

# Projet DKM - DBPanda



Présenté par CHAFFIN Antoine, YASSINE Taha et DELAUNAY Julien



## Choix des données

- Structure intéressante
- Centre d'intérêt commun
- Données facilement accessibles



**eSport**



**PandaScore**



# PandaScore

- Base de données sur l'eSport
- Fourni une API pour récupérer les différents objets
  - Données descriptives
  - Données en temps réel



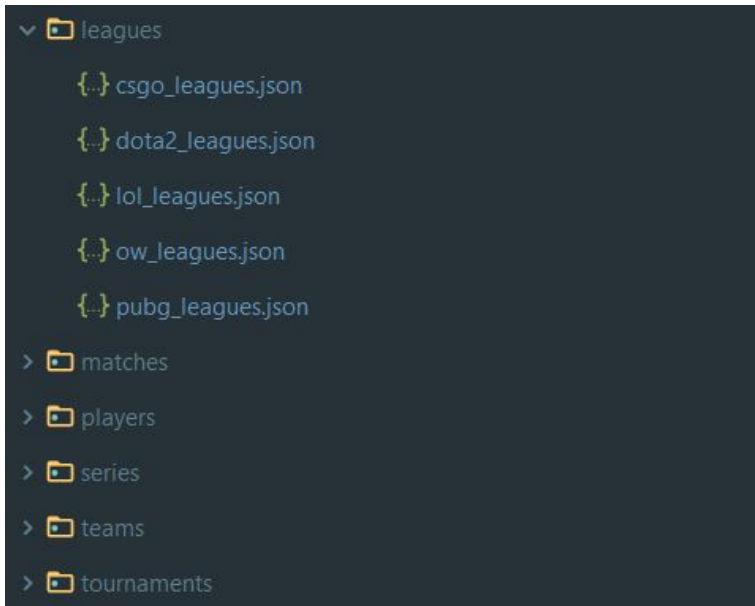
# Récupération des données

- Différents endpoints pour chaque type d'objet
  - Leagues, séries, tournois, matchs, équipes et joueurs
- Spécialisation de ces endpoint pour chaque jeu
  - CS:GO, LoL, PUBG, Dota 2, OW



# Récupération des données

- Script qui récupère chaque type d'objet pour un jeu donné
- Stock le JSON judicieusement
- Récupération de tous les objets pour tous les jeux

