

Refonte de la base de données

Ce document présente la refonte de la base de données avec les différentes tables mises en place. Nous partirons de la solution initiale pour expliquer le changement de certaines tables.

Table des matières

Définition des tables	2
Types de tables	2
Alias de table	2
Définition des clés étrangères	2
Glossaire	2
Schéma	3
Explication des tables : Partie analytics	3
Utilisateurs.....	3
Departements & Location	3
Groupes	Erreur ! Le signet n'est pas défini.
Analytics	5
Explication des tables : Partie application	6
App users	6
Facets owners	6
Facet department.....	7
Facet location.....	7
Facet group	7
Alimentation	8
Users.....	8
Groups	9
User_groups.....	10
Activity.....	10
Optimisations	12
Triggers	12
Table de « faits »	12

Définition des tables

Les tables sont nommées de la manière suivante :

<TABLE TYPE> _ <TABLE NAME> _ <TABLE ALIAS>

Types de tables

Il existe 4 types de table différents :

- T : Table regroupant des données
- TR : Table de référence regroupant des données dites statiques et qui ne bougeront pas dans le temps
- TJ : Table de jointure regroupant 2 tables
- VF : Vue de faits ou vue factorisée

Alias de table

L'alias de table est utilisé pour les noms de colonnes

Définition des clés étrangères

Concernant les clés étrangères, les valeurs physiques sont :

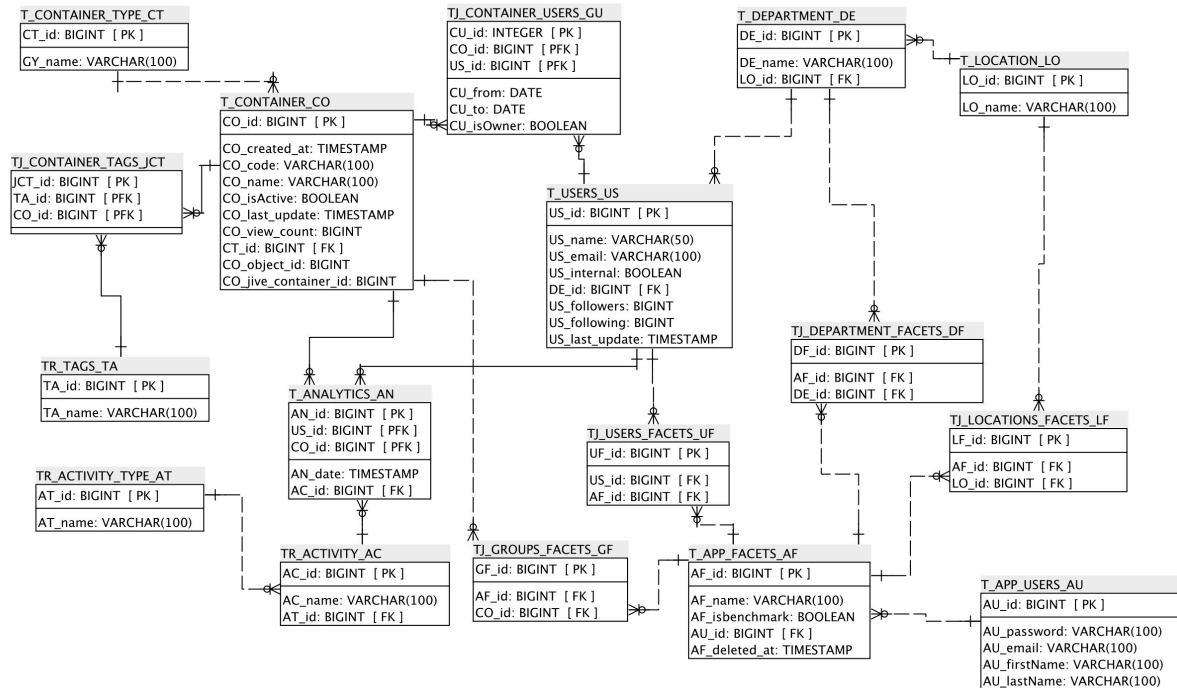
<Table source > _ <Table_dest + column>_FK

Lorsqu'une table dispose d'une clé étrangère, cette clé possède le nom de colonne de sa table source.

Glossaire

AI : Auto Increment

Schéma



Le schéma ci-dessus représente la refonte de la base de données.

