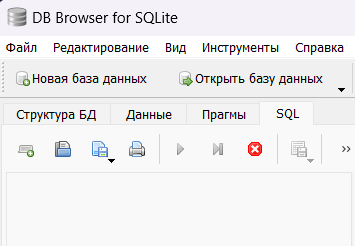
**Создание таблиц на примере SQLite**

Чтобы создать таблицу в графическом клиенте «DB Browser for SQLite», нужно перейти во вкладку SQL, как указано на Рисунке 1.

Рисунок 1. Вкладка SQL



Далее, покажем на примере Buyer создание таблицы:

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8 | **CREATE** **TABLE** "buyer" (  "id" INTEGER **NOT** **NULL** **UNIQUE**,  "name" TEXT **NOT** **NULL**,  "phone" INTEGER **NOT** **NULL**,  **PRIMARY** **KEY**("id" AUTOINCREMENT),  **CONSTRAINT** phone\_range  **CHECK**("phone" > 0)  ); |

Для создания таблицы используется команда «CREATE TABLE», после которой следует название таблицы, а в скобках заключено объявление её столбцов с указанием типов данных, свойств и ограничений.

Основные виды ограничений:

* NOT NULL – столбец обязателен к заполнению;
* UNIQUE – элемент должен быть уникален в рамках столбца;
* CHECK – устанавливает условия, без выполнения которых заполнение невозможно;
* DEFAULT – значение по умолчанию;
* AUTOINCREMENT – автоматически увеличивает значение на единицу при создании новой записи;

Чтобы объявить ограничение, можно использовать ключевое слово CONSTRAINT:

|  |  |
| --- | --- |
| 1  2 | **CONSTRAINT** название\_ограничения  **CHECK**("название\_атрибута" > 0) |

У каждой таблицы должен быть первичный ключ. Чтобы его объявить, используем команду PRIMARY KEY:

|  |  |
| --- | --- |
| 1  2 | "id" INTEGER **NOT** **NULL** **UNIQUE**,  **PRIMARY** **KEY**("id" AUTOINCREMENT) |

Иногда таблица связана с другими таблицами внешними ключами. Для создания внешних ключей используем команду FOREIGN KEY:

|  |  |
| --- | --- |
| 1  2 | "buyer\_id" INTEGER **NOT** **NULL**,  **FOREIGN** **KEY**("buyer\_id") **REFERENCES** "buyer"("id") |

Если появляется необходимость удалить таблицу, то используется команда DROP TABLE:

|  |  |
| --- | --- |
| 1 | **DROP** **TABLE** название\_таблицы; |

Для того, чтобы внести изменения в существующую таблицу, используется выражение ALTER TABLE:

|  |  |
| --- | --- |
| 1  2  3  4  5  6 | **ALTER** **TABLE** buyer  **RENAME** **TO** shopper; *-- переименование таблицы*  *-- Далее строка ALTER TABLE опускается*  **ADD** **COLUMN** status TEXT **NOT** **NULL**; *-- добавление столбца Status*  **DROP** **COLUMN** quantity; *-- удаление столбца*  **RENAME** **COLUMN** name **TO** fio; *-- переименование столбца* |

В базе данных «Shop» мы создали 7 таблиц:

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36  37  38  39  40  41  42  43  44  45  46  47  48  49  50  51  52  53  54  55  56  57  58  59  60  61 | **CREATE** **TABLE** "buyer" (  "id" INTEGER **NOT** **NULL** **UNIQUE**,  "name" TEXT **NOT** **NULL**,  "phone" INTEGER **NOT** **NULL**,  **PRIMARY** **KEY**("id" AUTOINCREMENT),  **CONSTRAINT** phone\_range  **CHECK**("phone" > 0)  );  **CREATE** **TABLE** "store" (  "id" INTEGER **NOT** **NULL** **UNIQUE**,  "city" TEXT **NOT** **NULL**,  "address" TEXT **NOT** **NULL**,  **PRIMARY** **KEY**("id" AUTOINCREMENT)  );  **CREATE** **TABLE** "category" (  "id" INTEGER **NOT** **NULL** **UNIQUE**,  "name" TEXT **NOT** **NULL**,  **PRIMARY** **KEY**("id" AUTOINCREMENT)  );  **CREATE** **TABLE** "producer" (  "id" INTEGER **NOT** **NULL** **UNIQUE**,  "name" TEXT **NOT** **NULL**,  **PRIMARY** **KEY**("id" AUTOINCREMENT)  );  **CREATE** **TABLE** "order" (  "id" INTEGER **NOT** **NULL** **UNIQUE**,  "date" TEXT **NOT** **NULL**,  "status" TEXT **NOT** **NULL**,  "buyer\_id" INTEGER **NOT** **NULL**,  **PRIMARY** **KEY**("id" AUTOINCREMENT),  **FOREIGN** **KEY**("buyer\_id") **REFERENCES** "buyer"("id")  );  **CREATE** **TABLE** "product" (  "id" INTEGER **NOT** **NULL** **UNIQUE**,  "name" TEXT **NOT** **NULL**,  "price" REAL **NOT** **NULL**,  "quantity" INTEGER **NOT** **NULL**,  "availability" BLOB **NOT** **NULL**,  "date" TEXT **NOT** **NULL**,  "producer\_id" INTEGER **NOT** **NULL**,  "category\_id" INTEGER **NOT** **NULL**,  "store\_id" INTEGER **NOT** **NULL**,  **PRIMARY** **KEY**("id" AUTOINCREMENT),  **FOREIGN** **KEY**("producer\_id") **REFERENCES** "producer"("id"),  **FOREIGN** **KEY**("category\_id") **REFERENCES** "category"("id"),  **FOREIGN** **KEY**("store\_id") **REFERENCES** "store"("id")  );  **CREATE** **TABLE** "order\_product" (  "order\_id" INTEGER **NOT** **NULL**,  "product\_id" INTEGER **NOT** **NULL**,  "quantity" INTEGER **NOT** **NULL**,  **PRIMARY** **KEY**("product\_id","order\_id"),  **FOREIGN** **KEY**("product\_id") **REFERENCES** "product"("id"),  **FOREIGN** **KEY**("order\_id") **REFERENCES** "order"("id")  ); |

**Домашнее задание:** создать базу данных «Shop», в которой будут использованы те же таблицы, что и выше, используя СУБД PostgreSQL.