



Identifiants uniques

ETAT DE L'ART ET BONNES PRATIQUES

Types d'identifiants

Quels sont les types d'identifiants uniques (clés) que vous connaissez?

Les trois types possibles sont:

- Identifiants naturels
- Identifiants numériques auto-incrémentés
- Identifiants unique de type GUID ou UUID

Identifiants naturels

- Basés sur des **données réelles** : par exemple acronymes, numéro type AVS, ISBN, no de client, etc
- **Intuitives** : pourraient par exemple être le «nom / prénom»
- Doit être unique... et doit garantir de le rester dans le temps
 - Que faire si on change le numéro de numéro de client (agrandissement, fusion, etc)?
 - Toutes les clés doivent être modifiées dans toute la DB : source de bugs et de problèmes d'intégrité!
 - Si utilisation No AVS par exemple, comment faire si l'on doit stocker des personnes étrangères (qui n'en possèdent pas)?

Identifiants numériques auto-incrémentés

- **Clé artificielle** : ne correspond à aucune donnée réelle
- Indépendante et **dépourvue de signification** (sémantique)
- Identifiant unique pour la table dans laquelle il existe
- **Généré automatiquement** par le SGBD
- N'a aucune signification en dehors du traitement des données (ne sera jamais affiché pour l'utilisateur)

Identifiants de type GUID (UUID)

- **Clé artificielle** : ne correspond à aucune donnée réelle
- Signifie Globally Unique Identifier
- Taille de 16 octets alphanumériques. Exemple : e03ecf63-526f-11e7-818c-28f10e0c95a0
- Généré aléatoirement (générateur pseudo-aléatoire)
- **5×10^{36}** combinaisons possibles
- Dérivé de UUID de Unix
- GUID beaucoup utilisés dans le monde Microsoft (Bases de registres, clés dans les produits style MSCRM, SharePoint, etc)
- Génération de GUID
 - avec MySQL : SELECT **UUID()**
 - avec MSSQL : SELECT **NEWID()**

Avantages / inconvénients de chaque méthode

	Clé naturelle	Clé numérique auto-incrémentée	GUID
Avantages	Intuitives Debug simple	Indépendantes des données Prennent peu de place Toujours auto-générées par le SGBD Debug assez simple	Unicité (quasi) garantie dans plusieurs DB (fusion) Sécurité : autres ID impossibles à deviner
Inconvénients	Risque de doublon (surtout dans le temps) Risque de lenteurs (clés longues) Maintenance complexe en cas de modification de structure de la clé	Risque de compromettre des données : les id peuvent être devinés car séquence Problèmes potentiels en cas de fusions de DB ou ID exposées à l'extérieur	Debug plus compliqué Taille des données conséquente : lenteur Mauvaise indexation

Conclusion

- **Pas de solution unique**, le débat est toujours ouvert et animé: chaque solution a ses défenseurs et ses détracteurs
- **Essayer d'éviter les clés naturelles** : intuitives mais dangereuses (unicité, informations compromises, lenteurs, etc)

Liens et références

- [Livre GoogleBook](#)
- [Wikipedia](#)
- [Coin de web](#)
- [Débat GUID / Auto-incrémenté](#)
- [Mythes sur les GUID](#)