

Procédure de réparation/optimisation des bases sqlites

Etape 1 : mise de la base en maintenance

Si un problème est détecté sur une base, il va falloir couper son accès. Il convient donc de prévenir les utilisateurs de la maintenance (téléphone, newsletters...)

Etape 2 : sauvegarde

On ne travaille **jamais** sur les bases directement.

1. Si la base est en cours d'utilisation, il faut la "démonter depuis l'assistance", ou peut aussi en profiter pour faire une version non cryptée (ne marche pas toujours)
2. Dupliquez le fichier de la base
3. Créer un dossier temporaire et y glisser les bases originales
4. Rapatrier la base sur votre poste. Le travail directement sur le serveur est déconseillé, surtout sur les grosses bases (en plus cela fait un niveau de sauvegarde supplémentaire)

Etape 3 : décryptage

Si vous n'aviez pas décrypté la base lors de l'étape de sauvegarde, vous pouvez le faire via l'utilitaire en récupérant la clef dans la base de cryptage. **A partir de là on travaille sur la base décryptée**

Etape 4 : Commande test

Dans un premier temps on travaille de préférence avec **SynDBExplorer**

Lancer la commande	<code>PRAGMA integrity_check;</code>	Le traitement peut être long sur de grosses bases La réponse "correcte" est OK Dans le cas contraire on peut tenter une commande de réindexation et réessayer
Lancer la commande	<code>REINDEX;</code>	Le traitement peut aussi être long sur de grosses bases S'il n'y a pas d'erreur, la commande ne répond rien (fin de changement du curseur)

Piste de réparation d'une base cassée

Créer une sauvegarde sql puis restaurer

Sauvegarde

Utilisez l'utilitaire sqlite3 en ligne de commande. La commande .recover est disponible depuis la version 3.29.0 et est plus robuste que .dump. Si .recover n'est pas disponible, utilisez l'ancienne méthode :

```
sqlite3 base_corrompue.s3dbBrut ".dump" > reparation.sql
```

Restauration

Reconstruction de la base. Une fois le fichier SQL généré, injectez-le dans un nouveau fichier de base de données.

NB : il faut enlever le startTransaction et le commit à la fin

autre piste :

⚠ Si base_corrompue.s3db est réellement corrompue, .dump peut échouer ou produire un dump incomplet. Pour tenter de récupérer un maximum de données, tu peux aussi utiliser

```
sqlite3 base_corrompue.s3db ".recover" > reparation.sql  
sqlite3 base_reparee.s3db < reparation.sql
```

.recover est généralement plus adapté à une base endommagée.

Etape 5 : Optimisation de la base

Fichier à effacer

On peut lancer sans danger la commande	Drop Table TableDeleted;
On ne peut pas effacer la piste d'audit, mais on peut	1. L'enregistrer à côté de la base Pour ce faire, enregistrer la piste en entier au format csv Repérer l'ID d'une piste ayant disons 3 ans Effacer Delete from PisteAudit Where ID<????? Il est conseillé d'utiliser SynDbExplorer même si on regarde l'ID dans DBBrowser Finir par un REINDEX puis un VACUUM

Etape 6 : Nettoyage

Après avoir remis la base en ligne, on ne laisse que le fichier de la piste d'audit avec la date de sauvegarde.

On efface les autres fichiers afin de ne pas amplifier les sauvegardes.

On garde sur son poste les versions de travail quelques jours en attente de vérification du client

Etape 7 : Information client

On indique au client :

1. les opérations faites
2. que la base est de nouveau en accès
3. on lui demande de tester que tout est OK

From:

<https://wiki-logeas.fr/certif/> - dokuwiki-certif

Permanent link:

<https://wiki-logeas.fr/certif/doku.php?id=certif:do:procedurereparationbase&rev=1790251427>

Last update: **2026/09/24 14:03**

