

IBDR - Groupe SAR - Stratégie de répartition

Ce document présente les différentes stratégies de répartition adoptées pour la base de données vidéo club du groupe SAR. Plutôt que de présenter le document table par table, nous présentons chaque stratégie, en détaillant à chaque fois les tables concernées.

Dans cette présentation, on suppose que notre base de données est formée d'un nombre $N+1$ non précisé (extensible) de sites, dont un site qu'on appellera le "siège social", et N sites qu'on appellera les "succursales".

Stratégie 1 : Table à état global géré par le serveur central (Siège social), réplication totale

Dans cette stratégie, l'état des entités concernées est global - toutes les succursales ont la même vision de la base de données - et géré par le siège social : c'est le siège social qui à chaque fois s'occupe de créer/mettre à jour les tuples dans la base de données.

Tables concernées

- Toutes les tables liées aux Films en tant qu'entité abstraite. En effet on veut partager l'information entre toutes les succursales à ce niveau. Les tables concernées sont : **Film**, **FilmProducteur**, **FilmActeur**, **FilmRealisateur**, **FilmDistinction**, **TypeDistFilm**, **TypeDistActeur**, **TypeDistinction**, **TypeDistReal**, **RealisateurDistinction**, **ActeurDistinction**, **Personne**, **FilmGenre**, **Genre**, **Langue**.
- Toutes les tables liées aux éditions : **EditionLangueSousTitres**, **EditionLangueAudio**, **Edition**, **EditeurEdition**, **Editeur**, **Pays**.
- La table **TypeAbonnement**. En effet, pour que une personne abonnée dans une succursale puisse louer sans encombre des films dans une autre succursale, les types d'abonnements sont standardisés.
- La table **Client**. En effet, de manière à simplifier la migration éventuelle des abonnements et la gestion de la liste noire des clients, nous avons décidé de gérer cette table de manière centralisée.

Changements de clé primaire

Les tables qui suivent cette stratégie n'ont pas à subir de modifications structurelles. En effet, comme elles sont gérées uniquement par le siège social, elles fonctionnent comme dans une base de données centralisée.

Type de réplication

Les tables utilisant cette stratégie seront répliquées sur toutes les succursales. Le type de réplication utilisé sera la réplication transactionnelle. Ce type de réplication est adapté dans les cas ou :

1. Une seule table est à l'origine de toutes écritures sur les tables (dans ce cas, le siège social)
2. Les changements doivent être répliqués rapidement dans les succursales

Ce n'est pas le seul type de réplication adapté, la réplication par snapshot pourrait également être adaptée, selon les contraintes de réactivité de l'application. La réplication transactionnelle permet une très bonne réactivité, mais est plus coûteuse en terme de ressources. Dans notre cas, la table client par exemple, nécessite la réplication transactionnelle. En effet, on veut que les modifications impactent immédiatement toutes les répliques.

On a plus de flexibilité pour les autres tables, mais nous avons décidé de gérer toutes les réplifications de manière transactionnelle dans un premier temps.

Stratégie 2 : Table locale avec clé primaire composite, réplication au besoin

Dans cette stratégie, l'état des entités est local, mais on a la nécessité de distinguer une entité d'une succursale de l'entité d'une autre succursale.

Tables concernées

La table concernée est la table **Abonnement**. En effet, on considère qu'un client loue principalement dans son vidéo-club ou il s'est inscrit, et qu'il n'y a donc pas besoin de stocker son abonnement de manière centralisée. Cependant, il faut quand même qu'un client aie la possibilité de louer dans une autre succursale que celle où il s'est inscrit.

Changements de clé primaire

On va rajouter un champs "succursale" à la clé primaire de la table, désignant directement la succursale où il a été créé. On intégrera ce champs à la clé primaire.

Le champ supplémentaire a deux utilités :

- Il permet à une succursale de retrouver la succursale d'origine d'un abonnement, de manière à récupérer ses informations.
- Il permet de maintenir l'unicité des entités Abonnement tout en gardant un ID autogénéré simple.

Type de réplication

On utilisera dans ce cas une réplication de type merge. On sait qu'on aura jamais de conflits, car un client ne peut se trouver dans deux succursales en même temps. Cette stratégie flexible permet que :

- Les base de données abonnées au publieur (succursale d'origine du client) puissent modifier l'abonnement
- Les base de données abonnées ne reçoivent que les changements qui les concernent. Si un client n'a jamais été dans une succursale, elle ne recevra pas les changements le concernant.

Stratégie 3 : Table locale avec clé primaire “uniqueidentifier”, non répliquée

Pour les cas où on veut maintenir l'unicité d'un objet local au niveau global, on peut utiliser un **uniqueidentifier** comme clé primaire en lieu et place d'un id auto généré.

Tables concernées

Nous avons opté pour cette solution dans le cas de la table **FilmStock**. En effet, on veut que les entités soient distinguées les unes des autres (pour que un client ne puisse pas retourner un film dans la mauvaise succursale par exemple), mais on a pas besoin de connaître la succursale d'origine. La solution plus simple de l'uniqueidentifier est donc adaptée.

Changements de clé primaire

Il suffit juste de modifier la clé primaire pour utiliser un uniqueidentifier en lieu et place d'un id auto généré de type identity.

Type de réplication

Pas de réplication

Stratégie 4 : Table locale sans modification par rapport au schéma centralisé

Dans certains cas, on a pas de nécessité de changer la modélisation par rapport au schéma centralisé, notamment pour les locations, où l'id d'une location n'est jamais utilisé pour les interactions avec les utilisateurs, ou dans le cas des relances, qui sont des entités liées à d'autres entités locales.

Tables concernées

Les tables **Location**, **RelanceRetard**, **RelanceDécouvert**.

Changements de clé primaire

Pas de changements

Type de réplication

Pas de réplication

