Gabriel LEBIS : Devoir 1 BDD réparties

Exercice 1

Le crawler est fait en Python avec **scapy**
Le site crawlé est d20pfsrd.com car mon crawler a été banni du site DXContent

Lien du code :
https://github.com/G4bleb/distributed-databases/blob/master/Devoir1/crawler/crawler.py

Exemple de sort parsé :

```
{
    "name": "Resistance",
    "school": "abjuration",
    "level": {
        "bard": 0,
        "cleric/oracle": 0,
        "druid": 0,
        "inquisitor": 0,
        "paladin": 1,
        "psychic": 0,
        "shaman": 0,
        "sorcerer/wizard": 0,
        "summoner/unchained summoner": 0,
        "witch": 0
    },
    "casting_time": "1 standard action",
    "components": [
        "V",
        "S",
        "M/DF (a miniature cloak)"
    ],
    "target/effect/area": "creature touched",
    "duration": "1 minute",
    "spell_resistance": true
},
```

Après avoir importé le json dans une base mongoDB avec mongoimport, on fait le map reduce :

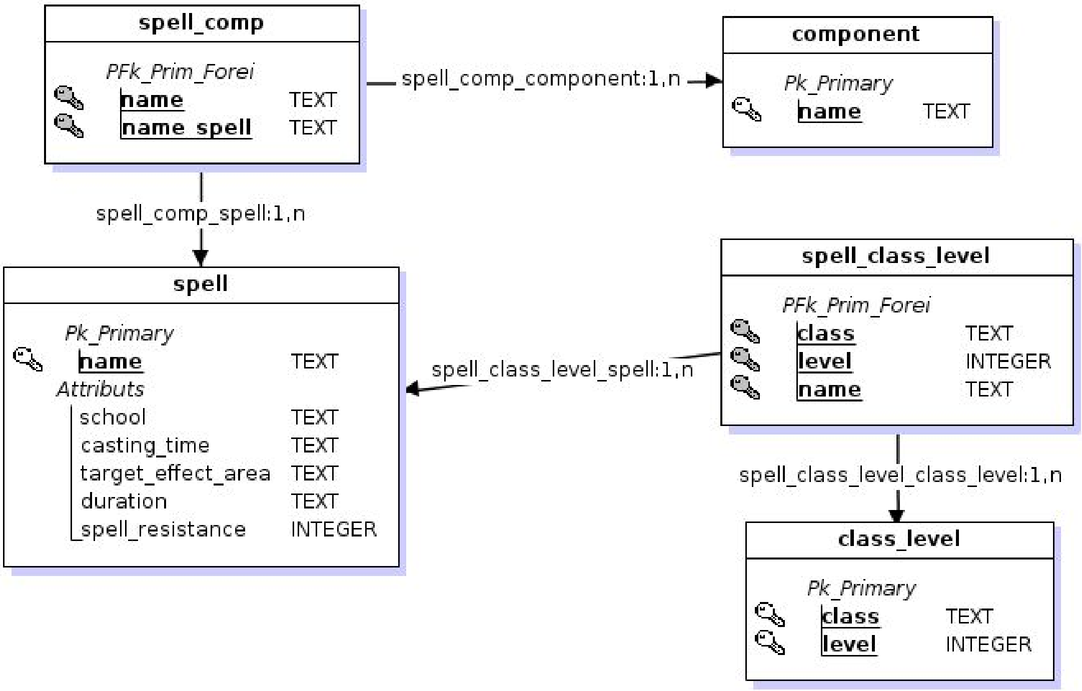
```
db.spells.mapReduce(
    function () { emit(this.name, 1); },
    function (key, values) {
        return values;
    },
    {
        query: { $and: [{ "level.sorcerer/wizard": { $lte: 4 } }, {
components: ["V"] }] },
        out: "usable_spells"
    }
)
```

```
> db.usable_spells.find()
{ "_id" : "Anti-Summoning Shield", "value" : 1 }
{ "_id" : "Anywhere But Here", "value" : 1 }
{ "_id" : "Blindness-Deafness", "value" : 1 }
{ "_id" : "Blur", "value" : 1 }
{ "_id" : "Buoyancy", "value" : 1 }
{ "_id" : "Desperate Weapon", "value" : 1 }
{ "_id" : "Dimension Door", "value" : 1 }
{ "_id" : "Feather Fall", "value" : 1 }
{ "_id" : "Flare", "value" : 1 }
{ "_id" : "Flare Burst", "value" : 1 }
{ "_id" : "Fool's Teleport", "value" : 1 }
{ "_id" : "Knock", "value" : 1 }
{ "_id" : "Liberating Command", "value" : 1 }
{ "_id" : "Mindlink", "value" : 1 }
{ "_id" : "Raven's Flight", "value" : 1 }
{ "_id" : "Shout", "value" : 1 }
{ "_id" : "Silent Table", "value" : 1 }
{ "_id" : "Steal Voice", "value" : 1 }
{ "_id" : "Touch of Blindness", "value" : 1 }
{ "_id" : "Wave Shield", "value" : 1 }
```

