

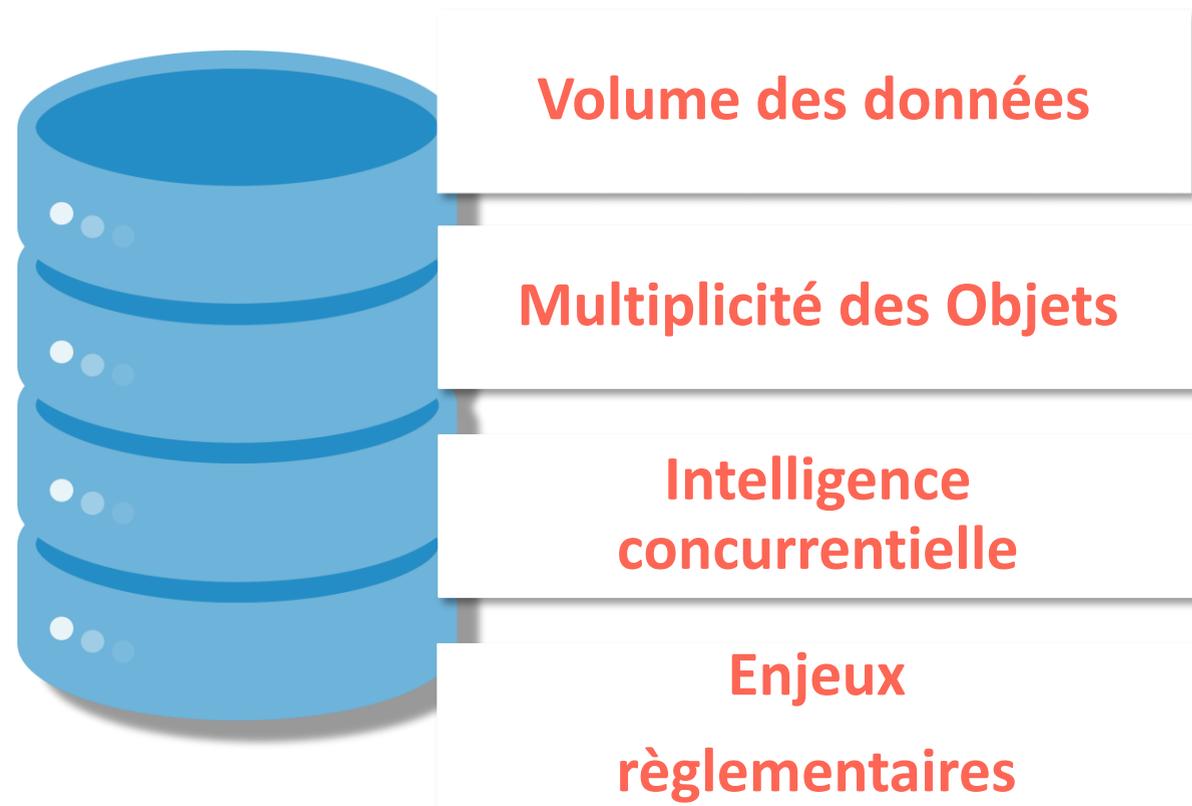


**More than meets the *i***

# 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



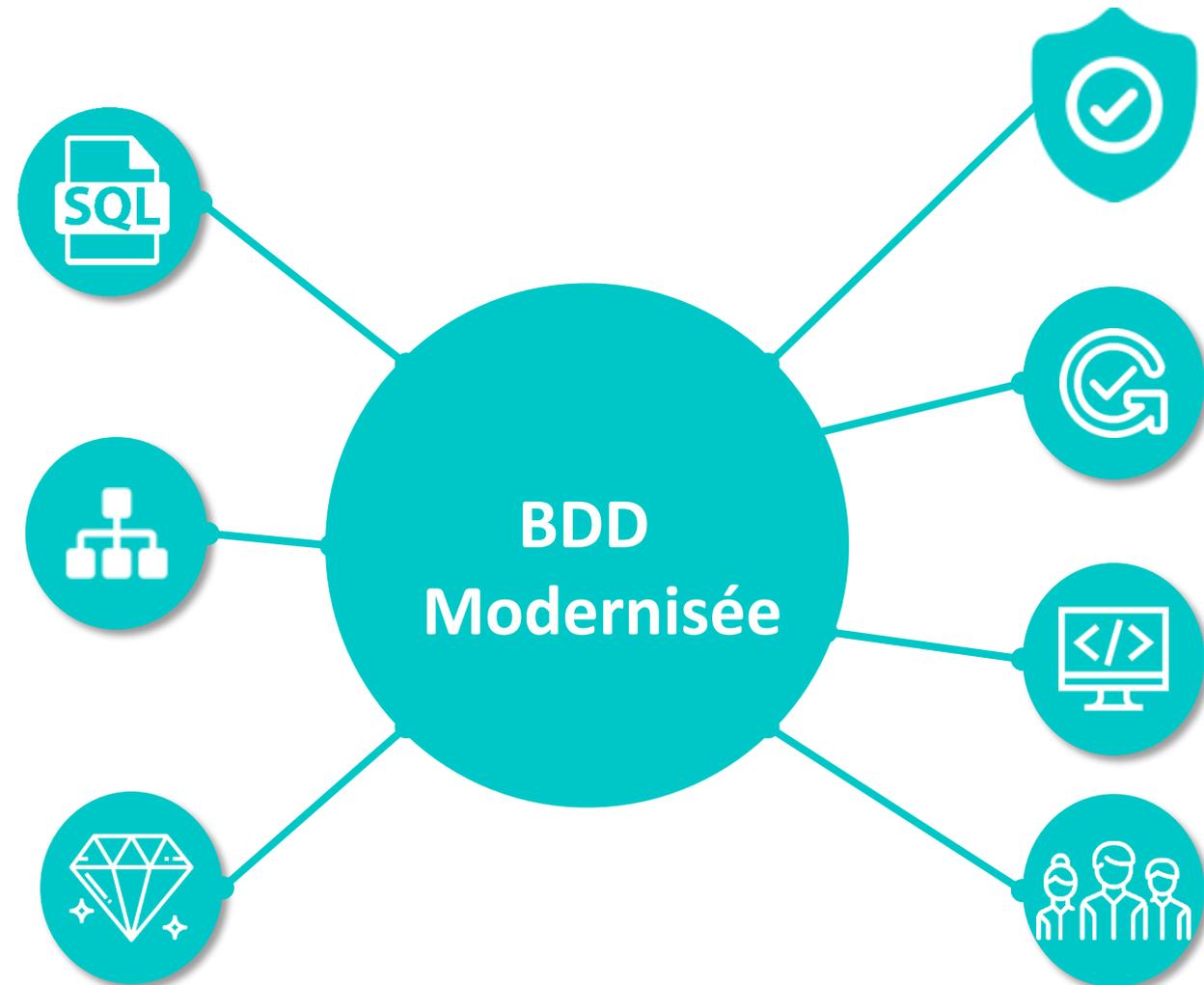
- **Complexité des structures**
- **Modifications fréquentes**
- **BD Intelligente**
- **Intégrer la BDD dans Dev OPS**

# Bénéfices de la modernisation de la BDD

Ouverture au nouveaux enjeux  
(technologies et personnes)  
sans compromettre l'existant

Documentation et  
compréhension de la base par  
l'ensemble de l'équipe

Assainissement et protection  
des données



Gestion réglementaire et agile  
des environnements de  
développement

Gouvernance de la base et du  
cycle DevOps

Génération assistée de code  
lié à la base de données

Travail collaboratif et  
coordonné autour de la base  
de données

# SQL est le standard mondial



## Nouveautés



=

DATA

Insight

understanding Temporal  
Observation Discernment Perception  
DB2 Judgment Wisdom  
Olap