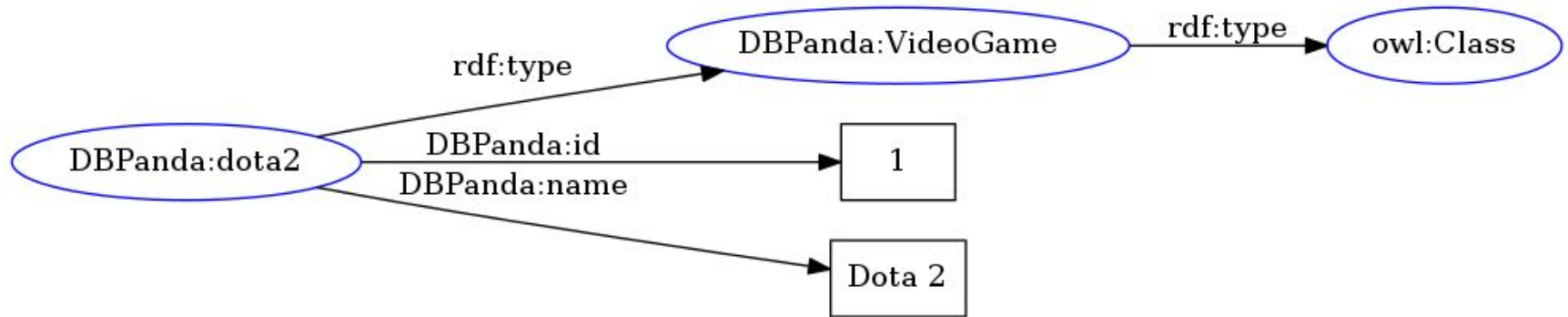




## Création de l'endpoint

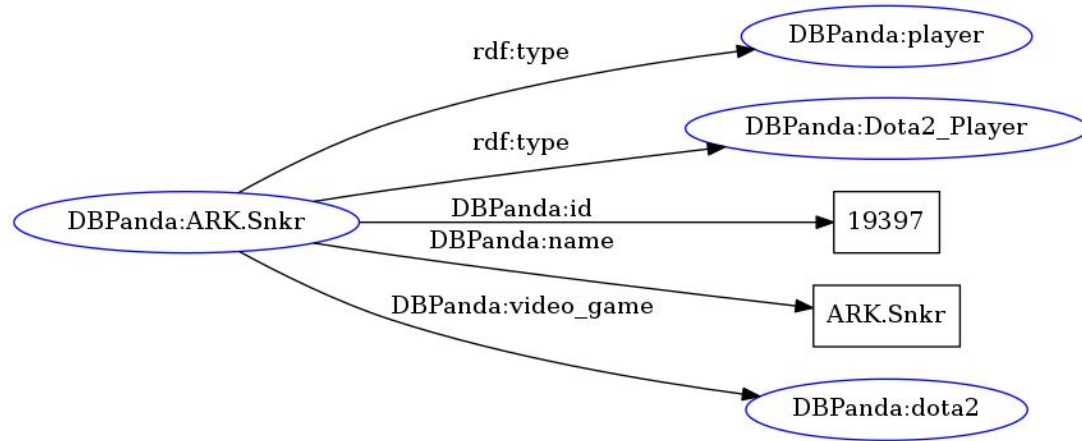
- Permet de tester tout au long de la création du modèle
- Utilisation de Fuseki
- Fuseki met à disposition la base de connaissance générée en créant un endpoint SPARQL

# Création du modèle



Namespaces:  
DBPanda: <http://DBPanda/>  
owl: <http://www.w3.org/2002/07/owl#>  
rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>  
xsd: <http://www.w3.org/2001/XMLSchema#>  
rdfs: <http://www.w3.org/2000/01/rdf-schema#>

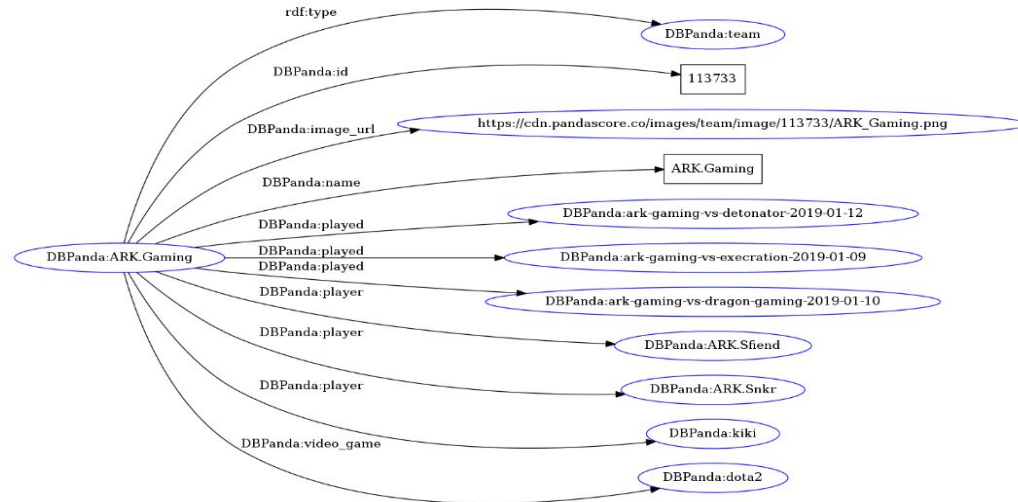
# Création du modèle



Namespaces:  
DBPanda: <http://DBPanda/>  
owl: <http://www.w3.org/2002/07/owl#>  
rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>  
xsd: <http://www.w3.org/2001/XMLSchema#>  
rdfs: <http://www.w3.org/2000/01/rdf-schema#>

