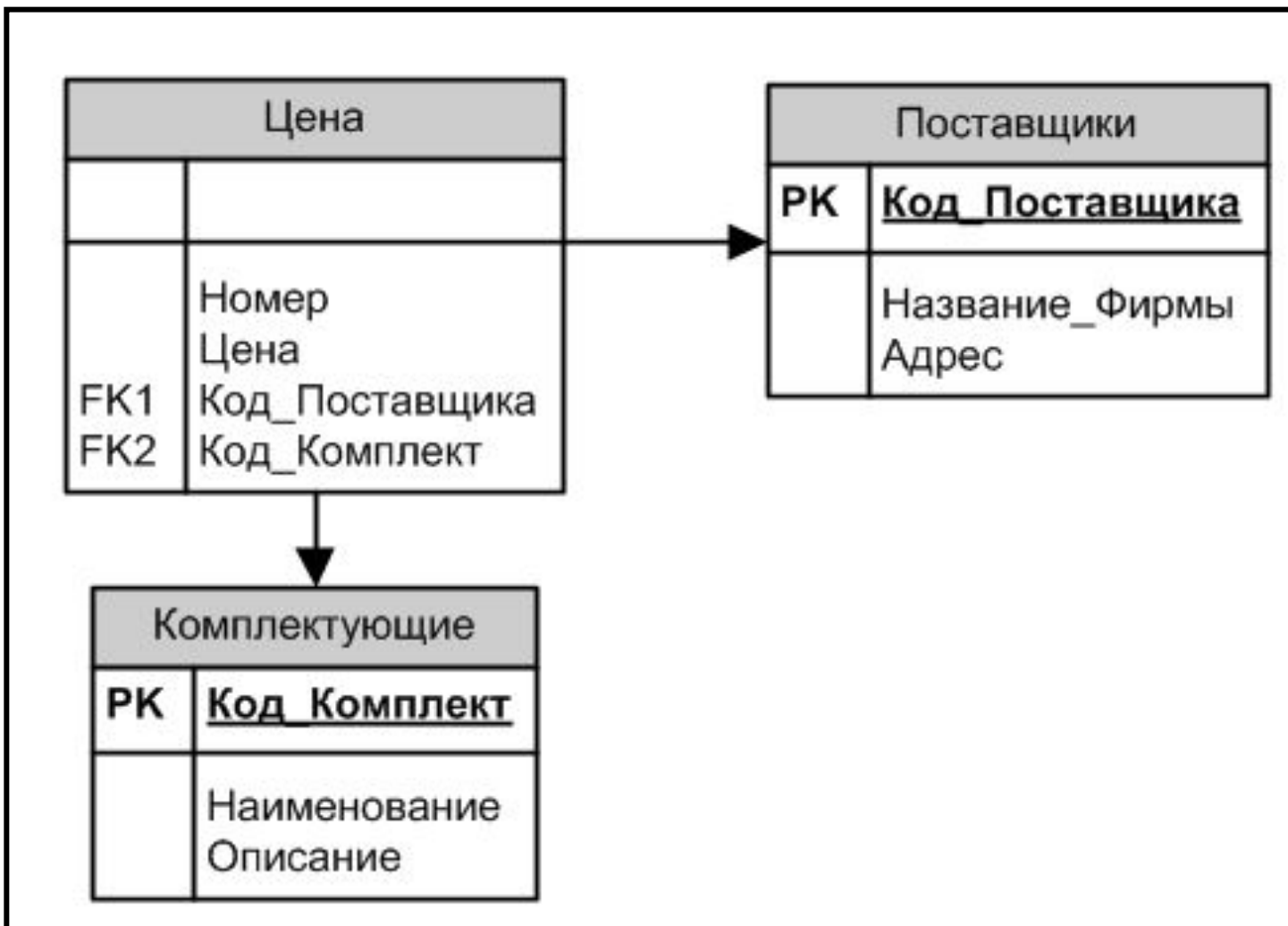
**Закройте окна «Журнал SQL»,  
«График», «Схема БД»**



