

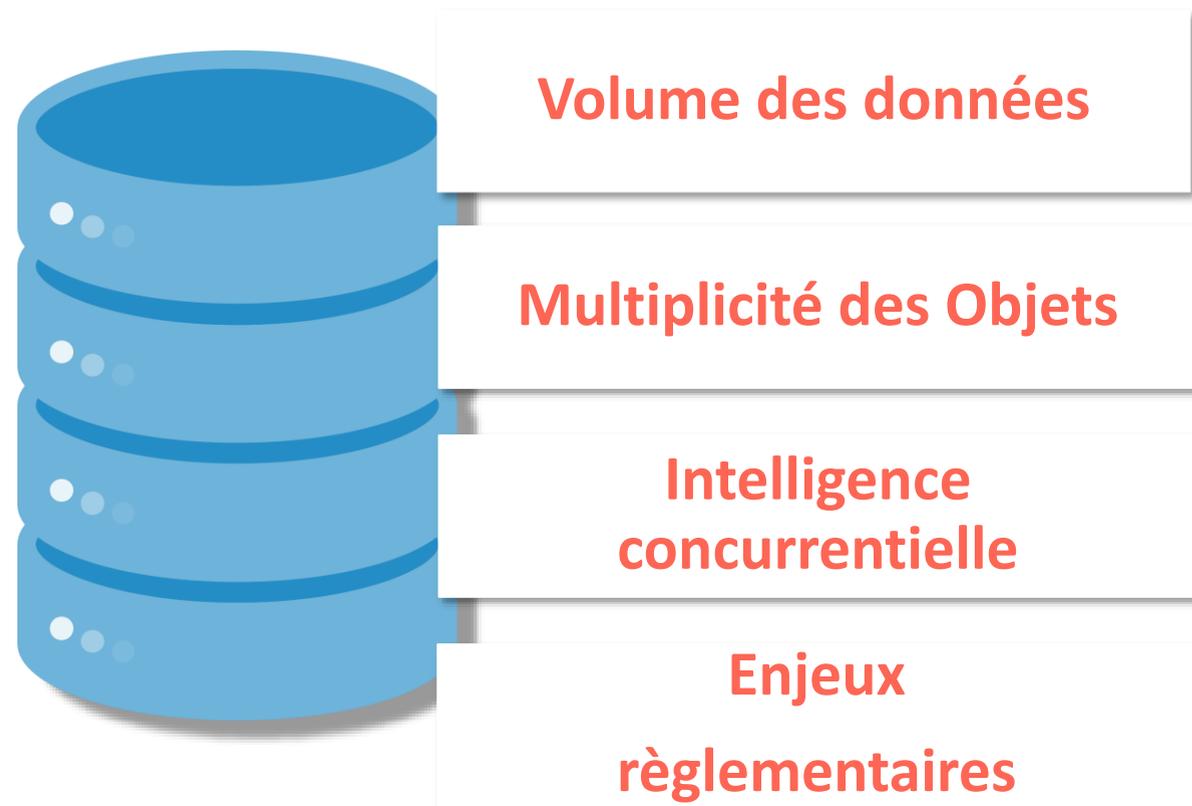


More than meets the *i*

Modernisation de la BD:

Comment s'ouvrir au futur tout en garantissant que l'existant continue de fonctionner?

Enjeux autour de la base de données



- Complexité des structures
- Modifications fréquentes
- BD Intelligente
- Intégrer la BD dans Dev OPS



SQL est le standard mondial



Nouveautés



DATA

Insight

understanding Temporal
Observation Discernment Perception
DB2 Judgment Wisdom
Olap

