AUFFRAY Baptiste
BOUTIN Loïc
PINEL Félix

Bases de données évoluées – Projet OLAP

Nous souhaitons réaliser un entrepôt de données afin de pouvoir utiliser des requêtes spécifiques à ce format de bases de données et le modéliser en plusieurs dimensions. Nous avons commencé par trouver un dataset correct, que nous avons integré avec Talend puis inséré dans MySQL afin d'effectuer des requêtes.

Présentation de notre sujet

Nous avons choisi de travailler sur des données concernant des publications scientifiques disponibles sur le site des archives ouvertes HAL. Nous avions d'abord pensé à prendre pour dataset le résultat d'une recherche par mot-clé, en confondant plusieurs types de document mélangés (livres, articles, photos...) selon un terme de recherche choisi arbitrairement. Mais les données sur les documents étaient finalement trop hétérogènes à cause des natures trop différentes des entrées, avec des attributs parfois trop spécifiques à certains types de documents, laissant par exemples de gros creux vides dans notre base de données.

Nous avons donc décidé de ne pas garder les attributs spécifiques aux documents sauf le NTT (voir attributs) comme exemple d'attribut spécifique. Nous avons aussi abandonné l'idée d'un terme de recherche pour avoir un résultat plus généraliste en terme de données. À partir de là, nous avons pu exporter notre recherche au format CSV sur le site de HAL en choisissant des attributs pertinents que nous allons détailler. Le nombre de résultats étant trop grand, seul un échantillon des tuples a été récupéré.

Présentation du dataset

Pour chaque entrée de document, nous avions dans le fichier .csv source les attributs suivants :

- halId_s : Identifiant unique géré dans HAL
- version_i : Version du document. La grande majorité des documents sont dans leur première version.
- uri_s : Lien URI du document dans l'archive HAL
- docType_s : Type de document
- doiId_s : Digital Object Identifier
- nntId_s : Numéro National de Thèse. Attribut spécifique aux thèses
- title_s : Titre du document
- abstract_s - summary: Résumé du document
- keyword_s : Mot-clés des thèmes du document. Multivalues
- authId_i : Identifiant des auteurs. Multivalues
- authFirstName_s : Prénoms des auteurs. Multivalues
- authLastName_s : Noms de famille des auteurs. Multivalues
- authFullName_s : Noms complets des auteurs. Multivalues
- producedDate_s : Date de publication
- domain_s : Domaines de connaissances concernés par le document. Multivalues
- language : Language du document

Certains de ces attributs nous ont posé des problèmes, notamment les multivalués. Par exemple, nous aurions souhaité effectuer des requêtes sur les noms des auteurs, mais le format n'étant pas uniforme entre les lignes, nous avons plutôt utilisé l'ID auteur pour effectuer nos

requêtes concernant les auteurs. Nous souhaitions également faire des requêtes sur les keywords mais encore une fois le format inconsistant rendait la tâche trop compliquée sans prétraitement exhaustif.

Un autre attribut plus embêtant a été la date de publication : En effet, la date n'est pas toujours complète (absence du jour mais surtout du mois chez certains tuples) voire manquante. Afin de rendre des requêtes tout de même possibles nous avons choisis de compter la présence des années seules comme la publication le premier janvier. Cela fausse donc certaines requêtes (par mois, surtout les 1er janvier) mais permet tout de même de facilement faire des recherches par année.

Intégration des données

Tables de dimension :