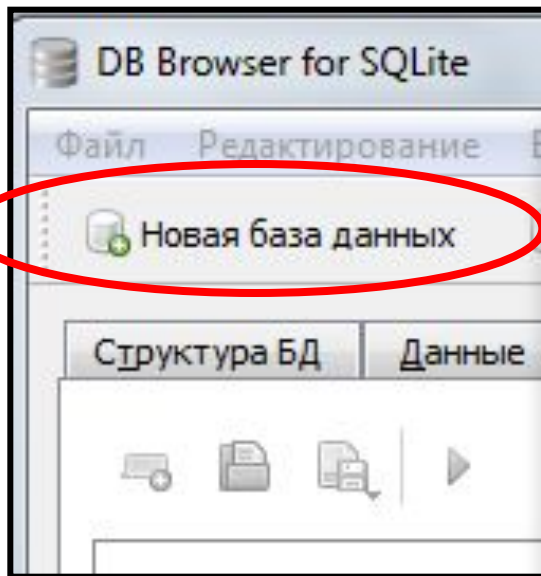
# Настройка интерфейса



# ER модель базы данных

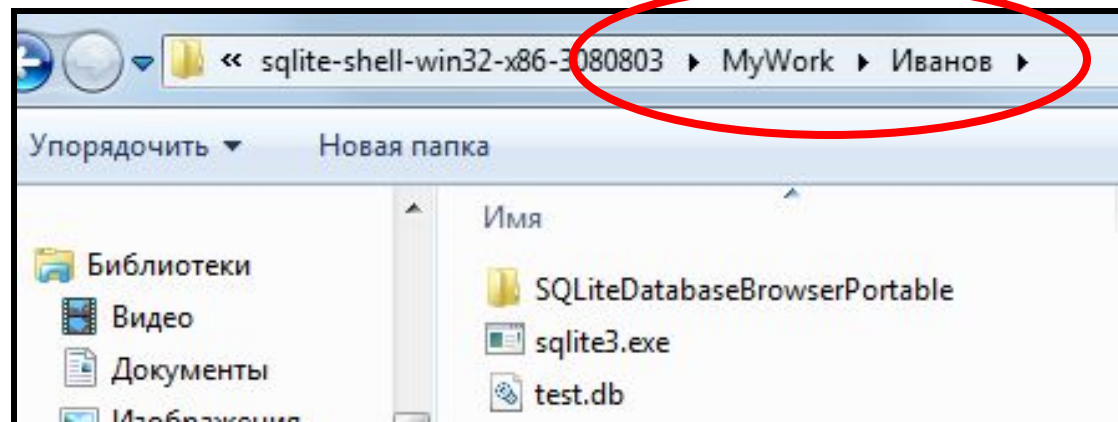


# Создание файла БД



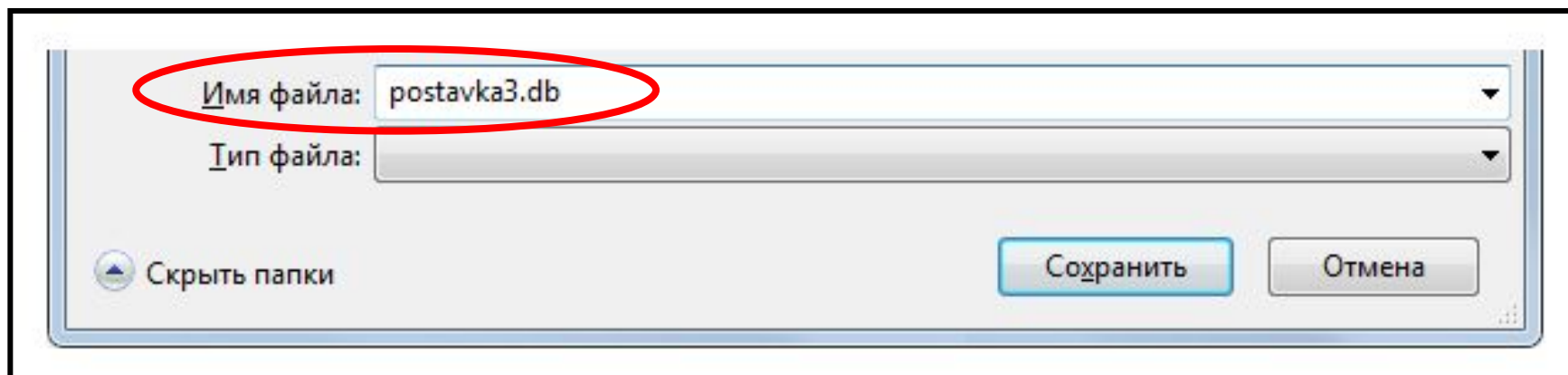
**1. Выполните команду  
«Новая база данных»**