Enjeu Ressources

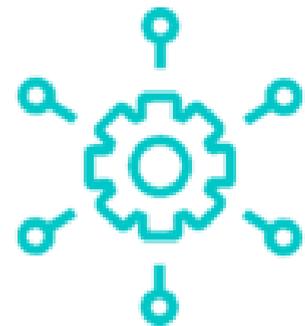


Un investissement limité

Moteur



sur



Déjà installé

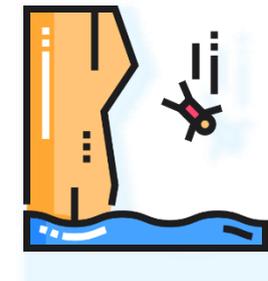
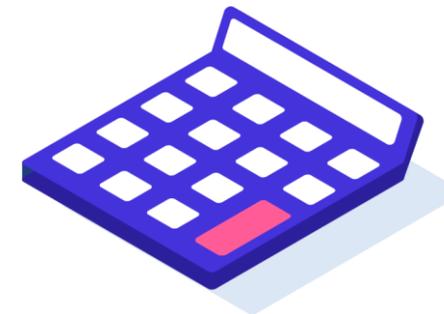


Performant



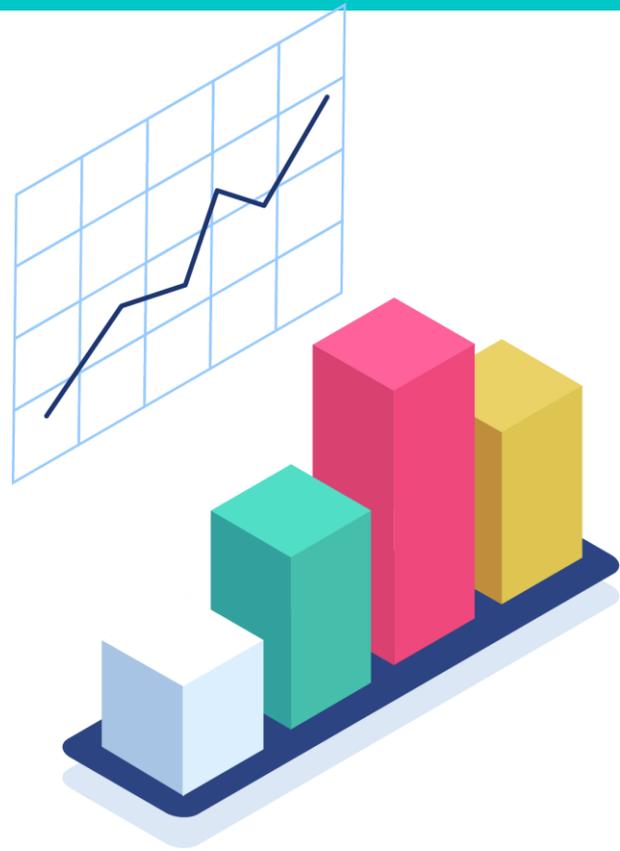
Fonctionnalités avancées

Dans + de 70% des cas
le remplacement est un échec



La modernisation
est moins couteuse &
moins risquée

Les freins



**Préserver le
business**



**Temps des
équipes**



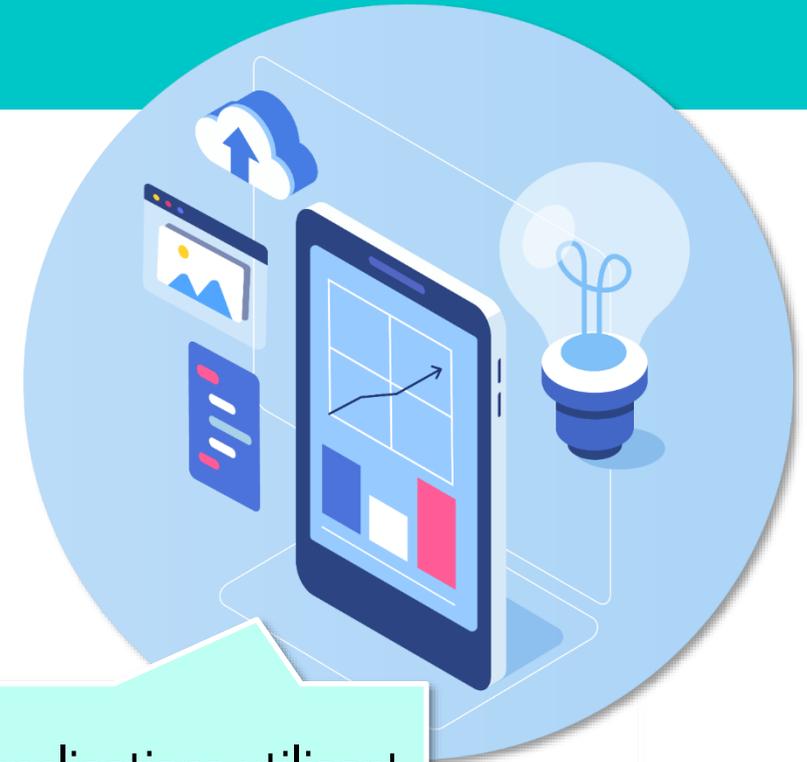
**Compétences
manquantes**

Objectif

