

# Devoir Maison - 1 - Bases de données - Yu-Gi-Oh

## I - Introduction<sup>1</sup>

Le jeu de cartes Yu-Gi-Oh issu du manga du même nom est un jeu de cartes consistant à faire s'affronter des monstres et de réduire à néant les points de vie de son adversaire. Les cartes « Monstres » possèdent des niveaux d'attaque et de défense qui vont permettre lors d'un affrontement entre deux monstres de réduire les points de vie des joueurs. Il existe deux autres genres de cartes dans ce jeu : les cartes « Piège » et les cartes « Magie ».



(a) Carte monstre



(b) Carte piège



(c) Carte magie

FIGURE 1 – Les types de cartes du jeu Yu-Gi-Oh

Chaque carte « Monstre », « Piège » ou « Magie » fait partie d'un type. Par exemple, le monstre « Dragon blanc aux yeux bleus » est de type « Normal ». Il s'agit d'un monstre de la catégorie « Dragon » ayant pour attribut « Lumière ». Les cartes « Piège » ou « Magie » ont des effets différents en fonction du type et de la catégorie auxquelles elles appartiennent.

Un collectionneur a créé une base de données de l'ensemble des cartes dont il dispose. Dans cet exercice, nous n'utiliserons qu'une partie de cette base de données. Le diagramme entité-association de la partie de la base de données que l'on va utiliser est le suivant. Les mots en gras correspondent aux clés primaires des tables. On précise que dans ce genre de collections, il existe différentes cartes issues de différentes éditions et ont donc des références différentes.

**Attention**, il existe dans la table principale **cards** deux enregistrements **Card\_ID** et **ID\_Card**. **Card\_ID** est unique pour chaque carte de la collection. **ID\_Card** n'est pas nécessairement unique car il existe les mêmes cartes dans plusieurs éditions. Ainsi, **Card\_ID** est l'identifiant lié à chaque carte tandis que **ID\_Card** fait référence à l'identifiant d'un monstre, d'un piège ou d'une carte magie de l'édition de base du jeu. Les cartes ayant les mêmes caractéristiques d'une édition à l'autre, on peut retrouver celles-ci pour chacune des cartes via **ID\_Card**. D'après la Figure 4 ces deux cartes ont deux valeurs **Card\_ID** différents mais une même valeur pour **ID\_Card**.

1. Code Capytale : 21e6-703286



(a) Caractéristiques d'une carte monstre



(b) Une carte monstre de type « Effet »

FIGURE 2 – Les caractéristiques des cartes

## II - Travail sur la table cards

**Objectif :** Dans cette partie, on cherche à utiliser les données de la table **cards** afin d'obtenir diverses informations. Dans cette partie, on ne travaille que sur la table **cards**. Les commandes utiles pour traiter les questions sont : **SELECT**, **FROM**, **WHERE**, **COUNT**, **MAX**, **EXCEPT** et éventuellement **BETWEEN**.

- 1 De combien de cartes au total se compose la collection de cartes ?
- 2 Quels sont les niveaux d'attaque et de défense de la carte monstre **Guardian Grarl** ?
- 3 Quelles sont les cartes « Monstre » ayant un niveau d'attaque entre 500 et 550 et un niveau de défense entre 150 et 200 ?

Puisque certains enregistrements d'attaque et de défense ne sont pas renseignés, il y a un problème pour déterminer le maximum de la valeur de ces enregistrements. En effet, on aura le retour d'une composante vide en demandant **SELECT MAX(Atk) FROM cards**. Pour éviter ce problème, il aurait fallu que la personne ayant créé la base de données renseigne ces enregistrements par **NULL**.

- 4 Donner la commande SQL permettant de récupérer tous les enregistrements dont les attaques (ou défense) ne sont pas vides. (un champ vide est ici représenté par un champ rempli avec une chaîne de caractères vide).
- 5 Donner la commande SQL permettant de récupérer tous les enregistrements dont les attaques (ou défense) ne sont pas des chaînes de caractères vides à l'aide de la commande **EXCEPT**.
- 6 La commande précédente permet de créer une nouvelle table. Donner la taille de celle-ci et vérifier qu'elle est bien égale à  $5427 - 2235 = 3192$ , où 5427 est la réponse attendue à la question 1 et 2235 le nombre total de cartes magie et piège.
- 7 Déterminer via une commande SQL, le niveau d'attaque maximum que peut avoir un monstre.
- 8 Déterminez, à l'aide d'un processus de double sélection, les deux monstres qui ont ce niveau d'attaque maximum.
- 9 Déterminez les monstres qui ont le niveau d'attaque le plus grand tout en étant un monstre de niveau (**Level**) inférieur ou égal à 8. Vous indiquerez également leur niveau d'attaque.

## III - Jointure entre tables

**Objectif :** Dans cette partie de l'exercice, on va réaliser des jointures entre plusieurs tables afin d'obtenir différents types de renseignements sur les cartes.

- 10 Quel est le type de la carte ayant pour nom **Vampire Genesis** ? Justifier qu'il faut utiliser les tables **cards** et **types** pour cela.