**2. Перейдите в свою рабочую папку  
(см. далее)**



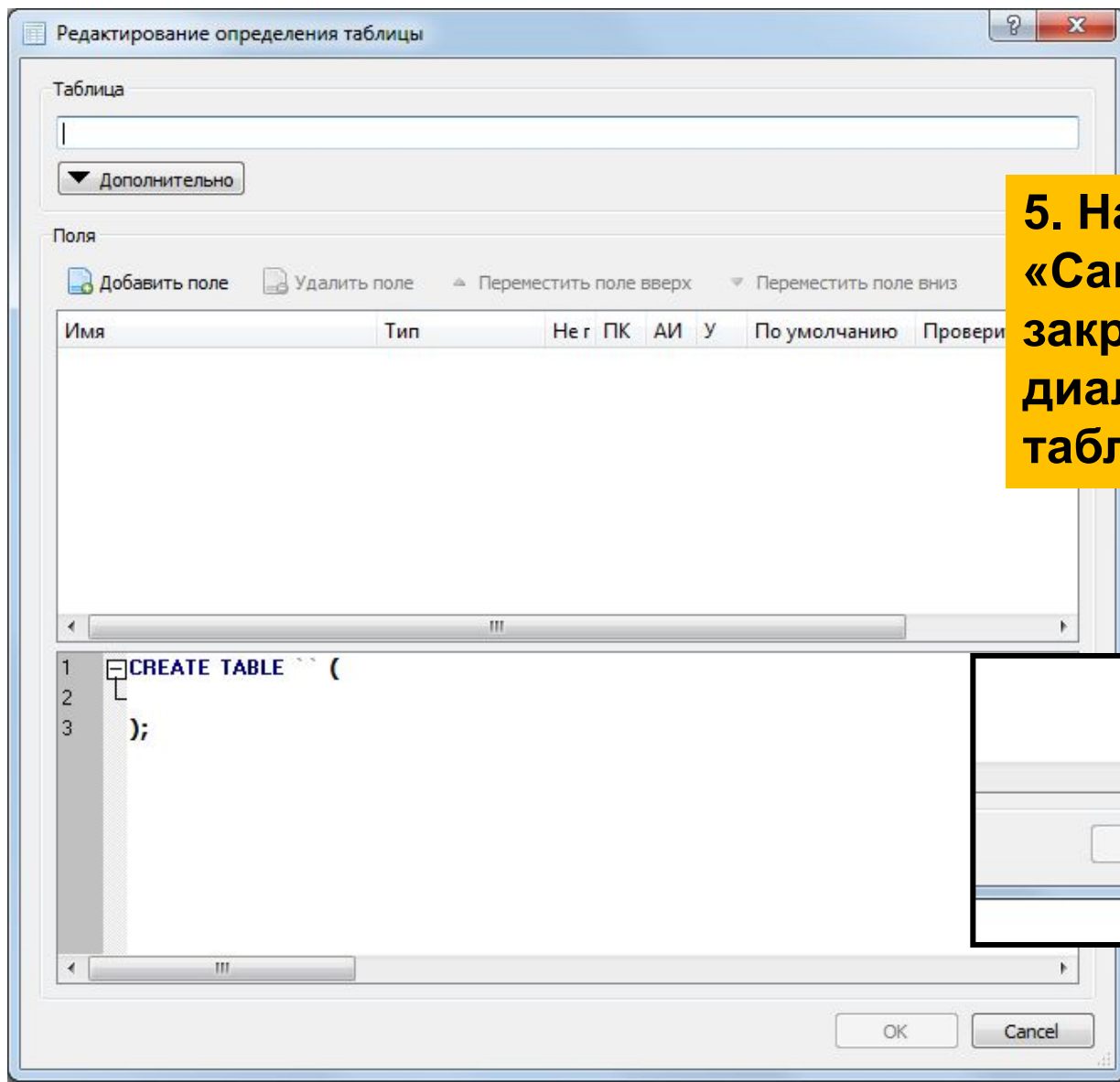
# Создание файла БД

## 3. Введите имя базы данных

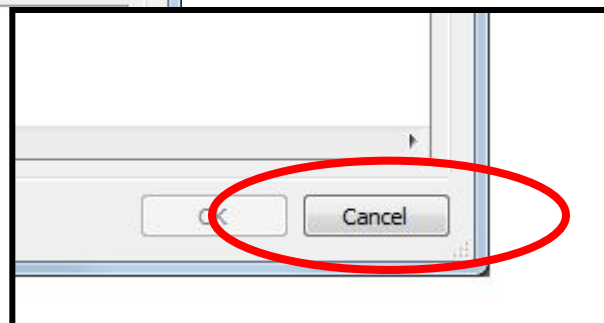


## 4. Нажмите кнопку «Сохранить», см. далее

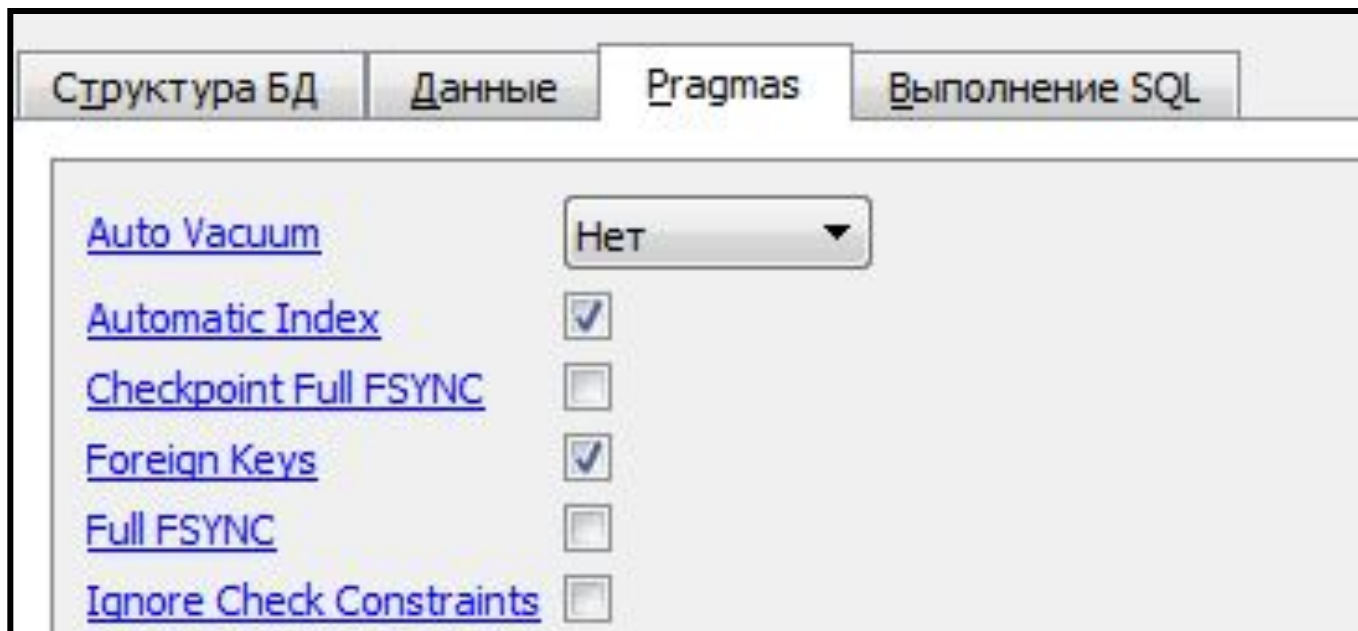
# Создание файла БД



**5. Нажмите кнопку «Cancel» для закрытия окна диалога создания таблиц.**

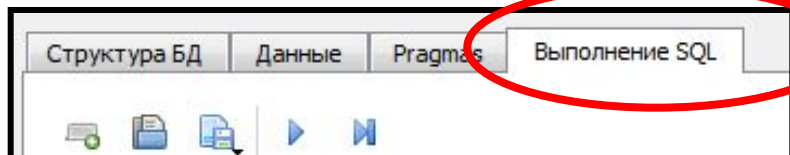


# Внешние ключи



- Откройте окно «Pragmas»
- Установите флажок «Foreign Keys» в активное состояние
- Нажмите кнопку «Save» в окне «Pragmas»

# SQL сценарий создания таблиц

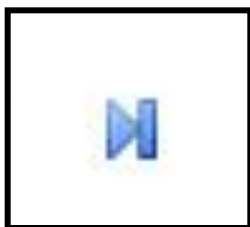


- Выберите окно записи SQL сценария.
- Сохраните сценарий в свой папке

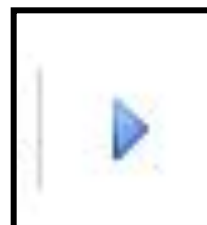


# SQL сценарий создания таблиц

- ❑ Сценарий – текстовый файл, в котором записаны SQL запросы.
- ❑ Обычно такие файлы имеют расширение sql
- ❑ Сценарий может содержать комментарий
  - ❑ Однострочный начинается с пары символов --
  - ❑ Многострочный должен находиться в скобках /\* \*/
- ❑ Запросы сценария можно выполнять все сразу
- ❑ По отдельности
- ❑ Либо выбирают группу – серию запросов



