1. Настроить работу SQLite и PostgreSQL, создать базу и хотя бы одну таблицу с данными в каждой из них (через IDE или через SQL).  
   В postgres можно создать нового пользователя (не postgres) и дать ему доступ к вашей новой базе, либо установить пароль юзеру postgres для подключения. Надо гуглить, развлекаться в консоли. Проще гуглить конкретные цели, чем читать всю документацию. Последующие задания делаете в любой базе на выбор, но с SQLite будет больше проблем в плане запросов, а вот настроить сложнее PostgreSQL.

**Все последующие задачи можно сделать в виде чистых SQL запросов в консоли, но немного полезнее будет попробовать через sqlite / psycopg в питоне.**

1. Написать функцию **task\_2,** которая принимает путь к sqlite или ссылку к postgres базе данных, создать таблицы  
   **Shops**:
   * 1. id - INTEGER PRIMARY
     2. name - VARCHAR (название магазина)
     3. address - VARCHAR NULLABLE (адрес магазина)
     4. staff\_amount - INTEGER (количество сотрудников)

**Departments**:

* + 1. id - INTEGER PRIMARY
    2. sphere - VARCHAR (направление работы отдела - например, одежда, игрушки, посуда и т.д.)
    3. staff\_amount - INTEGER (количество сотрудников)
    4. shop - FOREIGN KEY на Shops.id (в каком магазине отдел)

**Items:**

* + 1. id - INTEGER PRIMARY
    2. name - VARCHAR (название товара)
    3. description - TEXT NULLABLE (описание)
    4. price - INTEGER (цена)
    5. department - FOREIGN KEY на Department.id (в каком отделе продается)

1. Получаем в функцию **task\_3** путь к базе данных sqlite / ссылку на postgres. С помощью INSERT или дополнительной возможности модуля sqlite/psycopg заполнить таблицы данными (одной командой пачку строк).  
     
   Shops

|  |  |  |  |
| --- | --- | --- | --- |
| id | name | address | staff\_amount |
| 1 | Auchan | <null> | 250 |
| 2 | IKEA | Street Žirnių g. 56, Vilnius, Lithuania. | 500 |

Departments

|  |  |  |  |
| --- | --- | --- | --- |
| id | sphere | staff\_amount | shop |
| 1 | Furniture | 250 | 1 |
| 2 | Furniture | 300 | 2 |
| 3 | Dishes | 200 | 2 |

Items

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| id | name | description | price | department |
| 1 | Table | Cheap wooden table | 300 | 1 |
| 2 | Table | <null> | 750 | 2 |
| 3 | Bed | Amazing wooden bed | 1200 | 2 |
| 4 | Cup | <null> | 10 | 3 |
| 5 | Plate | Glass plate | 20 | 3 |

1. Получаем в функцию task\_4 путь к базе sqlite / ссылку на postgresql. Вернуть лист ответов на каждый запрос:
   1. Получить все поля по товарам, у которых есть описание.
   2. Получить все направления отделов, в которых более 200 сотрудников. Избегать повторений.
   3. Получить все адреса магазинов с названием, начинающихся на английскую букву “i” без учета регистра (для Postgres используйте ILIKE).
   4. Получить все названия товаров, которые продаются в отделах с мебелью (sphere == Furniture).
   5. Получить названия магазинов, где в продаже есть товары с описанием.  
      Избегать повторений.
   6. Получить для каждого товара все его поля (кроме id) + все поля его отдела (кроме id), причем для всех полей отдела в ответе должна быть приписка department\_{название\_поля} + все поля его магазина (кроме id) с припиской shop\_{название\_поля}.
   7. Вернуть 3 - 4 по счету строки товаров из выборки, отсортированной по имени товара.
   8. Выбрать названия товаров и названия их отделов (у товара обязательно должен быть отдел и наоборот).
   9. Выбрать названия товаров и названия их отделов. Если отдела не существует, то в его поле должен быть NULL.
   10. Выбрать названия товаров и названия их отделов. Если в каком-то отделе нет товаров, то он должен попасть в ответ, а в колонке названия товара должен быть NULL.  
       В SQLite нет RIGHT JOIN. Если делаете в SQLite, то просто используйте обратный порядок присоединения таблиц с LEFT JOIN.
   11. Выбрать все возможные сочетания связанных или “одиноких” названий товаров и названий отделов (FULL JOIN). В SQLite нет FULL JOIN. Если делаете в SQLite, то используйте UNION (объединение двух результатов запросов) и обратный порядок присоединения таблиц с LEFT JOIN.
   12. Выбрать все возможные сочетания названий товаров и названий отделов независимо от связей (наш любимый вид JOIN).
   13. Выбрать количество товаров, сумму цен, максимальную цену, минимальную цену, среднюю цену для каждого магазина, где количество товаров больше одного.
   14. Выбрать названия магазинов и массивы всех цен его товаров. Задание только для PostgreSQL.
2. Получаем в функцию task\_5 путь к базе sqlite / ссылку на postgres. Удалить все товары, у которых цена больше 500 и у которых нет описания.
3. Получаем в функцию task\_6 путь к базе sqlite / ссылку на postgres. Удалить все товары, у которых магазин не имеет адреса.