**Faire coexister
l'existant**



**Avec l'ouverture
au futur**



Temps d'arrêt de la BDD minime
pendant la modernisation

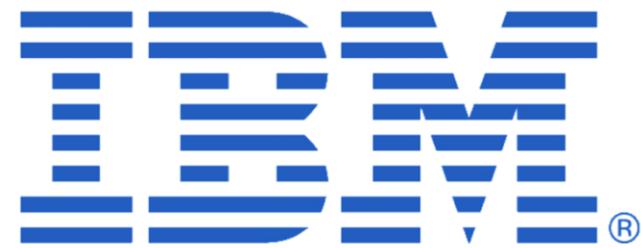
Les applications existantes
fonctionnent comme avant

Les nouvelles applications utilisent
immédiatement les nouvelles
fonctionnalités de la BDD

La transition s'effectue de manière progressive et fluide, à votre rythme

Comment

Méthode



Solution



Mise en oeuvre

VOUS



itheis

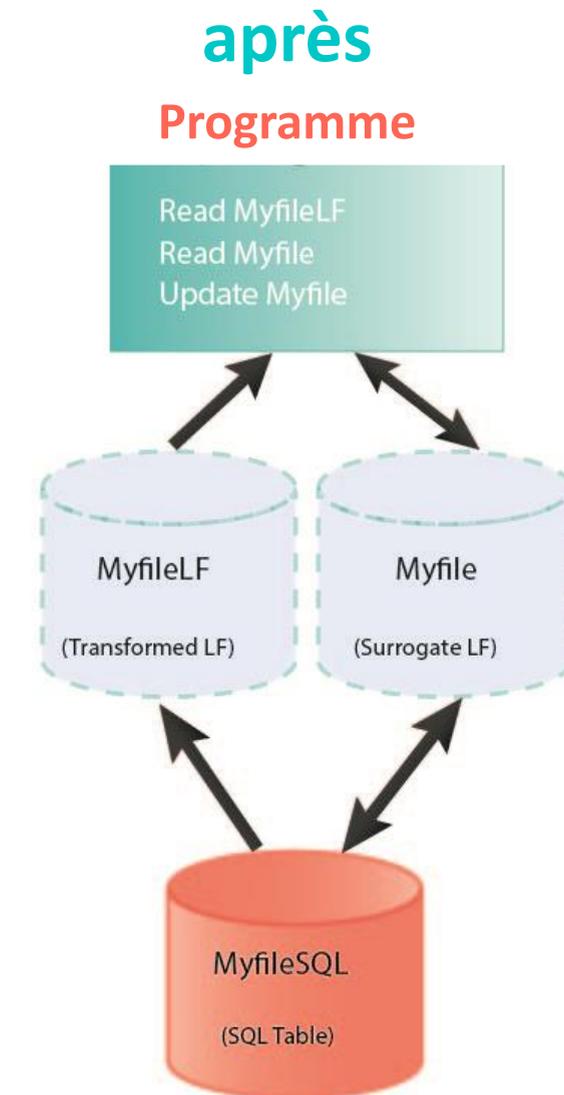
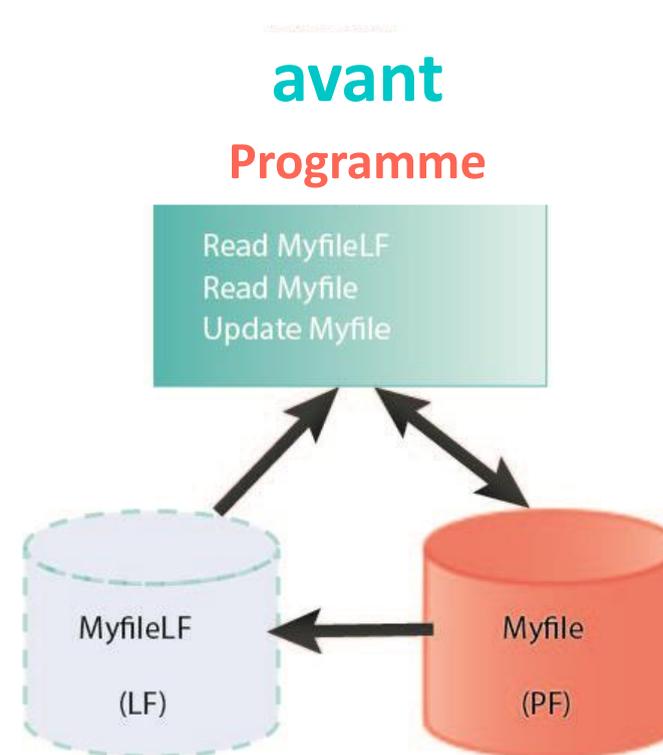