Version SQLite

La base est conçue avec JMerise, qui convertit le MCD en script SQLite.

MLD de la base :



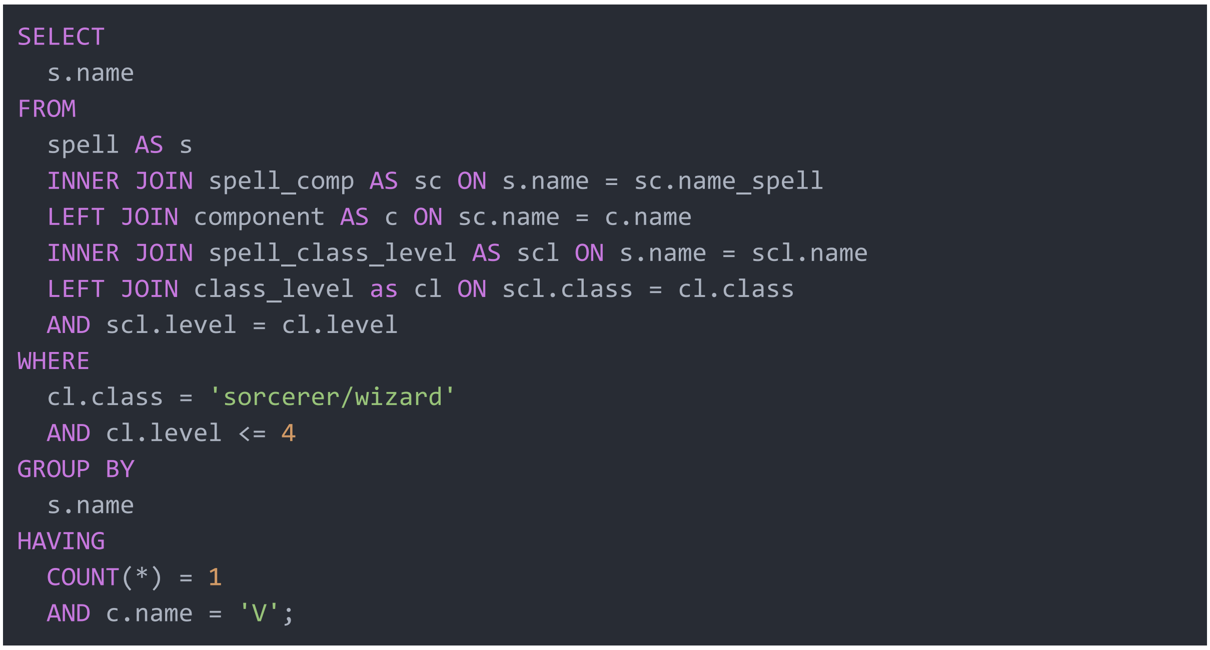
Lien du script de création des tables :

https://github.com/G4bleb/distributed-databases/blob/master/Devoir1/sqlite/create.sql

Avec un script Python, on change le json en script SQL d'insertion.