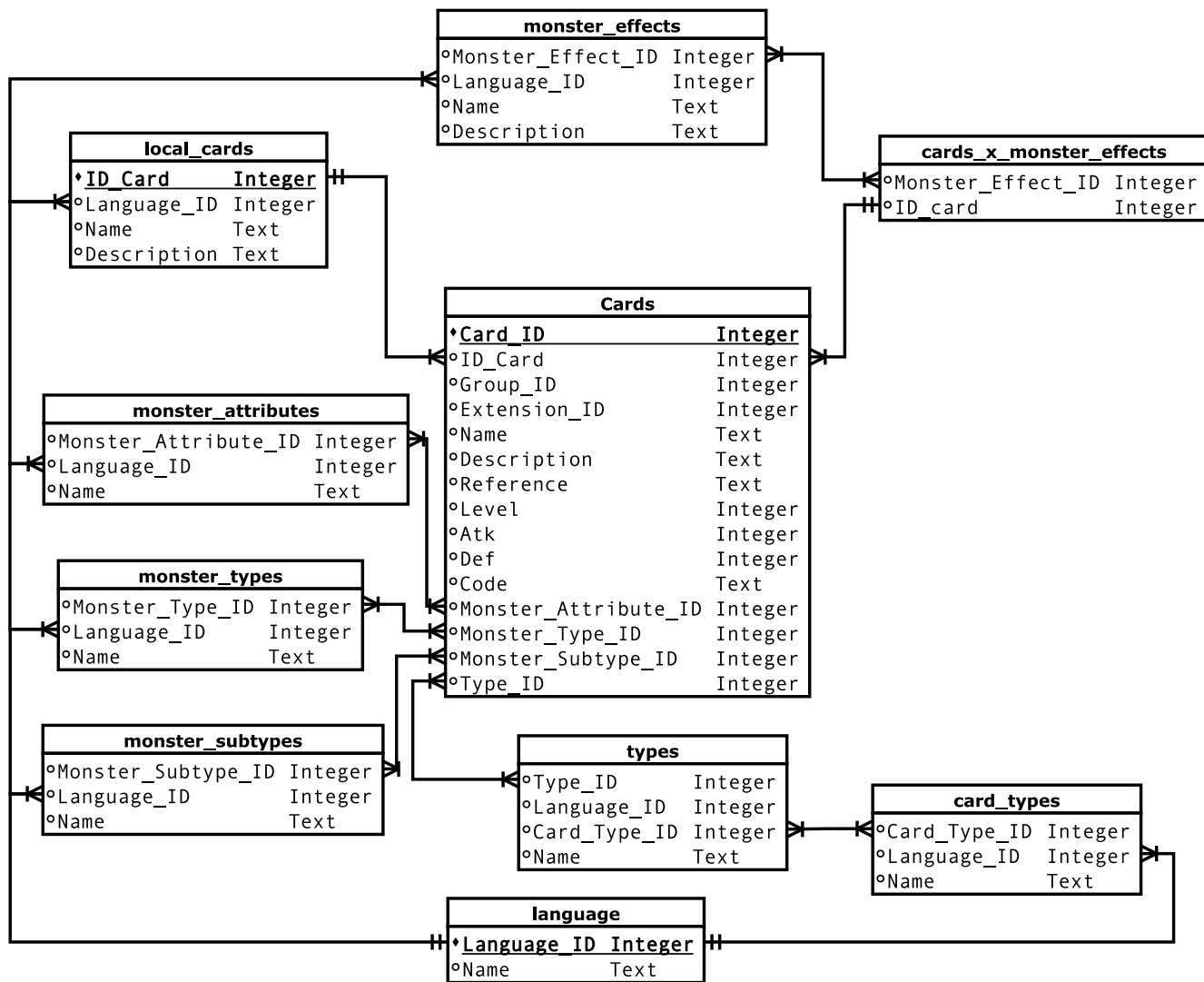


FIGURE 3 – Diagramme entité-association



(a) Monstre d'une édition spécifique



(b) Monstre d'une autre édition spécifique

FIGURE 4 – Différentes éditions

- 11 Quel est l'effet de la carte ayant pour nom **Vampire Genesis** ?

## IV - Opérations d'agrégations

**Objectif :** On souhaite savoir quels sont en moyenne les types de monstres les plus performants (Aqua, Bête, Bête-Guerrier, Dinosaur, etc.). Pour obtenir cette information, on va utiliser la commande **AVG** qui est une commande d'agrégation des données, mais on pourra aussi utiliser **GROUP BY**, **ORDER BY** ou **HAVING**.

- 12 En effectuant la jointure entre les tables **cards** et **monster\_types**, réaliser le calcul du niveau d'attaque moyen par type de monstre de la collection de cartes de la base de données fournies. On vous demande d'afficher la projection du niveau d'attaque moyen et du type de monstre en **français** (**Language\_ID=2**).

- 13 Classer maintenant dans l'ordre croissant ces résultats.

- 14 Écrire la requête qui ne sélectionne que les groupes qui comportent au moins 300 monstres.

## V - Autojointure sur la table cards

**Objectif :** Le collectionneur veut séparer en deux sa collection : d'un côté les cartes dont il dispose de façon unique et de l'autre celles dont il dispose dans plusieurs références.

- 15 Récupérer les noms distincts des cartes ayant plusieurs références. *Indication : on pourra effectuer une jointure de **cards** sur elle même en conservant les paires des cartes vérifiant certaines contraintes sur **Id\_Card** et **Card\_Id**.*
- 16 Récupérer les noms distincts des cartes apparaissant dans au moins 10 éditions. (*Attention : C'est une question pas aussi si facile qu'elle en a l'air*).

## VI - Retour sur la table cards

**Objectif :** On va utiliser dans cette partie les commandes **LIMIT** et **OFFSET** et voir leur utilité.

- 17 Donner la commande permettant d'obtenir dans l'ordre décroissant les niveaux d'attaque des monstres de la table **cards** (on peut notamment réutiliser une requête du début du DM).
- 18 À partir de la question précédente, donner la commande permettant d'obtenir les niveaux d'attaques **distincts** des monstres de la table **cards**.
- 19 Donner les 10 premiers niveaux d'attaques distincts de la collection de cartes de la base de données étudiée.
- 20 Donner les 5 niveaux d'attaques distincts entre la 11<sup>ème</sup> et la 15<sup>ème</sup> position dans l'ordre décroissant.