Explication des tables : Partie analytics

Utilisateurs

La table T_USERS_US comporte les champs suivants :

Champ	Type	Comment
US_id	BIGINT	Clé primaire
US_name	VARCHAR(100)	Nom + prénom
US_email	VARCHAR(100)	
DE_id	BIGINT	Clé étrangère département
US_followers	BIGINT	
US_following	BIGINT	
US_last_update	TIMESTAMP	

Departements & Location

Un utilisateur appartient à un et à un seul département. Ce dernier correspond à une unique localisation

T_DEPARTEMENT_DE

Champ	Type	Comment
DE_id	BIGINT	Clé primaire AI
DE_name	VARCHAR(100)	
LO_id	BIGINT	Clé étrangère Location

T_LOCATION_LO

Champ	Type	Comment
LO_id	BIGINT	Clé primaire AI
LO_name	VARCHAR(100)	

Containers

Les groupes sont les containers de l'application Jive, ils sont liés à des utilisateurs, des types de groupe ainsi qu'à des tags.

T_CONTAINERS_CO

Champ	Type	Comment
CO_id	BIGINT	Clé primaire
CO_created_at	TIMESTAMP	
CO_code	VARCHAR(100)	Alias appli jive
CO_name	VARCHAR(100)	
CO_isActive	BOOLEAN	Default TRUE
CO_view_count	BIGINT	
CY_id	BIGINT	Clé étrangère group type
CO_object_id	BIGINT	Id container bdd jive
CO_jive_group_id	BIGINT	Identifiant api jive

Types de container

T_CONTAINER_TYPE_CY

Champ	Type	Comment
CY_id	BIGINT	Clé primaire
CY_name	VARCHAR(100)	

Tags

TR_TAGS_TA

Champ	Type	Comment
-------	------	---------

TA_id	BIGINT	Clé primaire AI
TA_name	VARCHAR(100)	

Tags des groupes

TJ_CONTAINER_TAGS_CT

Champ	Type	Comment
CT_id	BIGINT	Clé primaire AI
TA_id	BIGINT	FK tag
CO_id	BIGINT	FK groupe

Analytics

La partie analytics correspond aux résultats retourné par l'API jive dont nous verrons les données plus tard dans ce document.

T_ANALYTICS_AN

Champ	Type	Comment
AN_id	BIGINT	Clé primaire AI
US_id	BIGINT	FK user
GR_id	BIGINT	FK groupe
AN_date	TIMESTAMP	Date activité
AC_id	BIGINT	FK type activité

Activité

Une activité correspond à une action réalisée sur Jive.

T_ACTIVITY_AC

Champ	Type	Comment
AC_id	BIGINT	Clé primaire AI
AC_name	VARCHAR(100)	
AT_id	BIGINT	FK type d'activité

Type d'activité

Une activité concerne un type précis d'activité, grâce auquel on va pouvoir grouper les types d'actions

TR_ACTIVITY_TYPE_AT

Champ	Type	Comment
AT_id	BIGINT	Clé primaire AI
AT_name	VARCHAR(100)	

Explication des tables : Partie application

L'application dispose d'utilisateurs qui peuvent interagir avec l'application (login/register)

App users

T_APP_USERS_AU

Champ	Type	Comment
AU_id	BIGINT	Clé primaire AI
AU_password	VARCHAR(100)	
AU_email	VARCHAR(100)	
AU_firstName	VARCHAR(100)	
AU_lastName	VARCHAR(100)	

Il est possible pour un utilisateur d'enregistrer des facets, ces facets concernent les tables suivantes :

- T_USERS_US : Filtre sur les owners
- T_DEPARMENT_DE : Filtre sur les départements
- T_LOCATION_LO : Filtre sur les localisations
- T_GROUPS_GR : Filtre sur les communauté

Facet

T_APP_FACETS_AF

Champ	Type	Comment
AF_id	BIGINT	Clé primaire AI
AF_name	VARCHAR(100)	Nom enregistré
AF_isBenchmark	BOOLEAN	
AU_id	BIGINT	FK app user
AF_deleted_at	TIMESTAMP	

Facets owners

On appelle cette facette owners car le filtre ne concerne que les propriétaires d'une communauté

TJ_USERS_FACETS_UF

Champ	Type	Comment
UF_iid	BIGINT	Clé primaire AI
US_id	BIGINT	FK users
AF_id	BIGINT	FK facet

Facet department

TJ_DEPARTMENT_FACETS_DF

Champ	Type	Comment
DF_id	BIGINT	Clé primaire AI
DE_id	BIGINT	FK department
AU_id	BIGINT	FK facet

Facet location

T_LOCATION_FACETS_LF

Champ	Type	Comment
UF_id	BIGINT	Clé primaire AI
LO_id	BIGINT	FK location
AF_id	BIGINT	FK facet

Facet group

T_GROUP_FACETS_GF

Champ	Type	Comment
UF_id	BIGINT	Clé primaire AI
GR_id	BIGINT	FK group
AF_id	BIGINT	FK facet

Les tables des facets sont alimentées lorsque l'utilisateur décide de sauvegarder sa configuration.