Один запрос



Несколько  
запросов



## Ограничения полей таблиц

Для создания таблиц используют запрос  
**CREATE TABLE**

Для полей можно задать следующие ограничения:

- PRIMARY KEY** – первичный ключ таблицы
- NOT NULL** – запрет наличия неопределенных значений
- COLLATE nocase** – сравнение текстовых данных без учета регистра символов
- DEFAULT значение** – значение по умолчанию, если в поле не были введены данные

# Типы полей таблицы

№	Класс памяти	Описание
1	INTEGER	Данные целого типа
2	REAL	Данные вещественного типа
3	TEXT	Символьные данные
4	BLOB	Binary Large Object Block(Блок байтов)

# Создание таблицы Поставщики



# Создание таблиц

```
crea_db_Postavka.sql ✕  
1 --Создание первой таблицы  
2 CREATE TABLE Postavshiki (Kod_Postavshika INTEGER PRIMARY KEY,  
3 Nazvanie_Firmy TEXT NOT NULL COLLATE nocase,  
4 Adres TEXT NOT NULL DEFAULT 'Москва');
```

- Выделите щелчком мыши первую строку запроса (Строка 2)
- Выполните запрос используя пиктограмму



## Ссылочная целостность

Для создания внешних ключей используют ограничение  
**REFERENCES** **Имя\_Главной\_Таблицы**  
(**Поле\_Главной\_Таблицы**)

Это ограничение дополняется запретом на удаления записей в главной таблице, с которыми связаны записи в зависимой таблице  
**ON DELETE RESTRICT**  
**DEFERRABLE INITIALLY DEFERRED**

## Создание таблиц



- ❑ Введите еще два запроса для создания остальных таблиц.
- ❑ Сохраните и выполните запросы используя пиктограмму панели инструментов

--Создание второй таблицы

```
CREATE TABLE Komplektuishie (Kod_Komplekt INTEGER PRIMARY KEY,  
Naimenovanie TEXT NOT NULL COLLATE nocase,  
Opisanie TEXT NOT NULL DEFAULT 'Отсутствует');
```

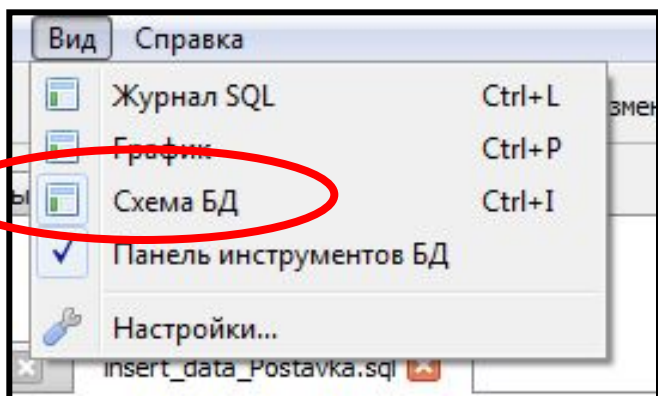
--Создание связующей таблицы

```
CREATE TABLE Cena (Nomer INTEGER, Cena REAL,  
Kod_Postavshika INTEGER REFERENCES Postavshiki(Kod_Postavshika)  
ON DELETE RESTRICT DEFERRABLE INITIALLY DEFERRED,  
Kod_Komplekt INTEGER REFERENCES Komplektuishie(Kod_Komplekt)  
ON DELETE RESTRICT DEFERRABLE INITIALLY DEFERRED  
);
```