resolution
software

La méthode LF Surrogate d'IBM

Pour chaque fichier physique reconfiguré dans une table SQL, un fichier logique appelé substitut(/surrogate) LF est généré.

Les applications existantes s'appliqueront désormais au LF de substitution qui a remplacé le fichier physique et continuera à fonctionner comme avant, SAUF qu'elles écriront et liront des données dans la nouvelle table SQL.



Caractéristiques

Caractéristiques du LF Surrogate	Caractéristiques du LF Transformé	Caractéristiques du Tableau SQL
<ul style="list-style-type: none">• Même nom que le PF d'origine• Mêmes colonnes que le PF d'origine• Mêmes mots clés que le PF d'origine• Même chemin d'accès que le PF d'origine• Même FLI que le PF d'origine• Pointe vers la nouvelle table SQL	<ul style="list-style-type: none">•Même nom que le LF d'origine•Mêmes colonnes que le LF d'origine•Mêmes mots-clés que le LF d'origine•Même chemin d'accès que le LF d'origine•Même FLI que le LF d'origine•Le nouveau LF pointe vers la ou les nouvelles tables SQL	<ul style="list-style-type: none">•Nouveau nom système et nom long de la table•Noms systèmes des colonnes conservés•Nouveaux noms longs pour les colonnes•Colonnes existantes du fichier original PF préservées•Ajout de nouvelles colonnes (Identité, Audit, Dates, FKs)•Préserve les autorisations, les paramètres de compilation, la journalisation•Création d'index SQL (uniquement lorsqu'ils partagent le même chemin d'accès que la clé d'origine du LF)

La méthodologie Modernize-DB



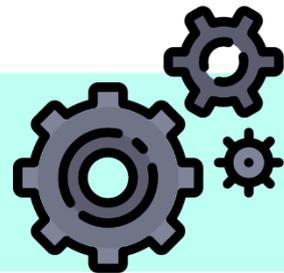
Nomenclature automatisée

via templates configurables
(Tables et Colonnes)



Diagnostic

(Surrogate nécessaire ? CL Impliquées, Code Source défaillant, multi membre, fichier de travail...)