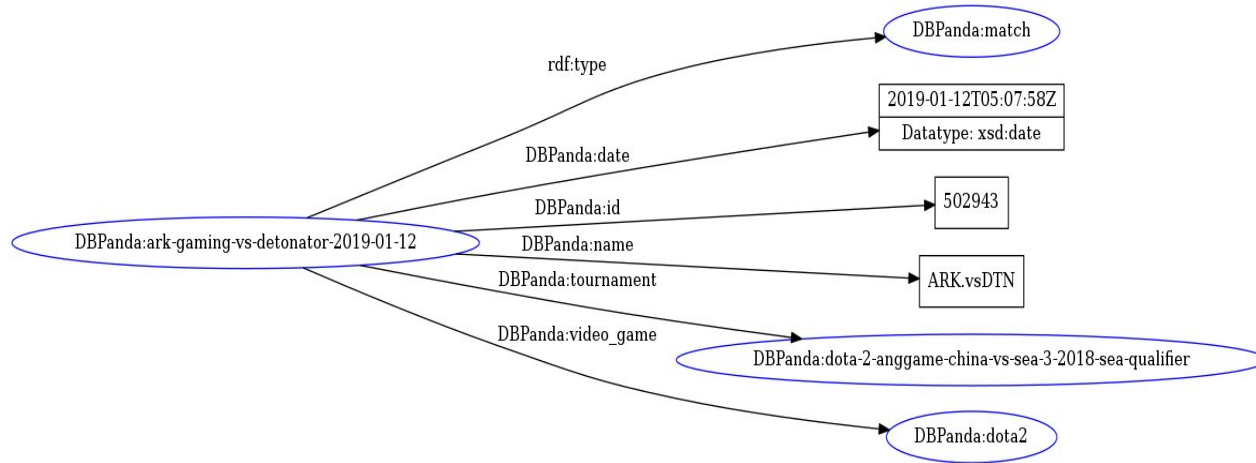


# Création du modèle



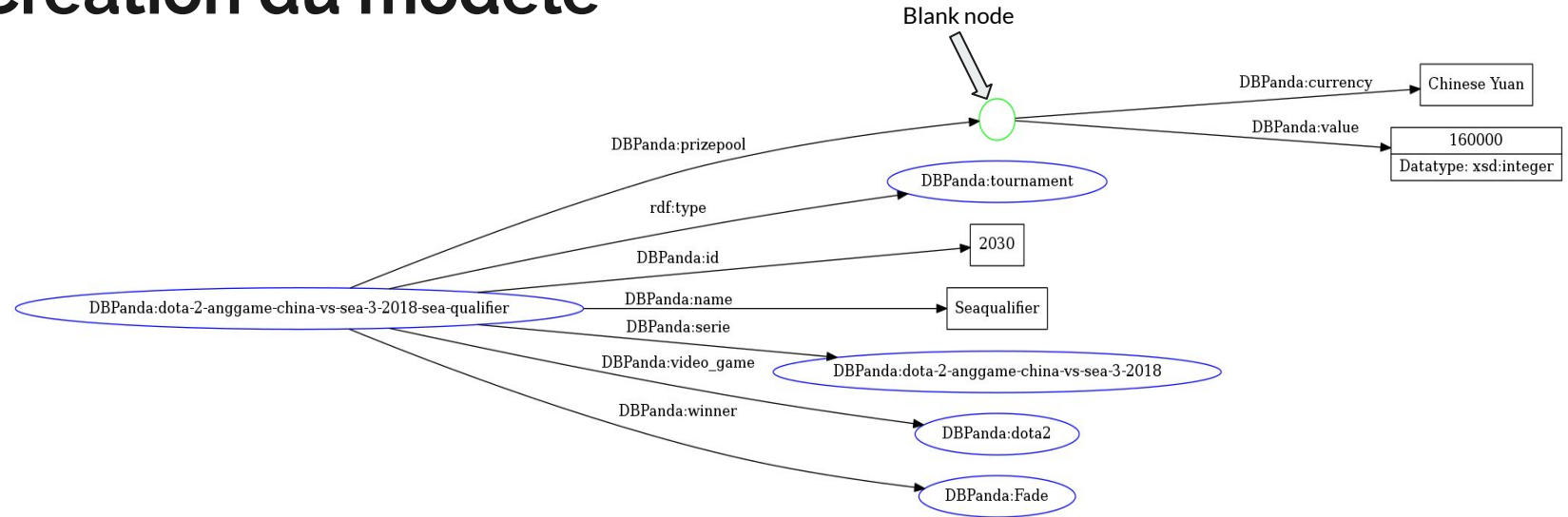
Namespaces:  
DBPanda: <http://DBPanda/>  
owl: <http://www.w3.org/2002/07/owl#>  
rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>  
xsd: <http://www.w3.org/2001/XMLSchema#>  
rdfs: <http://www.w3.org/2000/01/rdf-schema#>