# Просмотр структуры БД

Структура БД		
Имя	Тип	Схема
Создать таблицу    Изменить таблицу    Удалить таблицу		
Таблицы (3)		
Cena		CREATE TABLE Cena (Nomer INTEGER, Cena REAL, Kod_Postavshika INTEGER REFERENCES Postavshiki(Kod_Postavshika) ON DELETE RESTRICT DEFERRABLE INITIALLY DEFERRED, Kod_Komplekt INTEGER REFERENCES Kompletuishie(Kod_Komplekt) ON DELETE RESTRICT DEFERRABLE INITIALLY DEFERRED )
Nomer	INTEGER	`Nomer` INTEGER
Cena	REAL	`Cena` REAL
Kod_Postavshika	INTEGER	`Kod_Postavshika` INTEGER
Kod_Komplekt	INTEGER	`Kod_Komplekt` INTEGER
Kompletuishie		CREATE TABLE Kompletuishie (Kod_Komplekt INTEGER PRIMARY KEY, Naimenovanie TEXT NOT NULL COLLATE nocase, Opisanie TEXT NOT NULL DEFAULT 'Отсутствует')
Kod_Komplekt	INTEGER	`Kod_Komplekt` INTEGER
Naimenovanie	TEXT	`Naimenovanie` TEXT NOT NULL
Opisanie	TEXT	`Opisanie` TEXT NOT NULL DEFAULT 'Отсутствует'
Postavshiki		CREATE TABLE Postavshiki (Kod_Postavshika INTEGER PRIMARY KEY, Nazvanie_Firmy TEXT NOT NULL COLLATE nocase, Adres TEXT NOT NULL DEFAULT 'Москва')
Kod_Postavshika	INTEGER	`Kod_Postavshika` INTEGER
Nazvanie_Firmy	TEXT	`Nazvanie_Firmy` TEXT NOT NULL
Adres	TEXT	`Adres` TEXT NOT NULL DEFAULT 'Москва'

# Вывод схемы БД



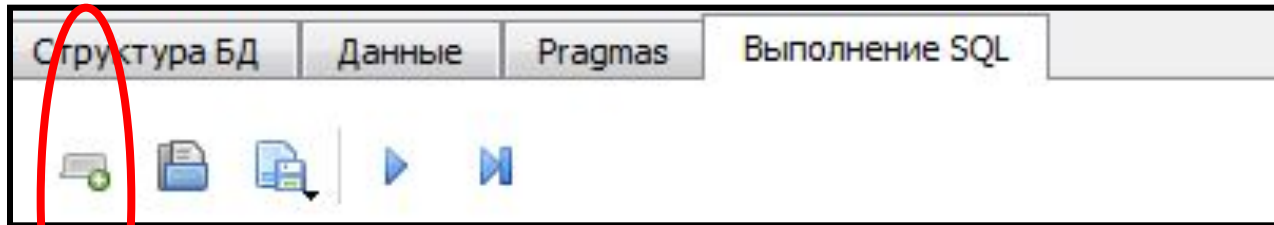
- ❑ Откройте схему данных
- ❑ Раскройте таблицы

Схема БД

Имя	Тип
Таблицы (3)	
Cena	
Komplektuishie	
Kod_Komplekt	INTEGER
Naimenovanie	TEXT
Opisanie	TEXT
Postavshiki	
Kod_Postavshika	INTEGER
Nazvanie_Firmy	TEXT
Adres	TEXT
Индексы (0)	
Представления (0)	
Триггеры (0)	



# SQL Сценарий заполнения таблиц



**1. Откройте еще одно окно для записи SQL сценария**



**2. Сохраните новый сценарий под именем insert\_data\_Postavka в Вашей папке**

## Заполнение таблицы Поставщики

Код_Поставщика	НазваниеФирмы	Адрес
10	Азимут	
20	Алмаз	Питербург
30	Пульсар	Ижевск
40	Квазар	Ростов
50	Гранат	Воронеж

## Заполнение таблицы Поставщики

Введите запросы для добавления записей в таблицу поставщиков в соответствии с таблицей

```
INSERT INTO Postavshiki (Kod_Postavshika,Nazvanie_Firmy) VALUES (10,'Азимут');  
INSERT INTO Postavshiki (Kod_Postavshika,Nazvanie_Firmy,Adres) VALUES (20,'Алмаз','Питербург');  
INSERT INTO Postavshiki (Kod_Postavshika,Nazvanie_Firmy,Adres) VALUES (30,'Пульсар','Ижевск');  
INSERT INTO Postavshiki (Kod_Postavshika,Nazvanie_Firmy,Adres) VALUES (40,'Квазар','Ростов');  
INSERT INTO Postavshiki (Kod_Postavshika,Nazvanie_Firmy,Adres) VALUES (50,'Гранат','Воронеж');
```

