

Haskell Project Report

Group 23

Junyu Lin, 210470873

Yuzhao Cao, 190401036

A stack-based Haskell app using sqlite database is implemented by our group which aims to harvest information from the Web, save data on the sqlite database and interact . The topic is monster manual of Dungeons & Dragons e5, a game and the website of JSON data selected is <https://github.com/mdogdope/DND-5E-Data/blob/4fc4efa217ce9c31a6c7bb4.93cc95580a7e065dc/monsters.json>.

The five main Haskell files that make up the haskell-project are Types.hs, Fetch.hs, Parse.hs, Database.hs, and Main.hs. Besides, database file is monster.sqlite and packages are manually referenced in file haskell-project.cabal including aeson , bytestring, http-conduit and sqlite-simple.

1. Data load

Dataset type is Json. The function of this part is realized by Fetch.hs and Parse.hs.

Fetch.hs is used to download Web data by function "download url". In Parse.hs module "parseMonsters" parses downloaded data into Haskell datatype and also Json string is converted to data of instance type "Monster" and "Monsterone" which two types are created by Types.hs by "renameFields" and "customOptions" with Aeson "FromJSON".

Part of data as example from url:

```
{ "monster": [
  {
    "name": "Aarakocra",
    "size": "M",
    "type": "humanoid (aarakocra), monster manual",
    "alignment": "neutral good",
    "ac": "12",
    "hp": "13 (3d8)",
    "speed": "20 ft., fly 50 ft.",
    "str": "10",
    "dex": "14",
    "con": "10",
    "int": "11",
```

```
        "wis": "12",
        "cha": "11"
        ...
    },
    ...
]
```

2. Haskell data type definition

Types.hs defines all the Haskell data types including "Entry", "Strtable_", "Monsterone" and "Monster" that we will be using for this project.

"Monster" and "Monsterone" derive "Show" and "Generic" for subsequent operation. "Entry" and "Strtable_" are created for table "entries" and "strs" in database monster.sqlite. Among them, there are two attributes called "id" and "fk_str" which are used to connect with table "entries" and "strs".

3. DB table

Database.hs creates database table, saves and retrieves data from and to a database using again the appropriate Haskell data types created by Types.hs.

Tables from monster.sqlite:

```
sqlite> .schema
CREATE TABLE strs (id INTEGER PRIMARY KEY AUTOINCREMENT, name
VARCHAR(100) NOT NULL, size VARCHAR(10) NOT NULL, monstertype VARCHAR(80)
NOT NULL, str VARCHAR(10) NOT NULL);

CREATE TABLE sqlite_sequence(name, seq);
CREATE TABLE entries (alignment VARCHAR(40) NOT NULL, ac VARCHAR(40) NOT
NULL, hp VARCHAR(40) NOT NULL, fk_str INTEGER);
```

In addition, Database.hs also provides extra features which allow the user to filter out the information from the data. "queryStrAllEntries" fetches all monsters which have the strength value specified by the user.

Use this sqlite statement to filter and connect the two tables through the unique serial number in the two tables and finally output these attributes:

```
SELECT name, size, monstertype, alignment, ac, hp, str FROM entries
inner join strs on entries.fk_str == strs.id WHERE alignment=?
```

"queryAlignmentAllEntries" fetches monsters which have the same alignment that is been specified.

"querySizeAllEntries" fetches monsters which have the same size that is been specified.

"queryStrTotalEntries" Counts the total length of data that is been produces from "queryStrAllEntries".

Haskell core for count function:

```
let total = length (map str strEntries)
```

4. App

Main.hs provides a interactive design of which the user can get access to create and initialise an sqlite database, download data and save to database, run queries on the database. In able to run queries on the database, the user have to first download data and save to database.

Function menu:

```
-----  
Welcome to Dungeons & Dragons 5e monster book  
(1) Download data  
(2) All monsters by strength(str) value  
(3) total monsters by strength(str) value  
(4) Monsters with same alignment  
(5) All monsters by size  
(6) Quit  
-----
```

Examples of running results:

```
-----  
Welcome to Dungeons & Dragons 5e monster book  
(1) Download data  
(2) All monsters by strength(str) value  
(3) total monsters by strength(str) value  
(4) Monsters with same alignment  
(5) All monsters by size  
(6) Quit  
-----  
Choose an option > 3  
Enter your favorite str value [e.g. 1-30] > 29  
Looking for strength value 29 entries...  
"Total entries: 80"
```