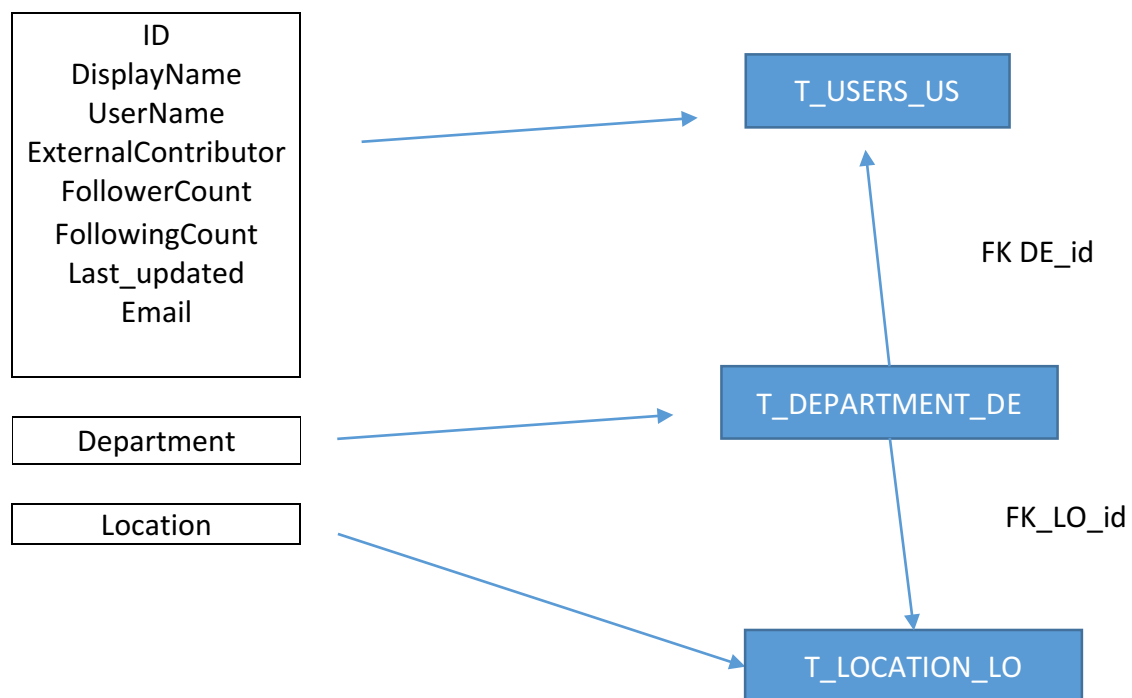
Alimentation

L'alimentation de la base se fait grâce à des appels vers l'API Jive et Analytics pour l'activité.

L'API Jive permet d'alimenter certaines parties de la bdd.

Users

Appel depuis l'api users :