- Выделите запросы мышью
- Используя пиктограмму выполните выбранные запросы



# Просмотр содержания таблицы

```
SELECT * FROM Postavshiki;
```

- ❑ Добавьте запрос на извлечение данных
- ❑ Выполните его с помощью пиктограммы

	Kod_Postavshika	Nazvanie_Firmy	Adres
1	10	Азимут	Москва
2	20	Алмаз	Питербург
3	30	Пульсар	Ижевск
4	40	Квазар	Ростов
5	50	Гранат	Воронеж



## Заполнение таблицы Комплектующие

Код_Комплект	Наименование	Описание
1	Изделие1	Комплект
2	Изделие6	
3	Изделие7	
4	Изделие3	Комплект
5	Изделие4	Комплект
6	Изделие5	
7	Изделие2	

# Заполнение таблицы Комплектующие

Введите запросы для добавления записей в таблицу комплектующих в соответствии с таблицей

--Таблица Комплектующие

```
INSERT INTO Komplektuishie (Kod_Komplekt,Naimenovanie,Opisanie) VALUES (1,'Изделие1','Комплект');
INSERT INTO Komplektuishie (Kod_Komplekt,Naimenovanie) VALUES (2,'Изделие6');
INSERT INTO Komplektuishie (Kod_Komplekt,Naimenovanie) VALUES (3,'Изделие7');
INSERT INTO Komplektuishie (Kod_Komplekt,Naimenovanie,Opisanie) VALUES (4,'Изделие3','Комплект');
INSERT INTO Komplektuishie (Kod_Komplekt,Naimenovanie,Opisanie) VALUES (5,'Изделие4','Комплект');
INSERT INTO Komplektuishie (Kod_Komplekt,Naimenovanie) VALUES (6,'Изделие5');
INSERT INTO Komplektuishie (Kod_Komplekt,Naimenovanie) VALUES (7,'Изделие2');
```

- Выделите запросы мышью и используя пиктограмму
- Выполните выбранные запросы

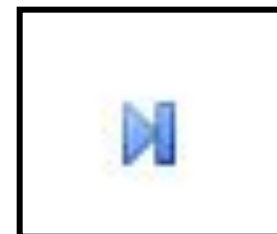


# Просмотр содержания таблицы

```
SELECT * FROM Komplektuishie;
```

Введите запрос на извлечение и выполните его с помощью пиктограммы

	Kod_Komplekt	Naimenovanie	Opisanie
1	1	Изделие1	Комплект
2	2	Изделие6	Отсутствует
3	3	Изделие7	Отсутствует
4	4	Изделие3	Комплект
5	5	Изделие4	Комплект
6	6	Изделие5	Отсутствует
7	7	Изделие2	Отсутствует



## Заполнение таблицы Цена

Номер	Цена	Код_Поставщика	Код_Комплект
1	10000	30	1
2	120000	40	3
3	30000	40	7
4	11000	50	1
5	23000	50	6



# Заполнение таблицы Цена

--Таблица Цена

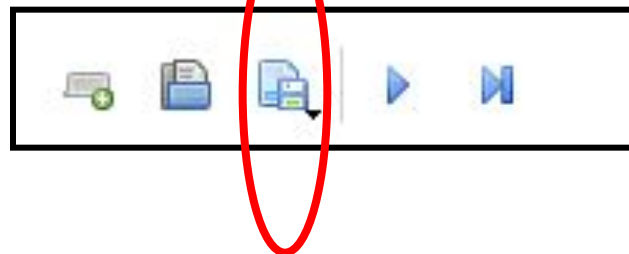
```
INSERT INTO Цена (Nomer,Cena,Kod_Postavshika,Kod_Komplekt) VALUES (1,10000,30,1);
INSERT INTO Цена (Nomer,Cena,Kod_Postavshika,Kod_Komplekt) VALUES (2,120000,40,3);
INSERT INTO Цена (Nomer,Cena,Kod_Postavshika,Kod_Komplekt) VALUES (3,30000,40,7);
INSERT INTO Цена (Nomer,Cena,Kod_Postavshika,Kod_Komplekt) VALUES (4,11000,50,1);
INSERT INTO Цена (Nomer,Cena,Kod_Postavshika,Kod_Komplekt) VALUES (5,23000,50,6);
SELECT * FROM Цена;
```

	Nomer	Cena	Kod_Postavshika	Kod_Komplekt
1	1	10000.0	30	1
2	2	120000.0	40	3
3	3	30000.0	40	7
4	4	11000.0	50	1
5	5	23000.0	50	6

- ❑ Добавьте запросы для заполнения таблицы Цена и просмотра ее содержания.
- ❑ Выполните запросы.