# Création du modèle



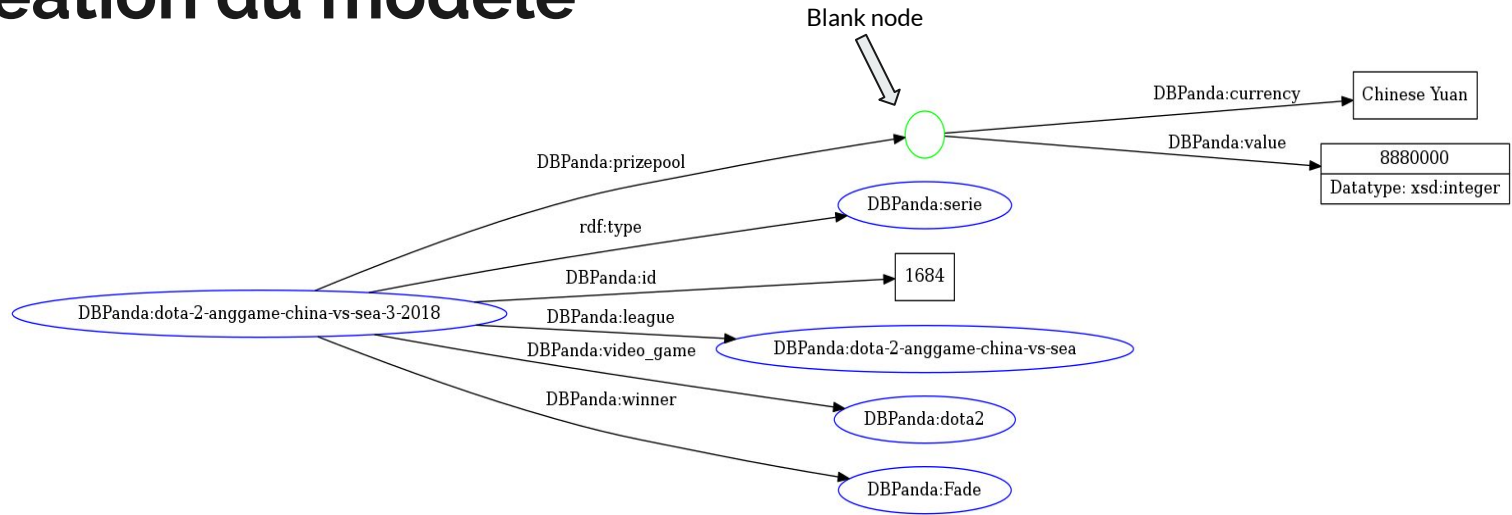
Namespaces:  
DBPanda: <http://DBPanda/>  
owl: <http://www.w3.org/2002/07/owl#>  
rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>  
xsd: <http://www.w3.org/2001/XMLSchema#>  
rdfs: <http://www.w3.org/2000/01/rdf-schema#>

# Création du modèle



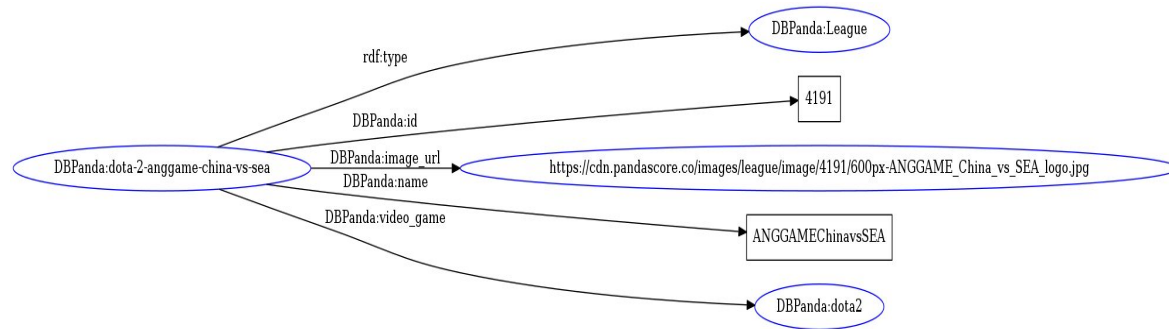
Namespaces:  
DBPanda: <http://DBPanda/>  
owl: <http://www.w3.org/2002/07/owl#>  
rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>  
xsd: <http://www.w3.org/2001/XMLSchema#>  
rdfs: <http://www.w3.org/2000/01/rdf-schema#>

# Création du modèle



Namespaces:  
DBPanda: <http://DBPanda/>  
owl: <http://www.w3.org/2002/07/owl#>  
rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>  
xsd: <http://www.w3.org/2001/XMLSchema#>  
rdfs: <http://www.w3.org/2000/01/rdf-schema#>

# Création du modèle



Namespaces:  
DBPanda: <http://DBPanda/>  
owl: <http://www.w3.org/2002/07/owl#>  
rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>  
xsd: <http://www.w3.org/2001/XMLSchema#>  
rdfs: <http://www.w3.org/2000/01/rdf-schema#>