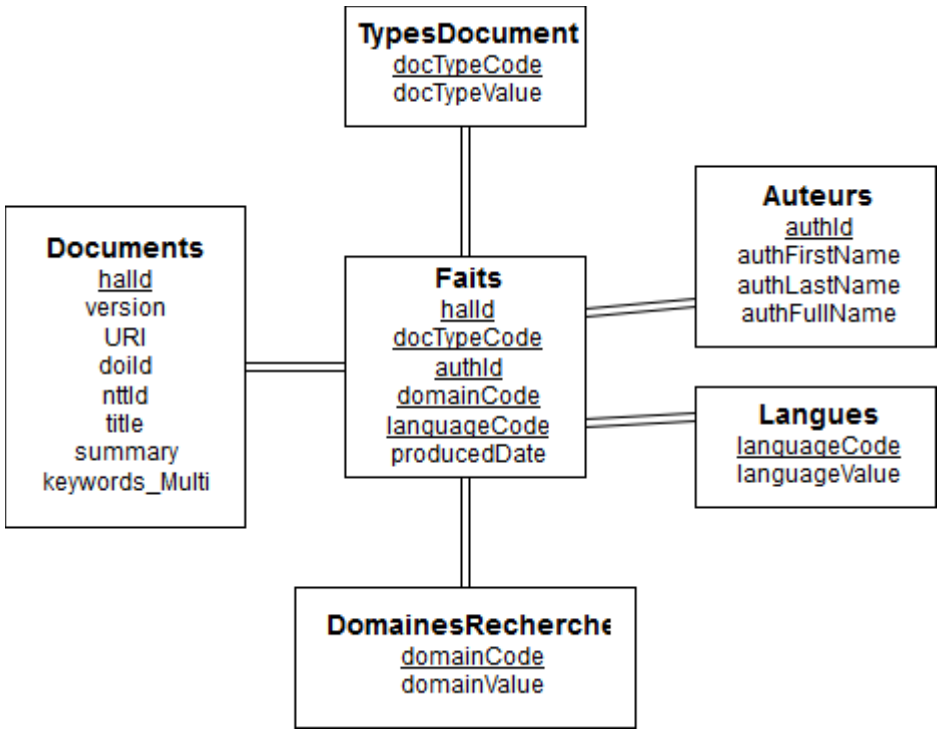
Nous avons créé des tables de dimension pour les attributs auteurs, type de document, langue, et domaines de recherche. Les types de documents sont présentés sous la forme de codes, que nous avons fait correspondre à leur valeur en se basant sur l'API de HAL, disponible en JSON. Les langues sont présentés sous la forme de codes ISO 639, que nous avons fait correspondre à leur valeur en anglais grâce à un fichier CSV (source : https://datahub.io/core/language-codes#resource-language-codes-full). Les domaines de recherche sont présentés sous forme de codes dont la signification se trouve dans le référentiel de HAL (AuréHAL), et que nous avons mappé dans une table à l'aide de l'API de HAL sous forme de fichier XML.

Les données sur les auteurs se trouvant déjà dans notre dataset, nous avons normalisé les attributs multivalués authID (codes identifiants des auteurs), authFirstName, authLastName, authFullName grâce à la fonction tNormalize de Talend, puis mappé ces différents attributs entre eux grâce à la fonction tMap. Nous obtenons une table avec les noms des auteurs et leur authID comme clé primaire.

Table(s) de faits :

Pour mieux gérer les attributs multivalués qui multiplieraient le nombre de tuples redondant dans la table de fait, nous avons créé une table « Document », agissant comme une table de dimension, contenant les informations uniques à chaque document qui ne seront pas liées aux tables de dimensions. Sa clé primaire est le HalId, qui la relie aussi à la table de faits de même clé primaire, qui contient toutes les combinaisons entre les informations sur les documents et les clés étrangères qui les relient aux tables de dimensions.

Nous obtenons le schéma de tables suivant :



Requêtes

Nous avons inséré nos tables dans MySQL puis constitué les requêtes suivantes :

Requête 1 :
Permet de récupérer le nombre de langues différentes par année et domaine d'étude.

```
SELECT YEAR(producedDate) as yearDate, domainCode, COUNT(*) as nbLanguage
FROM(
        SELECT producedDate, domainCode, languageCode
        FROM faits group by domainCode, languageCode
) AS temp
GROUP BY domainCode;
```

Résultat : Voici les dix premiers tuples de la sortie :

yearDate	domainCode	nbLanguage
2017	chim	2
2017	chim.anal	1
2017	chim.cata	1
2017	chim.coor	1
2017	chim.geni	1
2017	chim.mate	2
2017	chim.orga	2
2017	chim.poly	1
2017	chim.theo	1