## Enjeu Ressources

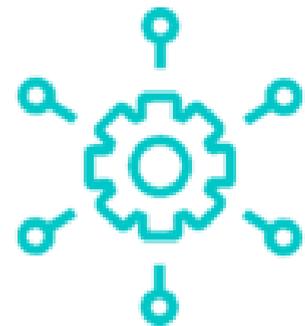


# Un investissement limité

Moteur



sur



Déjà installé

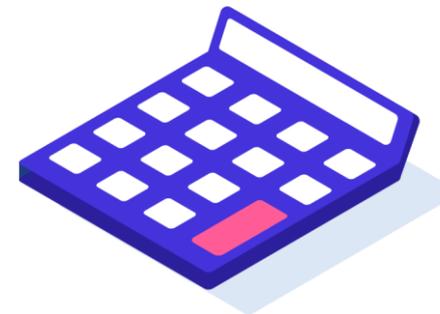


Performant



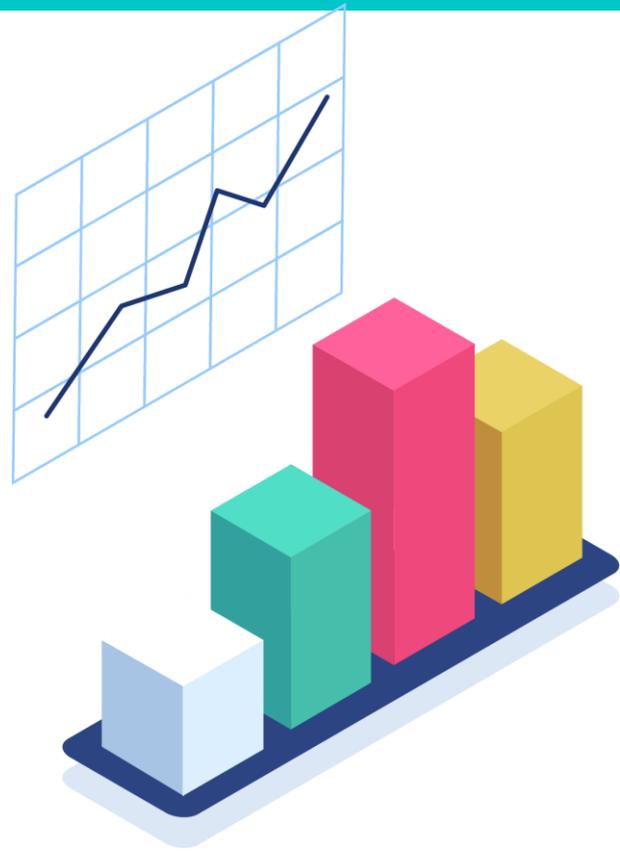
Fonctionnalités avancées

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



La modernisation  
est moins couteuse &  
moins risquée

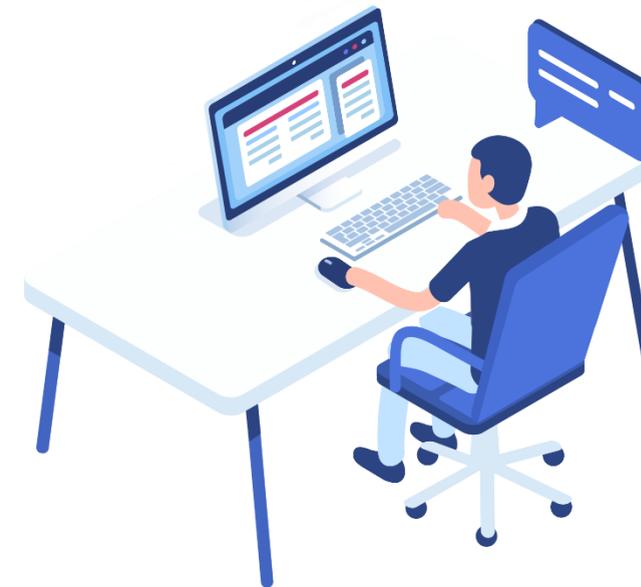