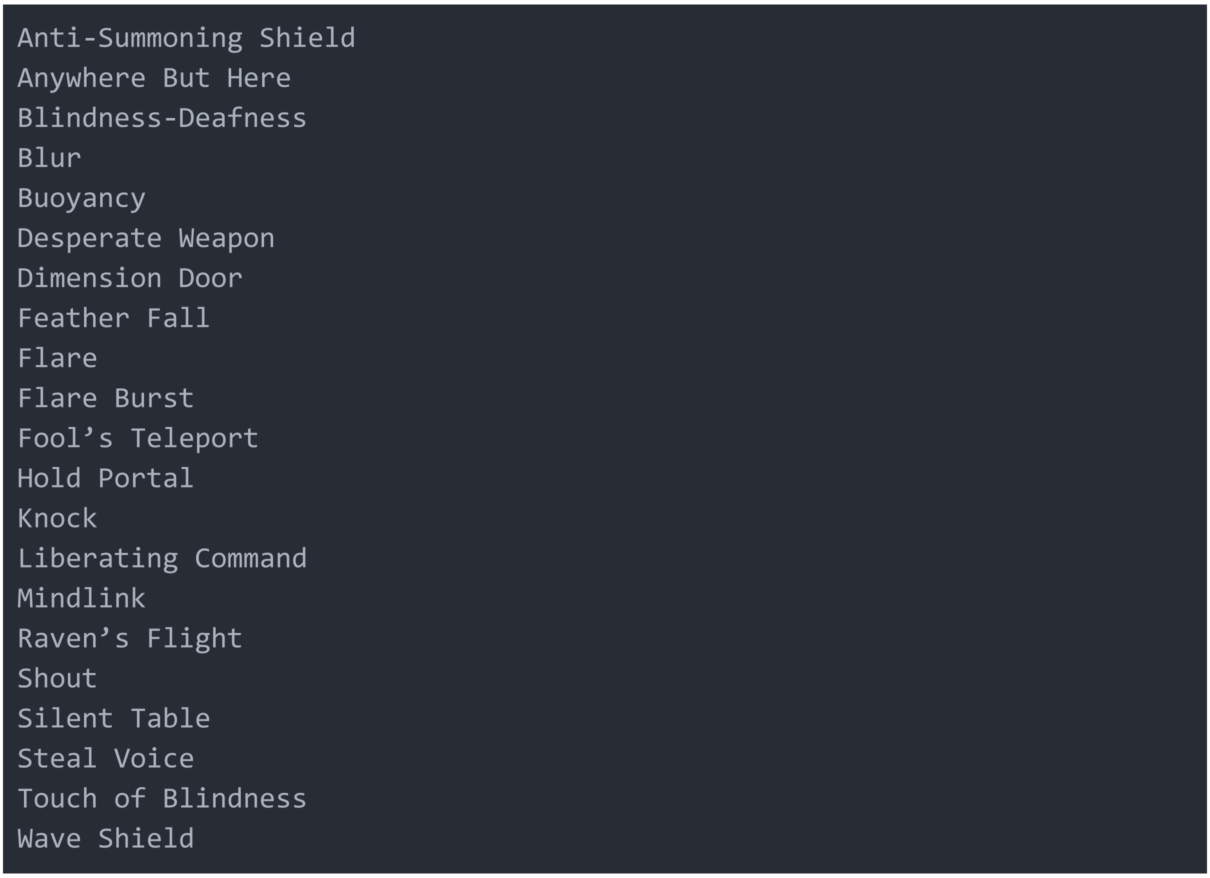
Lien du code du script : https://github.com/G4bleb/distributed-databases/blob/master/Devoir1/jsonToSQL/parserJSON.py

Requête SQLite :



```sql
SELECT
  s.name
FROM
  spell AS s
  INNER JOIN spell_comp AS sc ON s.name = sc.name_spell
  LEFT JOIN component AS c ON sc.name = c.name
  INNER JOIN spell_class_level AS scl ON s.name = scl.name
  LEFT JOIN class_level as cl ON scl.class = cl.class
  AND scl.level = cl.level
WHERE
  cl.class = 'sorcerer/wizard'
  AND cl.level <= 4
GROUP BY
  s.name
HAVING
  COUNT(*) = 1
  AND c.name = 'V';
```

Retour :