Requête 2 :
Rend le top 10 des auteurs les plus présents en tant qu'auteur dans les différents documents.

```
SELECT authHal.authId, authFullName, COUNT(halId) AS nb
FROM (
```

```
        SELECT authId, halId
        FROM faits
        GROUP BY authId, halId
) AS authHal, auteurs
WHERE authHal.authId = auteurs.authId
GROUP BY authId
ORDER BY nb DESC
LIMIT 10;
```

Résultat :

authId	authFullName	nb
1668793	Annelies Braffort	50
91255	R. Duran	41
454153	Ange Nzihou	37
1667920	Sylvain Perrot	29
350896	Jean-Paul Kleider	25
1413605	Jean-Pierre Chaumeil	25
149150	Emmanuel Grimaud	24
1668762	Vincent Porhel	24
91251	M. Goñi-Urriza	23
894612	B. Lauga	22

Requête 3 :

Nous déclarons d'abord cette vue :

```
CREATE VIEW MonoNoyau AS
   SELECT DISTINCT(halId), docTypeCode, languageCode, producedDate
   FROM faits;
```

Cela crée une vue à partir de la table des faits contenant les attributs non multivalués de celle ci.

Le CUBE n'étant pas supporté par notre version de MySQL, cette requête simule cette instruction pour obtenir un résultat similaire, sur la vue « MonoNoyau » de notre table de faits.

```
SELECT halId, docTypeCode, languageCode, producedDate, COUNT() AS nb
FROM monoNoyau
GROUP BY halId, docTypeCode, languageCode, producedDate WITH ROLLUP
UNION
SELECT halId, docTypeCode, languageCode, producedDate, COUNT() AS nb
FROM monoNoyau
GROUP BY docTypeCode, languageCode, producedDate, halId WITH ROLLUP
UNION
SELECT halId, docTypeCode, languageCode, producedDate, COUNT() AS nb
FROM monoNoyau
GROUP BY languageCode, producedDate, halId, docTypeCode WITH ROLLUP
GROUP BY docTypeCode, languageCode, producedDate, halId WITH ROLLUP
UNION
SELECT halId, docTypeCode, languageCode, producedDate, COUNT() AS nb
FROM monoNoyau
GROUP BY producedDate, halId, docTypeCode, languageCode WITH ROLLUP
```

Résultat : Voici les dix premiers tuples de la sortie :

halId	docTypeCode	languageCode	producedDate	nb
anses-01633976	ART	en	2016-01-01 00:00:00	1
anses-01633976	ART	en	NULL	1
anses-01633976	ART	NULL	NULL	1
anses-01633976	NULL	NULL	NULL	1
cea-01632567	ART	en	2017-10-25 00:00:00	1
cea-01632567	ART	en	NULL	1
cea-01632567	ART	NULL	NULL	1
cea-01632567	NULL	NULL	NULL	1
cel-01632228	LECTURE	en	2017-09-01 00:00:00	1

On crée la vue correspondant à ce cube pour de futures requêtes :

```
CREATE VIEW MonoFaitsCube AS
SELECT halId, docTypeCode, languageCode, producedDate, COUNT() AS nb
FROM monoNoyau
GROUP BY halId, docTypeCode, languageCode, producedDate WITH ROLLUP
UNION
SELECT halId, docTypeCode, languageCode, producedDate, COUNT() AS nb
FROM monoNoyau
GROUP BY docTypeCode, languageCode, producedDate, halId WITH ROLLUP
UNION
SELECT halId, docTypeCode, languageCode, producedDate, COUNT() AS nb
FROM monoNoyau
GROUP BY languageCode, producedDate, halId, docTypeCode WITH ROLLUP
GROUP BY docTypeCode, languageCode, producedDate, halId WITH ROLLUP
UNION
SELECT halId, docTypeCode, languageCode, producedDate, COUNT() AS nb
FROM monoNoyau
GROUP BY producedDate, halId, docTypeCode, languageCode WITH ROLLUP
```