# Les freins



**Préserver le  
business**



**Temps des  
équipes**



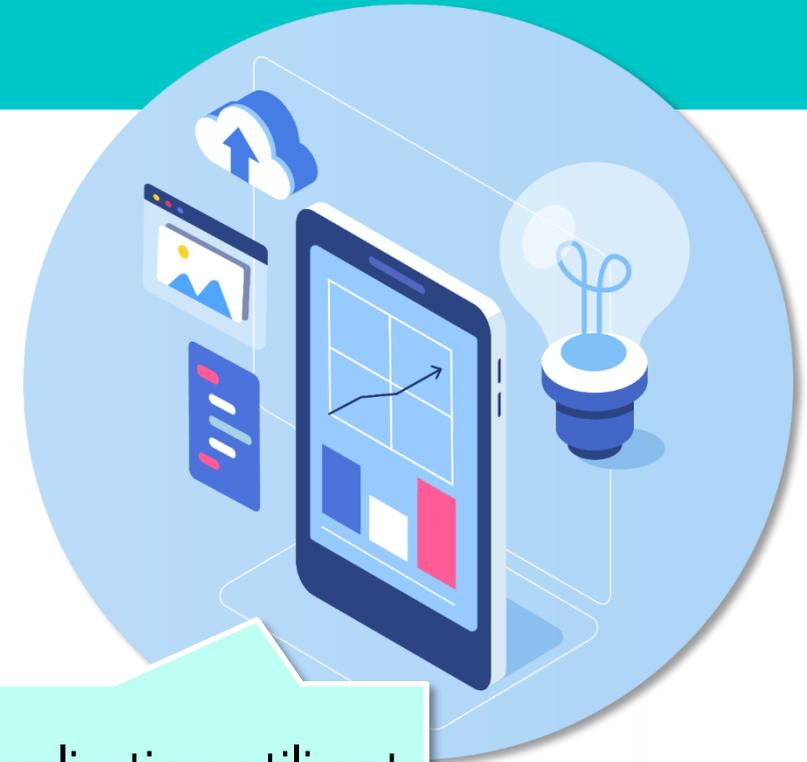
**Compétences  
manquantes**

# Objectif

**Faire coexister  
l'existant**



**Avec l'ouverture  
au futur**



Temps d'arrêt de la BDD minime  
pendant la modernisation

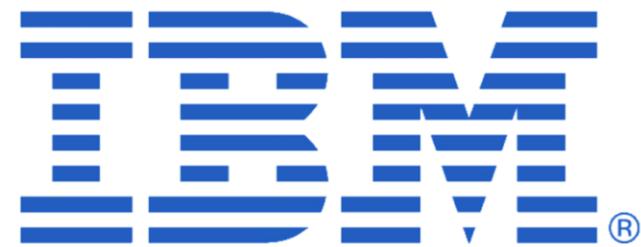
Les applications existantes  
fonctionnent comme avant