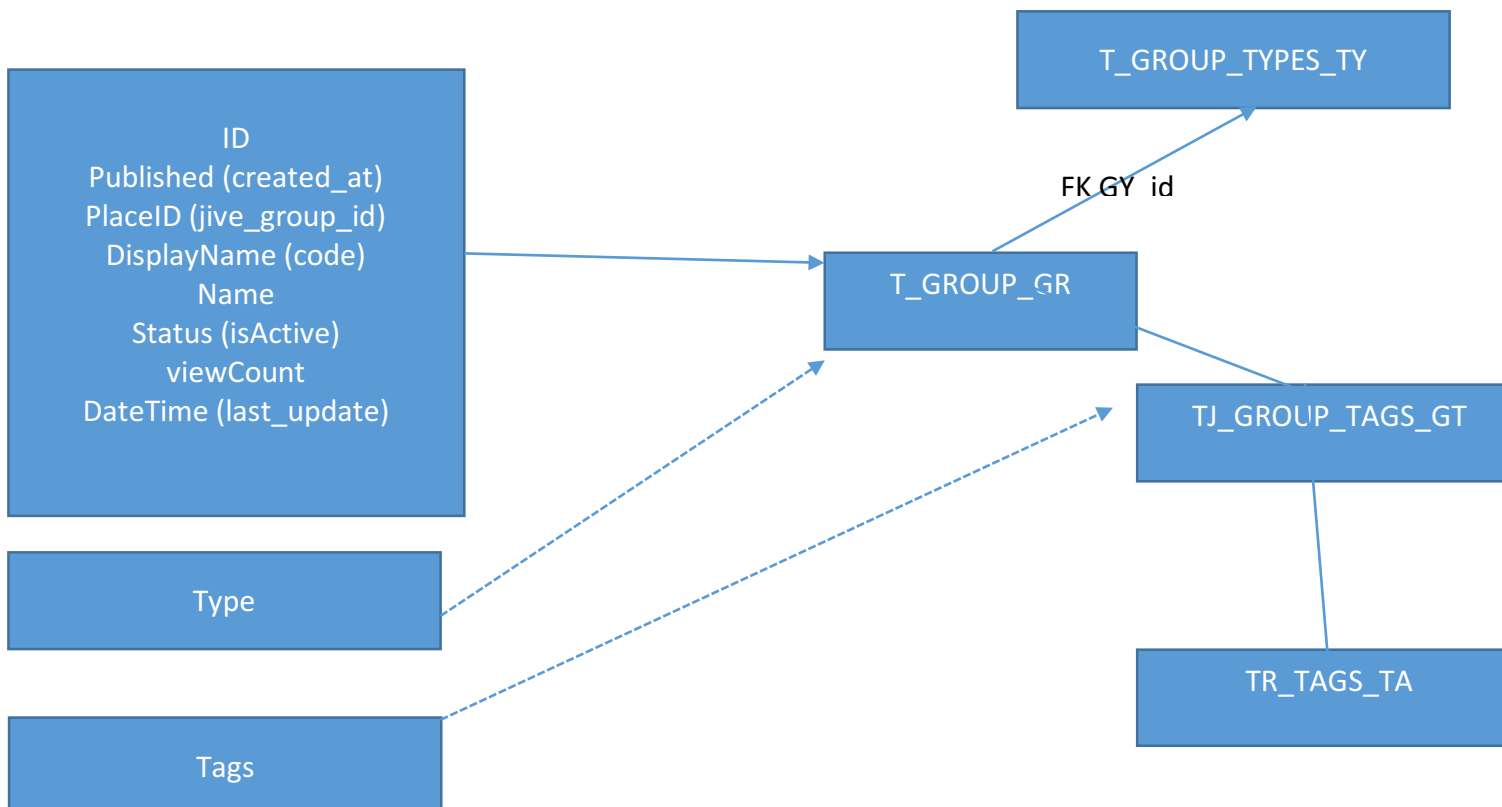
A chaque entrée d'un nouvel utilisateur, on vérifie si le département est bien présent dans la table département, sinon on l'ajoute, de même pour la location.

Groups

Les champs récupérés sous forme de requête sont les suivants :

ID	ID groupe bdd jive
Published	Date publication
PlaceID	Id container api
DisplayName	Alias du nom pour jive (ex : type-de-blog)
Name	Ex : Type de blog
Parent	Container parent
Status	isActive ?
viewCount	viewCount
followerCount	
Type	
Tags	Array
DateTime	Last update date

Le traitement des données de l'api étant fait en amont, on s'intéresse ici aux données utiles au remplissage de la base



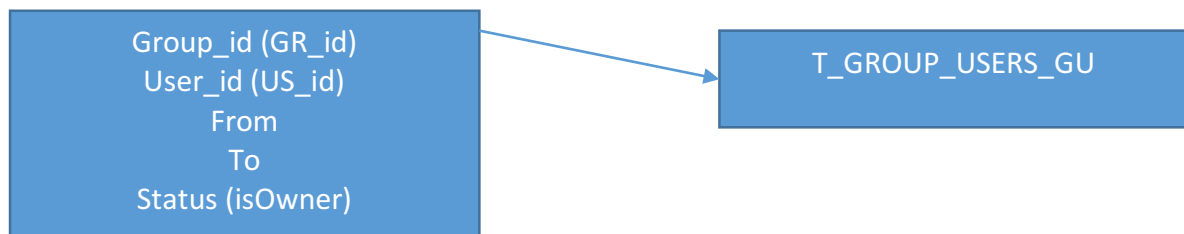
Pour clarifier, un container correspond à 1 type de groupe. Il existe 4 types de groupe :

- Blog
- Project
- Group
- Space

Un container peut contenir 1 ou plusieurs tags. Il n'est actuellement pas possible de récupérer la liste de tous les tags utilisés dans Jive. Ainsi, il faudra avant chaque insertion vérifier si le tag est bien présent dans la table TR_TAGS_TA avant de l'ajouter à la table de jointure TJ_GROUP_TAGS_GT