Génération des objets

de la nouvelle base - qui remplace la précédente- détaillée et re-jouable



Vérification des Datas



Copie des datas



Script SQL pour mise en Prod

après test
(conforme Dev-Ops)

Colonnes Supplémentaires



Colonnes automatiquement maintenues par le système lui-même

- **Identité**

- **Colonnes d'audit**

- Temps d'insertion de l'enregistrement
- Temps de Modification de l'enregistrement
- Utilisateur ayant inséré l'enregistrement
- Utilisateur ayant modifié l'enregistrement
- Programme ayant effectué la mise à jour
- ...



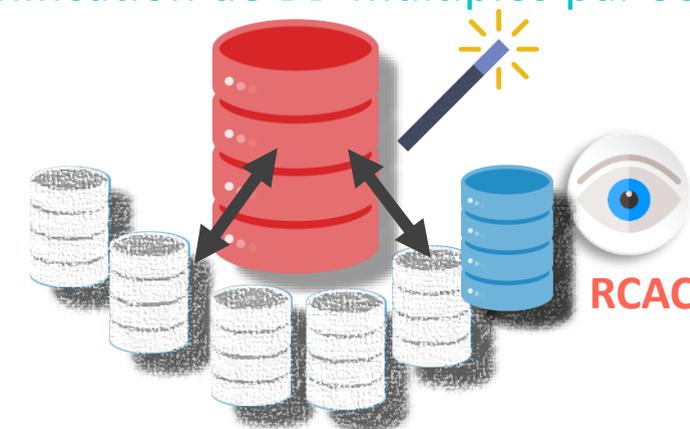
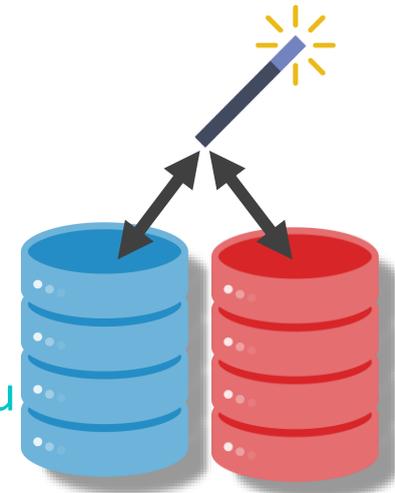
Colonnes pouvant être maintenues par des triggers

- **Trigger bidirectionnels**

- Création de Dates au format Date
- Création de FK basé sur l'identité du parent

