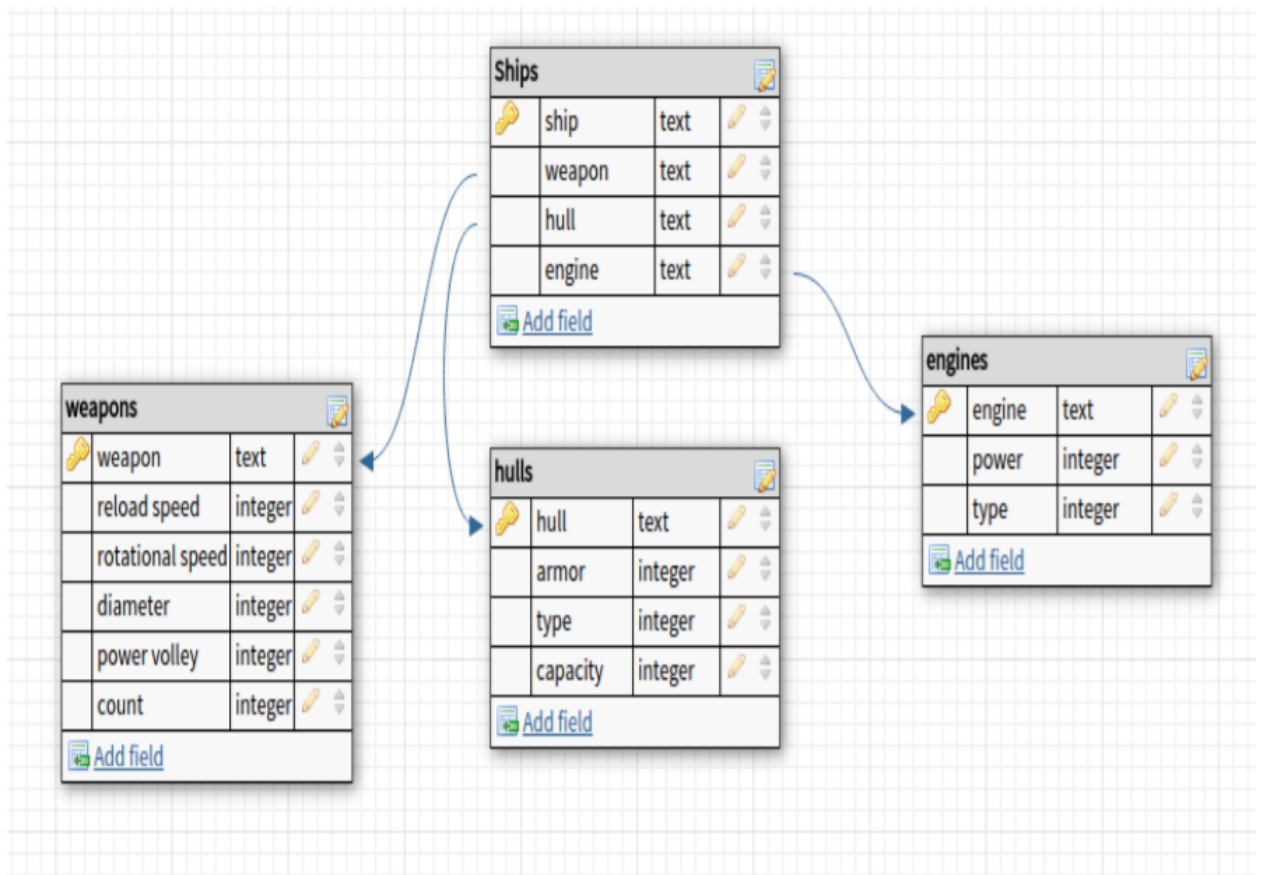


Тестовое задание на специалиста по автоматизации тестирования

Задание призвано оценить работу с Python, уровень владения культурой программирования. Показать кандидату с чем ему придется столкнуться. Несмотря на это, данные в задании имеют очень упрощенную схему относительно реальных.

Цели задания

1. Написать python-скрипт, создающий SQLite базу по указанной схеме.
Primary key – текстовое поле weapon / ship / hull / engine соответственно.



2. Создать скрипт, который будет рандомно заполнять значения в созданной базе. Вполне сгодятся названия: *Ship-1, Ship-2, Weapon-1* и т. д.

Количество записей для каждой таблицы:

ships: **200**

weapons: **20**

hulls: **5**

engines: **6**

Диапазон значений для целочисленных параметров: **1-20**

3. Создать session-score фикстуру, которая получает текущее состояние базы данных и создает временную новую базу, в которой рандомизируются значения:
 - a. Для каждого корабля меняется на случайный **один** из компонентов: корпус, оружие или двигатель
 - b. В каждом компоненте меняется **один** из случайно выбранных параметров на случайное значение из допустимого диапазона (см. выше)
4. Написать автотесты, сравнивающие данные из исходной базы с полученной рандомизированной:
 - a. Для каждого корабля должно быть 3 теста, проверяющие его оружие, корпус и двигатель.
 - b. Тест должен падать с **assert**:
 - i. Когда значение параметра компонента не соответствует тому, что было до запуска рандомизатора.

Пример вывода:

Ship-2, weapon-1

reload speed: expected 1, was 2

diameter: expected 2, was 3

Ship-2, hull-3

type: expected 1, was 2

Ship-3, engine-6

power: expected 22, was 13

- ii. Когда изменилось оружие, корпус или двигатель. Выводить, что было до этого и, что сейчас.

Пример вывода:

Ship-5, engine-4

expected engine-1, was engine-4

Требования к выполненному заданию

- Версия Python – 3.8
- Тесты должны быть написаны с использованием фреймворка **pytest**
- В качестве параметризации использовать `pytest.mark.parametrize` или хук `pytest_generate_tests`.
- В результате прогона должно получиться **600** тестов.
- В результате выполнения задания должно быть по крайней мере следующее:
 - Скрипт, создающий и заполняющий исходную базу данных
 - Python-модуль, содержащий тесты
 - (Опционально) `conftest.py`, содержащий фикстуры и хуки
- Стиль кода – PEP8.