

Réplication Introduction

MySQL —

I► Nous allons voir dans cet article différents cas de réplication MySQL .

I – La réplication classique UniDirectionnelle:

Les bases de données du serveur Maître (N°1) sont recopiées en temps réel vers le serveur Esclave (N°2)



Comment se gère un tel modèle :

Cas N°1 :

Les requêtes d'écriture se font sur le serveur N°1

Les requêtes de lecture se font sur le serveur N°2

Ce qui permet de répartir les charges **en lecture et en écriture**

Cas N°2 :

Le serveur N°1 est considéré comme le serveur en production

Le serveur N°2 est considéré comme un **Backup Server**

Cas N°3 :

Mix des 2...

Si il y a des conflits d'insert avec des clés identiques (**1062. « DUPLICATE ENTRY FOR... »**) , la réplication va s'arrêter net et l'on risque de perdre beaucoup d'informations de sauvegarde.

Si le serveur maître s'arrête pour des raisons x (dump ou lock

...1053. « Query partially completed on the master ...), la réplication s'arrête également sur le serveur esclave et ne reprends pas.

Il vaut mieux les ignorer automatiquement avec la directive :
slave-skip-errors=1062,1053

Ou mieux gérer l'incrémentation des clés(ID) différemment sur chaque serveur

Sur le serveur 1 :

```
auto_increment_increment      = 10
auto_increment_offset         = 1
```

Sur le serveur 2 :

```
auto_increment_increment      = 10
auto_increment_offset         = 2
```

II – La réplication Croisée:



Dans ce cas de figure, les deux serveurs sont à la fois Maître et Esclave et la synchronisation se fait dans les deux sens;

Optimisation de la charge et Sécurisation des données !