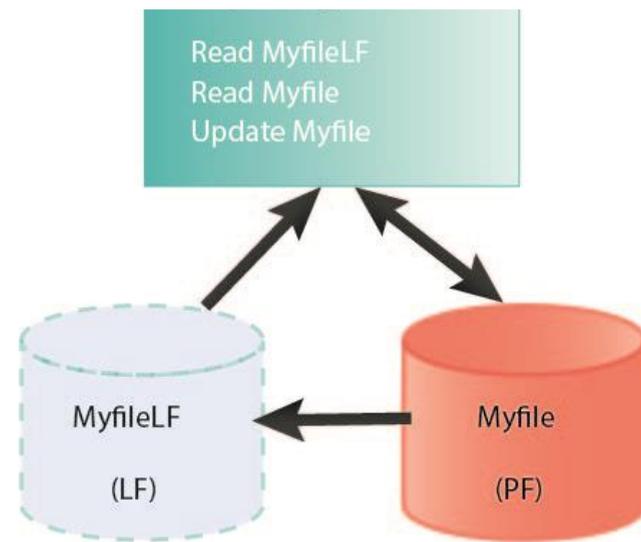
- **Trigger + RCAC**

- Unification de BD multiples par Société



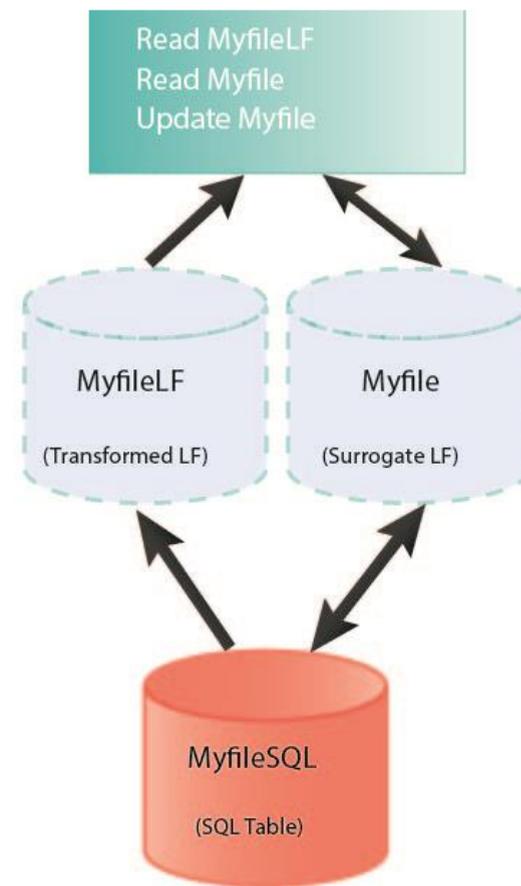
avant

Programme



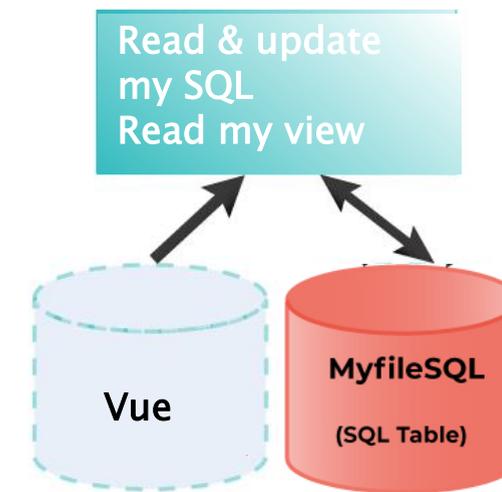
intermédiaire

Programme

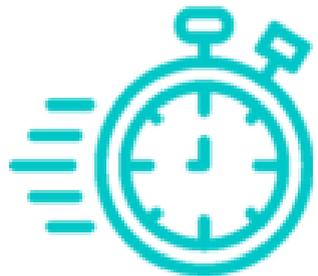


final

Programme



Les bénéfices Modernize-DB



**Modernisation +
rapide avec - de
risques et d'efforts**



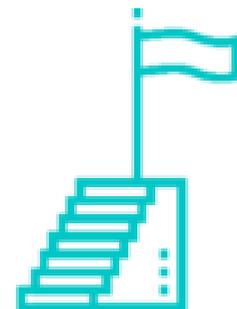
**Protection des
applications
existantes**



**Processus
exhaustif**



**Amélioration de la
qualité des données**



**Permet une
modernisation
progressive**

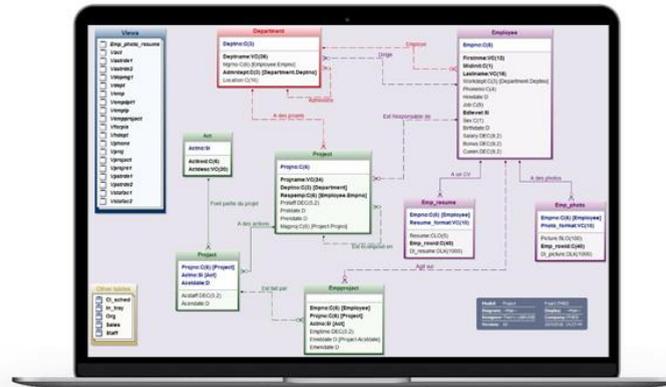


**Pas d'expertise
SQL requise**

Xcase for i : une suite complète pour DB2

Modernize-DB

Passage à SQL



Relate-DB

Découverte
Implémentation des
Relations

Viewer-DB

Vision partagée &
personnalisée de la BDD

Evolve -DB

Gouvernance
Coordination via modèle
graphique

Test-DB

Génération de bases de
données de Test

Anonymize DB

Anonymisation
des bases de test

La suite ?

Demo

POC

Merci