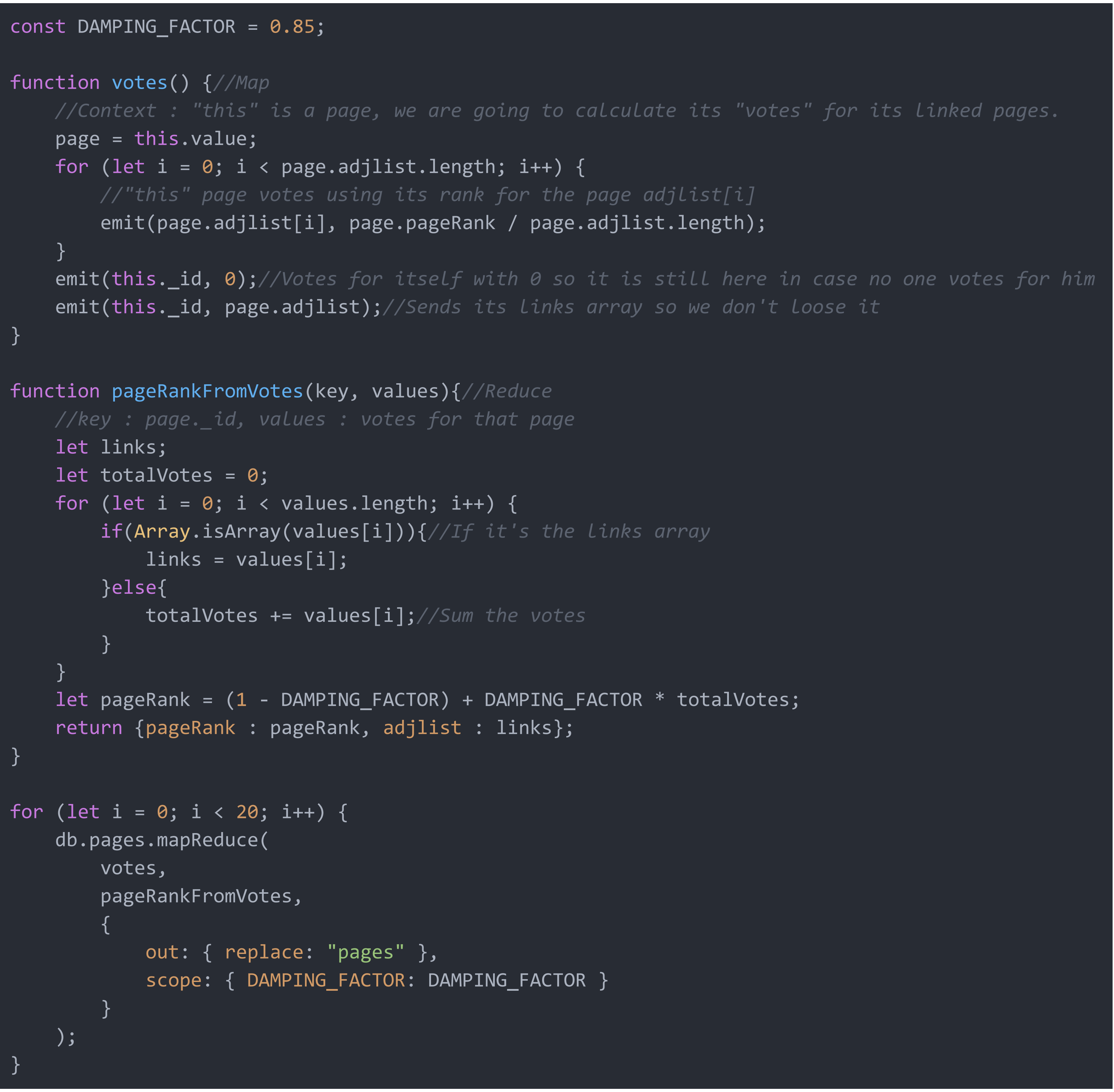
```
Anti-Summoning Shield
Anywhere But Here
Blindness-Deafness
Blur
Buoyancy
Desperate Weapon
Dimension Door
Feather Fall
Flare
Flare Burst
Fool’s Teleport
Hold Portal
Knock
Liberating Command
Mindlink
Raven’s Flight
Shout
Silent Table
Steal Voice
Touch of Blindness
Wave Shield
```