# Modélisation via Jena

## Jena

- OntModel
- NsPrefix
- OntClass
- Ressource
- ObjectProperty



## RDF / RDFS / Owl

- Modèle complet (auquel on ajoute les éléments)
- Préfixe du namespace (<http://DBPanda/> = DBPanda:)
- OwlClass
- Sujet/Objet avec des URI
- Prédicat



# Modélisation via Jena

- Création de ressource dans le modèle (sujet)
- Ajout de property à nos ressources avec comme objet :
  - Des littéraux
  - Des littéraux typés
  - D'autres ressources
- > Création de triplets
- Property "<http://www.w3.org/1999/02/22-rdf-syntax-ns#type>" avec une OntClass comme objet pour l'assignation d'une classe
- InverseProperty pour les liaisons entre sujets
- Traitement de chaque objet/jeu indépendamment
  - Mutualisation des méthodes
- Extraction au format souhaité



# Exploration via SPARKLIS

- Tests en temps réel pendant la création de la base
  - Vérification de la modélisation et des relations
- Choix de certaines méthodes de modélisation
  - Pas d'utilisation de bag
- Parcours intuitif de la base et aide à la création de requêtes utiles



---

# Requêtes SPARQL



## Somme totale des gains de chaque joueur

```
SELECT DISTINCT ?player (SUM(?value) AS ?total)
WHERE { ?player a DBPanda:player .
        /* On sélectionne tous les joueurs */
        ?thing DBPanda:winner ?player .
        /* On récupère ce que le joueur a gagné (séries, tournois...) */
        ?thing DBPanda:prizepool ?prizepool .
        /* Ce que le joueur à gagné doit posséder un prizepool */
        ?prizepool DBPanda:currency "United States Dollar" .
        /* On fixe la monnaie du prizepool afin
        de ne pas sommer des valeurs dans différentes monnaies */
        ?prizepool DBPanda:value ?value }
        /* On récupère les valeurs qu'on sommera */
GROUP BY ?player
        /* On affiche les résultats pour chaque joueur */
```



## Équipes ayant joué un match dans une league précise

```
SELECT DISTINCT ?team
WHERE { ?team a DBPanda:team .
        /* On sélectionne les équipes */
        ?match a DBPanda:match .
        /* On sélectionne les matchs */
        ?match DBPanda:tournament ?tournament .
        /* Les matchs doivent avoir faire parti d'un tournoi */
        ?tournament DBPanda:serie ?serie .
        /* Le tournoi doit faire partie d'une série */
        ?serie DBPanda:league <http://DBPanda/esl-one> .
        /* La série doit faire partie de la league ESL-ONE */
        ?team DBPanda:played ?match . }
        /* On récupère toutes les équipes qui ont joué un de ces matchs */
```



## Équipes multi gaming

```
SELECT DISTINCT ?team
WHERE { ?team a DBPanda:team .
        /* On sélectionne toutes les équipes */
        ?team DBPanda:video_game ?video_game_1 .
        /* On récupère les jeux des équipes (variable 1) */
        ?team DBPanda:video_game ?video_game_2 .
        /* On récupère les jeux des équipes (variable 2)*/
        FILTER(?video_game_1 != ?video_game_2) }
        /* On filtre de sorte à ce que le jeu 1 soit différent du jeu 2*/
```



## Classement des équipes par nombre de victoires

```
SELECT DISTINCT ?team (COUNT(DISTINCT ?match) AS ?number_of_win)
/* On fait la somme des matchs gagnés */
WHERE { ?team a DBPanda:team .
        /* On récupère les équipes */
        ?match a DBPanda:match .
        /* On récupère les matchs */
        ?match DBPanda:winner ?team . }
/* On fait le lien entre les matchs et les vainqueurs */
GROUP BY ?team
/* On affiche les résultats pour chaque équipe */
ORDER BY DESC(?number_of_win)
/* On trie par ordre décroissant pour avoir le classement */
```



## Joueurs ayant gagné un match depuis une certaine date

```
SELECT DISTINCT ?player
WHERE { ?player a DBPanda:player .
        /* On selectionne les joueurs */
        ?match a DBPanda:match .
        /* On sélectionne les matchs */
        ?match DBPanda:winner ?player .
        /* On fait le lien entre les matchs et les vainqueurs */
        ?match DBPanda:date ?date .
        /* On récupère la date des matchs */
        FILTER ( (?date > "2018-01-01"^^xsd:date) ) }
        /* On conserve seulement les matchs qui ont eu lieu après la date
```



## “Fusion” des requêtes

- Somme total des gains de chaque joueur dans une league précis
- Classement des équipes sur une league ou un jeu
- Equipe multigaming qui a gagné un match depuis une certaine date
- ....