



Test de la structure de la base via "pragma integrity_check"

Dans une base de données SQLite, il arrive parfois que l'on soit amené à exécuter la commande PRAGMA integrity_check. Cette opération est un outil de vérification interne qui permet de s'assurer que la base de données est cohérente et qu'aucune corruption n'est présente. Même si SQLite est réputé pour sa robustesse et sa fiabilité, plusieurs situations peuvent entraîner des problèmes :

- Arrêts imprévus ou plantages : Si le système ou l'application qui utilise la base de données s'arrête brutalement, certaines écritures peuvent rester incomplètes et créer des incohérences.
- Problèmes matériels : Une défaillance du disque, des secteurs corrompus ou des problèmes de mémoire peuvent altérer les données stockées.
- Mises à jour ou migrations : Lorsqu'on migre ou modifie la structure d'une base existante, des erreurs peuvent survenir si les scripts ou outils de migration ne sont pas parfaitement fiables.
- Multiples accès concurrents : Bien que SQLite gère le verrouillage, des accès simultanés mal coordonnés peuvent provoquer des anomalies.

La commande **PRAGMA integrity_check** va parcourir toutes les tables et index de la base, vérifier les relations internes, les contraintes et s'assurer que les pages du fichier de base de données sont cohérentes. Elle renvoie « ok » si tout est correct, ou signale des erreurs précises en cas de corruption.

Dans de nombreux cas, les erreurs détectées par PRAGMA integrity_check sont liées à des index corrompus plutôt qu'aux données elles-mêmes. SQLite fournit alors une solution simple : la commande REINDEX. En réindexant la base de données (ou des index spécifiques), on reconstruit les structures des index à partir des données réelles, ce qui résout souvent les incohérences détectées sans perdre de données.

Correctif

Si une erreur de ce type est détectée, le correctif proposé est de réindexer la base grâce à la commande SQLite **REINDEX**

Si l'erreur persiste ne pas continuer à utiliser la base et informer au plus vite les techniciens de Logeas Informatique Cette approche permet de maintenir la fiabilité d'une base SQLite, même lorsqu'elle a été soumise à des conditions d'utilisation difficiles, tout en limitant le risque de perte de données.

Qu'est-ce qu'un index dans une base de données ?

Un index est une structure spéciale qui améliore la rapidité des recherches et des tris dans une table. On peut le comparer à l'index à la fin d'un livre : au lieu de parcourir toutes les pages pour trouver un mot, on regarde directement dans l'index pour savoir où aller.

Dans une base SQLite

- Chaque table peut avoir zéro ou plusieurs index.
- Un index est créé sur une ou plusieurs colonnes, par exemple `Nom` ou `Email`.
- Il stocke les valeurs de ces colonnes dans un ordre particulier, avec un pointeur vers les lignes correspondantes de la table.
- Lorsqu'une requête fait une recherche ou un tri sur ces colonnes, SQLite peut utiliser l'index pour accéder rapidement aux données, sans lire toute la table.

Qu'est-ce que REINDEX ?

La commande **REINDEX** permet de reconstruire les index d'une base SQLite. Les index peuvent parfois se corrompre à cause :

- d'arrêts imprévus ou de plantages,
- de problèmes matériels (disque ou mémoire),
- ou de modifications structurelles de la base (migration, scripts).

REINDEX recrée les index à partir des données réelles, ce qui répare la plupart des incohérences détectées par `PRAGMA integrity_check`, sans toucher aux données elles-mêmes.

Syntaxe de base

- Réindexer toute la base :

```
```sql
```

```
REINDEX;
```
```

* Réindexer une table spécifique (tous ses index) :

```
```sql  
REINDEX nom_table;
```
```

* Réindexer un index précis :

```
```sql  
REINDEX nom_index;
```
```

Exemple concret

Supposons que nous avons une table `Clients` :

```
```sql  
CREATE TABLE Clients (
 ID INTEGER PRIMARY KEY,
```

```
Nom TEXT,
Email TEXT
);
```

```
CREATE INDEX idx_clients_nom ON Clients(Nom);
```

From:

<https://wiki-logeas.fr/log eas-web/> - **Documentation de Logeas-web.fr (client léger)**

Permanent link:

<https://wiki-logeas.fr/log eas-web/doku.php?id=admin:testbase:test001>

Last update: **2026/01/23 09:55**