# Проверка ссылочной целостности

Сохраните запросы SQL  
сценария используя  
пиктограмму

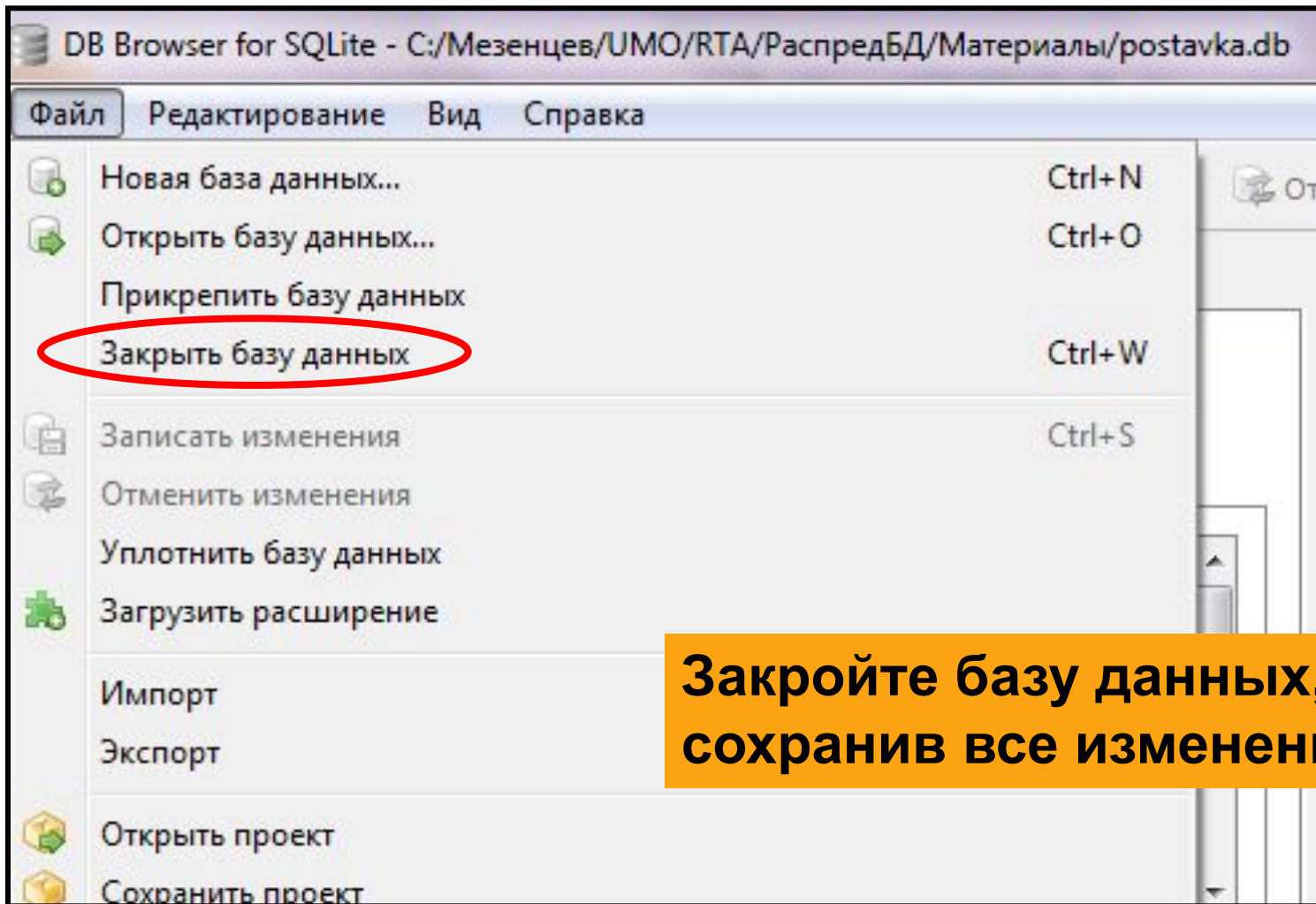


```
DELETE FROM Postavshiki;  
DELETE FROM Komplektuishie;
```

- Добавьте два запроса на удаление записей из главных таблиц
- Выполните запросы

**Запросы должны быть отклонены !!!**

```
FOREIGN KEY constraint failed: DELETE FROM Postavshiki;
```



**Закройте базу данных,  
сохранив все изменения**