Exercice 2 : Pagerank

On part de cette collection json, qu’on importe dans mongoDB :

```
    {
        "_id": "A",
        "value": {
            "pageRank": 1,
            "adjlist": [
                "B","C"
            ]
        }
    },{
        "_id": "B",
        "value": {
            "pageRank": 1,
            "adjlist": [
                "C"
            ]
        }
    },{
        "_id": "C",
        "value": {
            "pageRank": 1,
            "adjlist": [
                "A"
            ]
        }
    },{
        "_id": "D",
        "value": {
            "pageRank": 1,
            "adjlist": [
                "C"
            ]
        }
    }
```

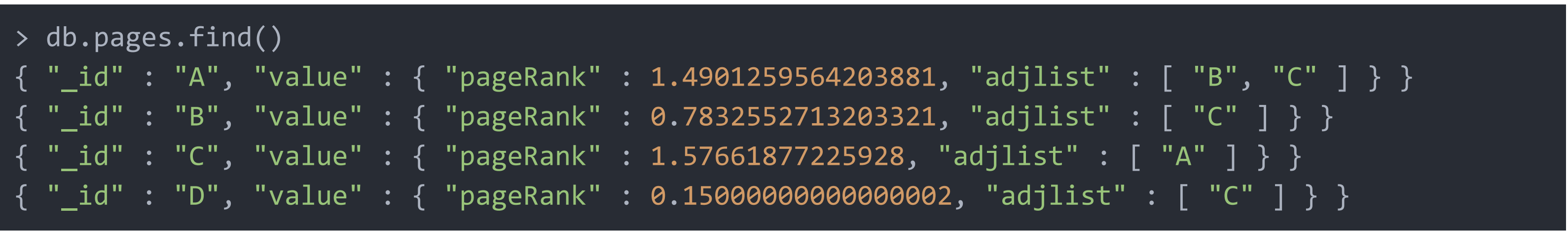
Code du map Reduce pour l'algorithme de pageRank :



```
const DAMPING_FACTOR = 0.85;

function votes() {//Map
    //Context : "this" is a page, we are going to calculate its "votes" for its linked pages.
    page = this.value;
    for (let i = 0; i < page.adjlist.length; i++) {
        //"this" page votes using its rank for the page adjlist[i]
        emit(page.adjlist[i], page.pageRank / page.adjlist.length);
    }
    emit(this._id, 0);//Votes for itself with 0 so it is still here in case no one votes for him
    emit(this._id, page.adjlist);//Sends its links array so we don't loose it
}

function pageRankFromVotes(key, values){//Reduce
    //key : page._id, values : votes for that page
    let links;
    let totalVotes = 0;
    for (let i = 0; i < values.length; i++) {
        if(Array.isArray(values[i])){//If it's the links array
            links = values[i];
        }else{
            totalVotes += values[i];//Sum the votes
        }
    }
    let pageRank = (1 - DAMPING_FACTOR) + DAMPING_FACTOR * totalVotes;
    return {pageRank : pageRank, adjlist : links};
}

for (let i = 0; i < 20; i++) {
    db.pages.mapReduce(
        votes,
        pageRankFromVotes,
        {
            out: { replace: "pages" },
            scope: { DAMPING_FACTOR: DAMPING_FACTOR }
        }
    );
}
```

(aussi trouvable ici : https://github.com/G4bleb/distributed-databases/blob/master/Devoir1/pagerank/map_reduce.js)

Résultat :



```
> db.pages.find()
{ "_id" : "A", "value" : { "pageRank" : 1.4901259564203881, "adjlist" : [ "B", "C" ] } }
{ "_id" : "B", "value" : { "pageRank" : 0.7832552713203321, "adjlist" : [ "C" ] } }
{ "_id" : "C", "value" : { "pageRank" : 1.57661877225928, "adjlist" : [ "A" ] } }
{ "_id" : "D", "value" : { "pageRank" : 0.15000000000000002, "adjlist" : [ "C" ] } }
```