

Pokédex

Antonin Dudermel & Nicolas Pons

1 fichiers nécessaires

Avant de lancer `dudermel-pons-projet.py`, s'assurer que les fichiers suivants sont présents dans le même répertoire que la base de donnée :

- `principal.py`
- `requetes.py`
- `main.py`
- `calculs.py`
- `interface.py`
- `textes.py`
- `pokedex.db`
- `effets.csv`
- `spe_phys_capa.csv`

Une fois ces vérifications faites, exécuter :

```
1 python3 dudermel-pons-projet.py
```

2 Aide

2.1 naviguer dans les menus

Après quelques politesses avec le Professeur Chen, vous voici entré en contact avec l'interface du Pokédex. À chaque menu, plusieurs options se présentent. Pour en sélectionner une, tapez le numéro en face de l'option puis appuyez sur la touche entrée

2.2 autocomplétion

Le Pokédex propose un système d'autocomplétion : si, à la demande de l'interface, vous entrez un mot mais qu'il n'est pas dans la liste des possibles, l'interface l'augmente tout en proposant la liste des champs à compléter. Si aucun champ ne correspond, l'interface soulève (et rattrape) l'erreur `ValueError`.

2.3 arborescence

1. chercher un pokémon
 - (a) par nom français
 - (b) par nom anglais
 - (c) par type
2. chercher une capacité
 - (a) par nom
 - (b) par type

- 3. Logiciel stratégique
 - (a) Calculateur de dégâts
 - (b) Étude d'équipe

3 Les requêtes

L'ensemble des requêtes est groupé dans le fichier `requetes.py`. La fonction `requete_simple` génère l'ensemble des requêtes simples. Le module `sqlite3` ne disposant pas de sécurité contre les injections de code pour tout ce qui est nom de table ou de champ, la fonction `securite` remplit ce rôle : si elle reçoit en entrée une chaîne de caractères contenant ; ou de taille supérieure à une limite `LIMSECU`, fixée par défaut à 12 (valeur supérieure à tous les noms de champs et de tables, mais inférieure à la taille de `DROP TABLE type`), elle soulève l'erreur `DangerError`. Cependant, compléter une requête n'est jamais proposé à l'utilisateur.

Pour des raisons pratiques et pour l'homogénéité du programme, nous avons préféré utiliser presque systématiquement les clés primaires de chaque table. On dispose ainsi de peu de requêtes complexes. Cependant, en voici quelques unes.

Une requête complexe est appelée par la fonction `evolution` : il s'agit de récupérer à partir de l'identifiant d'un pokémon la liste des noms des pokémons en lesquels il peut évoluer, ainsi que les conditions d'évolution. On effectue donc une requête croisée sur la table `evolution` et la table `pokemon`.

Les deux fonctions de recherche par type utilisent des requêtes complexes : elles créent une vue contenant la table `pokemon` ou `capacite` ayant reçu le filtre suivant : ne sont retenues que les capacités ou pokémons correspondant au type donné en argument. On lance ensuite une recherche simple sur cette vue avant de la supprimer.

4 Oups...

Si vous avez modifié ou détruit la base de données et que, par malheur, le programme plante entièrement, *Pas de Panique!* Il vous suffit, dans un terminal, de vous placer dans un dossier contenant les fichiers `atqspe.csv`, `tablecap.csv`, `pokemon.csv`, `evolution.csv` et `types.csv`, et de taper :

```
rm pokedex.db
sqlite3 dudermel-pons-projet.db < creer_tables.sql
```