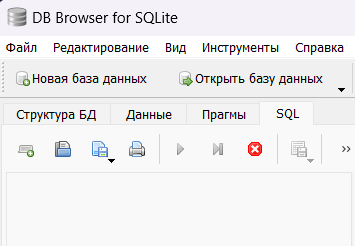
**Создание таблиц на примере SQLite**

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**,  "FIO" TEXT **NOT** **NULL**,  "PhoneNumber" INTEGER **NOT** **NULL**,  **PRIMARY** **KEY**("ID" AUTOINCREMENT),  **CONSTRAINT** PhoneNumber\_range  **CHECK**("PhoneNumber" > 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 | "BuyerID" INTEGER **NOT** **NULL**,  **FOREIGN** **KEY**("BuyerID") **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** FIO **TO** Name; *-- переименование столбца* |

В базе данных «Shop» мы создали 5 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 | **CREATE** **TABLE** "Buyer" (  "ID" INTEGER **NOT** **NULL** **UNIQUE**,  "FIO" TEXT **NOT** **NULL**,  "PhoneNumber" INTEGER **NOT** **NULL**,  **PRIMARY** **KEY**("ID" AUTOINCREMENT),  **CONSTRAINT** PhoneNumber\_range  **CHECK**("PhoneNumber" > 0)  );  **CREATE** **TABLE** "Shop" (  "ID" INTEGER **NOT** **NULL** **UNIQUE**,  "Location" TEXT **NOT** **NULL**,  **PRIMARY** **KEY**("ID" AUTOINCREMENT)  );  **CREATE** **TABLE** "Orderr" (  "ID" INTEGER **NOT** **NULL** **UNIQUE**,  "Quantity" INTEGER **NOT** **NULL**,  "BuyerID" INTEGER **NOT** **NULL**,  "Date" TEXT **NOT** **NULL**,  "Status" TEXT **NOT** **NULL**,  **PRIMARY** **KEY**("ID" AUTOINCREMENT),  **FOREIGN** **KEY**("BuyerID") **REFERENCES** "Buyer"("ID")  );  **CREATE** **TABLE** "Product" (  "ID" INTEGER **NOT** **NULL** **UNIQUE**,  "ShopID" INTEGER **NOT** **NULL**,  "UnitOfMessure" TEXT **NOT** **NULL**,  "Price" REAL **NOT** **NULL**,  "Quantity" INTEGER **NOT** **NULL**,  "Availability" BLOB **NOT** **NULL**,  "Date" TEXT **NOT** **NULL**,  "Name" TEXT **NOT** **NULL**,  **PRIMARY** **KEY**("ID" AUTOINCREMENT),  **FOREIGN** **KEY**("ShopID") **REFERENCES** "Shop"("ID")  );  **CREATE** **TABLE** "OrderProduct" (  "OrderID" INTEGER **NOT** **NULL**,  "ProductID" INTEGER **NOT** **NULL**,  **PRIMARY** **KEY**("ProductID","OrderID"),  **FOREIGN** **KEY**("ProductID") **REFERENCES** "Product"("ID"),  **FOREIGN** **KEY**("OrderID") **REFERENCES** "Orderr"("ID")  ); |

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