Requête 4 :
Retourne le nombre de documents dans chaque langues

```
SELECT domainValue, languageValue, nbDocument
FROM (
SELECT domainCode, languageCode, COUNT(*) AS nbDocument
FROM (SELECT halId, languageCode
FROM MonoNoyau) AS L,
(SELECT domainCode, halID
FROM faits
GROUP BY domainCode, halId) AS D
WHERE L.halId = D.halId
GROUP BY domainCode
ORDER BY domainCode,nbDocument DESC
) AS tmp NATURAL JOIN langues NATURAL JOIN domaines
ORDER BY nbDocument DESC;
```

Résultat :

languageValue	nbDocuments
English	1241
French	693
Spanish; Castilian	43
Italian	8
German	7
Japanese	3
Vietnamese	2
Arabic	1
Portuguese	1
Hungarian	1

Requête 5 :
Retourne le nombre de documents dans chaque langue, par domaine de recherche.

```
SELECT domainValue, docTypeValue, nbDocument
FROM (
SELECT domainCode, docTypeCode, COUNT(*) AS nbDocument
FROM (SELECT hallId, docTypeCode
FROM MonoNoyau) AS L,
(SELECT domainCode, halID
FROM faits
GROUP BY domainCode, hallId) AS D
WHERE L.hallId = D.hallId
GROUP BY domainCode
ORDER BY domainCode,nbDocument DESC
) AS tmp NATURAL JOIN types NATURAL JOIN domaines
ORDER BY nbDocument DESC;
```

Résultats : Voici les dix premiers tuples de la sortie :

domainValue	languageValue	nbDocument
Sciences de l'Homme et Société	French	858
Sciences de l'Homme et Société/Sociologie	Spanish; Castilian	362
Sciences de l'ingénieur [physics]	English	313
Informatique [cs]	English	308
Sciences de l'Homme et Société/Anthropologie sociale et ethnologie	Spanish; Castilian	283
Sciences de l'Homme et Société/Anthropologie biologique	Spanish; Castilian	267
Physique [physics]	English	217
Sciences du Vivant [q-bio]	English	213
Sciences de l'Homme et Société/Histoire	French	89

Requête 6 :
Cette requête crée le cube sur les données monovaluées de la table des faits.

```
SELECT hallId, docTypeCode, languageCode, producedDate, COUNT(*) AS nb
FROM monoNoyau
GROUP BY hallId, docTypeCode, languageCode, producedDate WITH ROLLUP

UNION

SELECT hallId, docTypeCode, languageCode, producedDate, COUNT(*) AS nb
FROM monoNoyau
```

```
GROUP BY docTypeCode, languageCode, producedDate, halId WITH ROLLUP
UNION
SELECT halId, docTypeCode, languageCode, producedDate, COUNT(*) AS nb
FROM monoNoyau
GROUP BY languageCode, producedDate, halId, docTypeCode WITH ROLLUP
UNION
SELECT halId, docTypeCode, languageCode, producedDate, COUNT(*) AS nb
FROM monoNoyau
GROUP BY producedDate, halId, docTypeCode, languageCode WITH ROLLUP
```

Résultats : Voici les dix premiers tuples de la sortie :

domainValue	docTypeValue	nbDocument
Sciences de l'Homme et Société	Communication dans un congrès	858
Sciences de l'Homme et Société/Sociologie	Chapitre d'ouvrage	362
Sciences de l'ingénieur [physics]	Article dans une revue	313
Informatique [cs]	Pré-publication, Document de travail	308
Sciences de l'Homme et Société/Anthropologie sociale et ethnologie	Chapitre d'ouvrage	283
Sciences de l'Homme et Société/Anthropologie biologique	Chapitre d'ouvrage	267
Physique [physics]	Article dans une revue	217
Sciences du Vivant [q-bio]	Communication dans un congrès	213
Sciences de l'Homme et Société/Histoire	Article dans une revue	89

