

## gDTC - Anomalie #1998

### Migration des tables du format MyISAM vers InnoDB

24/08/2007 10:21 - Sébastien Dinot

<b>Statut:</b>	Un jour peut-être	<b>Début:</b>	24/08/2007
<b>Priorité:</b>	Faible	<b>Echéance:</b>	
<b>Assigné à:</b>	Benjamin Drieu	<b>% réalisé:</b>	0%
<b>Catégorie:</b>		<b>Temps estimé:</b>	0.00 heure
<b>Version cible:</b>			

**Description**

Il y a au moins 2 **excellentes** raisons pour préférer dans notre cas (et dans la plupart des cas) le format InnoDB au format MyISAM :

1. InnoDB supporte les transactions, ce qui limite le risque d'inconsistance de la base (lorsque qu'on passe d'un état cohérent des données à un autre par le jeu de plusieurs requêtes successives et que les états intermédiaires fournissent une vue erronée des informations).
2. InnoDB supporte les clés étrangères et les contraintes d'intégrité qui vont avec (ON [UPDATE|DELETE] CASCADE...).

Développons un peu ce dernier point : via les clés étrangères, on crée des dépendances entre les enregistrements de tables différentes ou non :

- La mise à jour d'une clé dans un enregistrement provoque la mise à jour de tous les enregistrements qui en dépendent.
- La suppression d'un enregistrement provoque la suppression de tous les enregistrements qui en dépendent.

Dans notre cas, avec de telles contraintes d'intégrité, lorsque nous demanderions au SGBD de supprimer une personne, il supprimerait de lui-même les cotisations, les adhésions, les mails, etc. qui se réfèrent à cette personne : que du bonheur !

#### Historique

##### #1 - 24/08/2007 14:54 - Benjamin Drieu

On fait comment pour migrer ?

##### #2 - 24/08/2007 15:23 - Sébastien Dinot

La solution la plus simple est décrite ici :

<http://www.linux.com/articles/46370>

NB: contrairement à ce qui est écrit dans l'article, avec les versions actuelles de MySQL, il faut rechercher et remplacer les occurrences de « ENGINE=MyISAM » par « ENGINE=InnoDB » et non « TYPE=... ».

En plus de modifier les formats de stockage, il faudrait spécifier les clés étrangères en précisant les contraintes d'intégrité :

```
-----
CREATE TABLE actor
(
  actor_id INT NOT NULL AUTO_INCREMENT,
  [...]
) ENGINE=InnoDB DEFAULT CHARSET=latin1;

CREATE TABLE membership
(
  membership_id INT NOT NULL AUTO_INCREMENT,
  actor_id INT NOT NULL,
  [...]

CONSTRAINT membership_actor_id_reference
  FOREIGN KEY (actor_id)
  REFERENCES actor (actor_id)
  ON UPDATE CASCADE
  ON DELETE CASCADE,
  [...]
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
-----
```

=> Si un id d'acteur est mis à jour dans la table « actor », la référence est mise à jour dans tous les enregistrements correspondants de la table «

membership ».

=> Si un acteur est supprimé dans la table « actor », tous les enregistrements de la table « membership » qui le référence sont détruits aussi.

### **#3 - 31/10/2017 17:40 - Benjamin Drieu**

- *Statut changé de Confirmé à Un jour peut-être*

En fait, le gros point de cette migration est:

- ais-je envie de coder des transactions qui vont bien dans le code existant ?
- ais-je envie de mettre en place les clefs foreign et passer un peu de temps à débunker tout ce qui en découlera ?