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

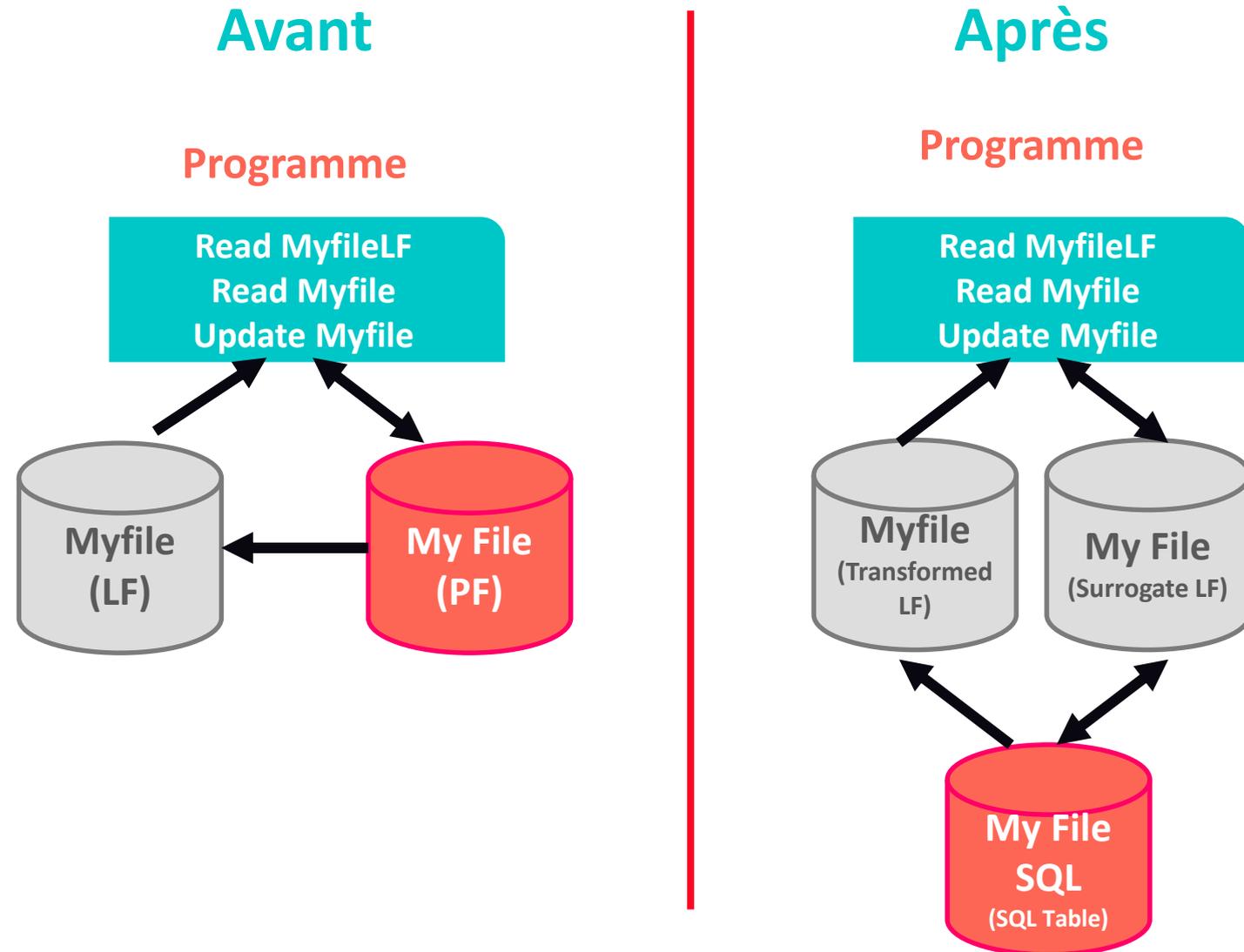


resolution  
software

# La méthode LF Surrogate d'IBM

Pour chaque fichier physique reconfiguré dans une table SQL, un fichier logique appelé substitut 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.



# La méthodologie Migrate-DB



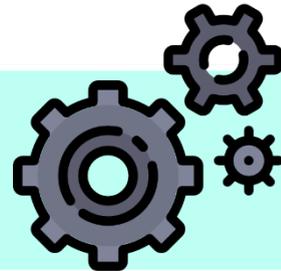
## Diagnostic

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



## Nomenclature automatisée

via templates  
configurables  
(Tables et Colonnes)



## Génération des objets

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



## Vérification des Datas



## Copie des datas



## Scripts SQL pour mise en Prod

après test  
(conforme Dev-Ops)

# Colonnes Supplémentaires



## Colonnes automatiquement maintenues par le système lui-même

- **Identité**

- **Colonnes d'audit**

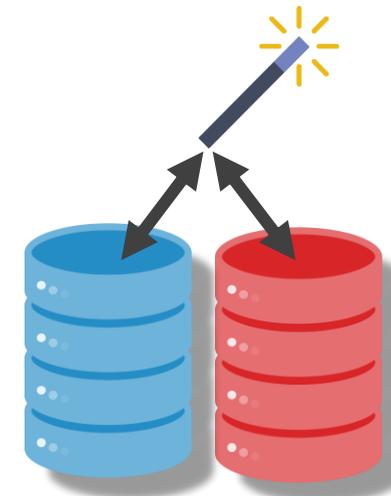
- Temps d'insertion de l'enregistrement
- Temps de Modification de l'enregistrement
- Utilisateur ayant inséré l'enregistrement
- Utilisateur ayant modifié l'enregistrement
- Programme ayant effectué la mise à jour
- ...



## Colonnes pouvant être maintenues par des triggers

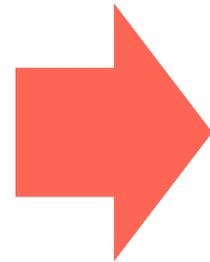
- **Trigger bidirectionnels**

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

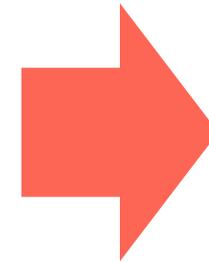


# La suite ?

Démo



POC



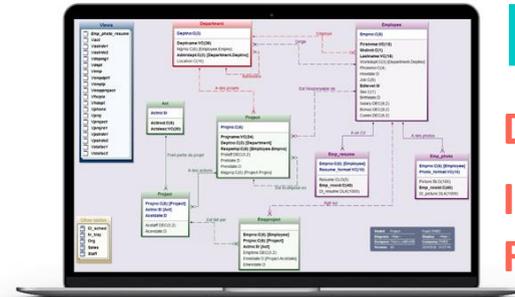
Implémentation

# Xcase for i : une suite complète pour DB2



## Migrate-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

# Merci