Requête 7 :
Cette requête affiche le nombre de document en français, c'est une requête simple mais qui montre la puissance du Cube construit précédemment, car c'est le cube qui permet de faire cela simplement !

```
SELECT *
FROM monofaitscube
WHERE halID IS NULL AND docTypeCode IS NULL AND producedDate IS NULL AND languageCode = "fr";
```

Résultat :

halId	docTypeCode	languageCode	producedDate	nb
NULL	NULL	fr	NULL	693

Requête 8 :
Cette requête retourne le nombre de document avec l'équivalent d'un Rollup sur Année et languageCode

```
SELECT YEAR(producedDate) AS Année, languageCode, SUM(nb) AS nbDocument
FROM MonoFaitsCube
WHERE halId IS NULL AND docTypeCode IS NULL
GROUP BY Année DESC, languageCode DESC;
```

Requête 9 :
Cette requête permet de classer les documents selon l'année de publication
Le but ici est d'imiter le système qui permet de trier les documents sur le site HAL.

```
SELECT YEAR(producedDate) AS Année, halId, title
FROM monofaitscube NATURAL JOIN documents
WHERE year(producedDate) IS NOT NULL
GROUP BY halID
ORDER BY Année DESC;
```

Résultats : Voici les dix premiers tuples de la sortie :

YEAR(producedDate)	halId	title
2019	halshs-01633164	Against the PredP theory of small clauses
2018	hal-01632626	« Participer » à la construction d'indicateurs. Technique de quantification et formes de la démocratie
2018	hal-01633127	Magnetron-sputtered copper/diamond-like composite thin films with super anti-corrosion properties
2018	hal-01633714	On the crystallographic\, stage I-like\, character of fine granular area formation in internal fish-eye fatigue cracks
2018	hal-01633729	Recent developments in nanostructured inorganic materials for sorption of cesium and strontium: Synthesis and shaping\, sorption capacity\, r
2018	hal-01632414	Learning Multi-Modal Word Representation Grounded in Visual Context
2018	hal-01632274	Uncertainty theory as a basis for belief reliability
2018	hal-01633203	Healthy New Towns. Représenter la ville santé en Grande-Bretagne.
2018	hal-01632091	Contemporary Daoist Funerary Practices and Afterlife Beliefs

Requête 10 :

Cette requête permet de classer les documents selon le type du document
Le but ici est d'imiter le système qui permet de trier les documents sur le site HAL.

```
SELECT docTypeCode, docTypeValue, halId, title
FROM monofaitscube NATURAL JOIN documents NATURAL JOIN types
WHERE docTypeCode IS NOT NULL AND docTypeValue IS NOT NULL
GROUP BY halID
ORDER BY docTypeCode DESC;
```

Résultats : Voici les dix premiers tuples de la sortie :

docTypeCode	docTypeValue	halId	title
UNDEFINED	Pré-publication, Document de travail	hal-01634080	Wideband low-profile cavity-backed V-folded bowtie antenna for ground penetrating radar: design\, computational modelii
UNDEFINED	Pré-publication, Document de travail	hal-01584963	Approximation of a Brittle Fracture Energy with a Constraint of Non-Interpenetration
UNDEFINED	Pré-publication, Document de travail	halshs-01370816	Wage Inequality and Skill Supplies in a Globalised World
UNDEFINED	Pré-publication, Document de travail	hal-01634075	Semi–analytical method for the identification of inclusions by air–cored coil interaction in ferromagnetic media
UNDEFINED	Pré-publication, Document de travail	hal-01633507	Big Data and HPC collocation: Using HPC idle resources for Big Data Analytics
UNDEFINED	Pré-publication, Document de travail	hal-01633423	A Hamilton-Jacobi-Bellman approach for the numerical computation of probabilistic state constrained reachable sets
UNDEFINED	Pré-publication, Document de travail	halshs-01631494	Confirmation bias and signaling in Downsian elections
UNDEFINED	Pré-publication, Document de travail	hal-01632943	Airy point process at the liquid-gas boundary
UNDEFINED	Pré-publication, Document de travail	hal-01633361	Analytical results for two users' forecast scheduling