User_groups

Il existe un traitement effectué sur dataiku qui permet de connaître quels utilisateurs sont associés à des groupes, le résultat est de la forme :



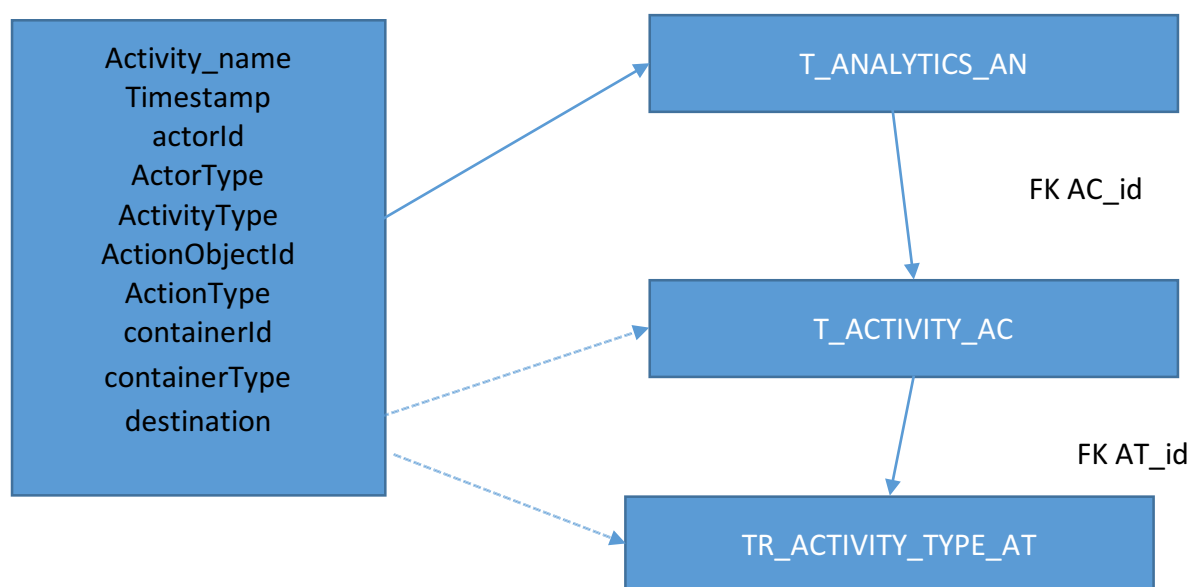
Activity

Ces données proviennent de l'api analytics, elles sont de la forme suivante :

Activity_name	
Timestamp	
actorId	Id utilisateur ayant fait l'action
ActorType	
ActivityType	
ActionObjectId	L'objet concerné par l'action
ActionType	Le type d'objet sur lequel a été effectué l'action
containerId	L'id du container pour une action de groupe
containerType	
destination	

On constate qu'il existe des types d'activités ainsi que des noms d'activités, c'est avec ces données que nous allons alimenter la table TR_ACTIVITY_TYPE_AT et T_ACTIVITY_AC

Dans la partie analytics, nous nous intéressons qu'aux actions effectuées sur les groupes : A confirmer



Il n'est pas possible de récupérer la liste de toutes les activités possibles ni le type d'activité associé, il faut donc compléter les tables T_ACTIVITY_AC et TR_ACTIVITY_TYPE_AT dès qu'une nouvelle activité est enregistrée.

La table T_ANALYTICS_AN est liée à un utilisateur et à un groupe, c'est toujours le cas pour une activité.



Il se peut qu'une activité ne concerne pas un groupe particulier comme pour le login d'un utilisateur par exemple. Il faut donc définir un id de groupe pour ce genre de cas.

Optimisations

Triggers

Création de triggers pour les insertions pour les cas suivants :

- Nouveau tag
- Nouvelle activité / type d'activité
- Nouveau département
- Nouvelle localisation

Table de « faits »

Les tables de faits ont pour but de réduire les calculs qui peuvent être effectués lors des requêtes. Cela permet de réduire drastiquement le temps de calcul tout en disposant de tous les résultats possibles.

Dans notre cas, les tables de fait pourront être modalisé par des vues. Dans notre application, on constate que les calculs sont redondants pour le calcul du nombre d'actif et d'enregistrés. Cependant, il est intéressant de regrouper le nombre d'actifs et d'enregistrés pour toutes les filtres possibles.