



## Situation actuelle

- Rappel Historique: DB2 est né en 1983.
- DB2 Version 5
  - ▶ plus supportée depuis le 31 décembre 2002
- DB2 Version 6
  - ▶ plus commercialisée depuis le 30 juin 2002
  - ▶ date de fin de support = 30 juin 2005
- DB2 Version 7
  - ▶ la version la plus utilisée
- DB2 Version 8
  - ▶ la dernière version disponible



## Annonce de DB2 Version 8

- Annoncée le 28 Janvier 2003
- Disponible (GA) depuis le 26 mars 2004
- Annonce simultanée de "DB2 Utilities Suite" Version 8
  
- Prérequis:
  - ▶ Machine supportant l'architecture Z (type z800, z890, z900, z990)
  - ▶ z/OS V1R3 minimum
  - ▶ si utilisé: IMS V7, CICS TS 1.3, WebSphere MQSeries pour OS/390 V5.2
  
- Migration
  - ▶ à partir de DB2 Version 7 seulement

DB2 UDB  
for z/OS  
Version 8



Breaking  
the  
Limits

## DB2 Version 8 : extension des limites

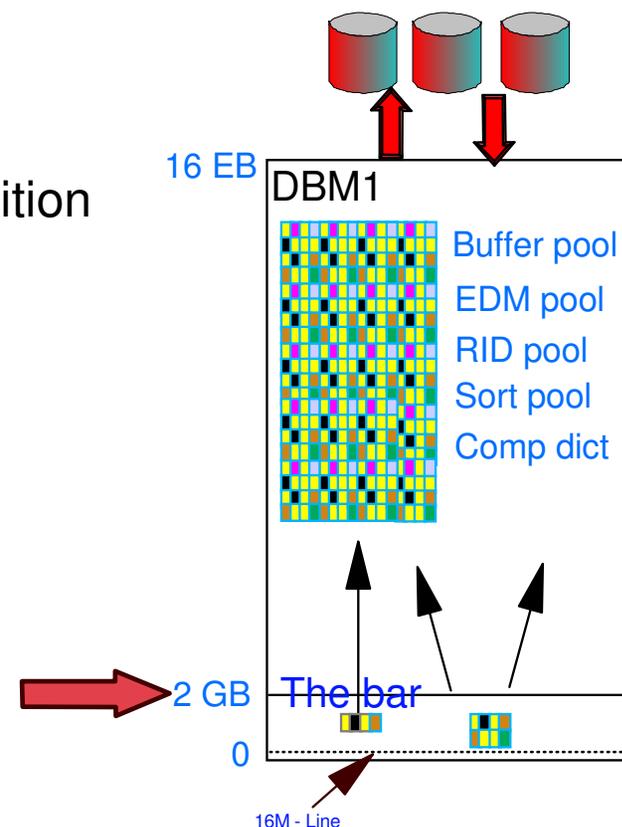
- Support des longs noms
  - ▶ Nom des tables, vues et alias jusqu'à 128 caractères (18 en V7)
  - ▶ Nom des colonnes jusqu'à 30 caractères (18 en V7)
  - ▶ Longueur de clé d'index jusqu'à 2000 caractères (255 en V7)
  - ▶ USER (SCHEMA) jusqu'à 128 caractères (8 en V7)
  - ▶ Current SQLID jusqu'à 128 caractères (8 en V7)
  - ▶ Current PACKAGESET jusqu'à 128 caractères (8 en V7)
  - ▶ Current PATH jusqu'à 2048 caractères (254 en V7)

→ compatibilité , portabilité vers les autres SGBDs
  
- Augmentation du nombre de Logs
  - ▶ Logs Actives: de 31 à 93
  - ▶ Logs Archives: de 1000 à 10000



## DB2 Version 8 : extension des limites

- Augmentation pour les Tablespaces Partitionnés:
  - ▶ Taille Maximum d'un Tablespace (pages de 32K): de 16 TB à 128 TB
  - ▶ Nombre de partitions: de 254 à 4096
    - dernier composant du nom du cluster identifie la partition
      - de 1 à 999: Axxx
      - de 1000 à 1999: Bxxx
      - de 2000 à 2999: Cxxx
      - de 3000 à 3999: Dxxx
      - de 4000 à 4096: Exxx
  
- Support du 64 bits
  - ▶ Mémoire virtuelle: de  $2^{31}$  à  $2^{64}$  (2 GB à 16 ExaBytes)
  - ▶ Au dessus des 2GB:
    - Bufferpools, EDMPool,
    - SortPool, RIDPool, verrous IRLM
    - dictionnaires de compression



## DB2 Version 8: améliorations SQL

- INSERT multi-lignes
  - ▶ INSERT INTO TAB1 FOR :hv ROWS  
VALUES (:ARRAY1, :ARRAY2)
  - ▶ Insertion de plusieurs lignes dans une table ou une vue en un seul ordre SQL
  
- FETCH multi-lignes
  - ▶ DECLARE C1 CURSOR  
WITH ROWSET POSITIONNING  
FOR SELECT COL1,COL2 FROM TAB1;  
OPEN C1;  
FETCH FROM C1  
FOR :hv ROWS INTO :ARRAY1, ARRAY2;
  - ▶ Concept de "rowset": ensemble de lignes retrouvé à travers un FETCH multi-lignes

→ portabilité et performance (surtout si réseau)



## DB2 Version 8: améliorations SQL

- GET DIAGNOSTICS
  - ▶ retourne les erreurs SQL
    - pour tout un ordre
    - pour chaque condition (si plusieurs erreurs ont lieu)
  - ▶ support des messages d'erreur SQL supérieur à 70 octets (limitation de la SQLCA)
  - ▶ nécessaire pour supporter les longs noms et les opérations multi-lignes
  
- ▶ 

```
INSERT INTO TAB1 FOR 5 ROWS VALUES (:ARRAY);
GET DIAGNOSTICS :ERR_COUNT = NUMBER;
DO II = 1 TO ERR_COUNT;
    GET DIAGNOSTICS FOR CONDITION :II
    :RC = RETURNED_SQLSTATE;
END;
```

## DB2 Version 8: améliorations SQL

- INSERT imbriqué dans un SELECT
  - ▶ Permet de retrouver des valeurs insérées par DB2 lors d'un INSERT
    - Colonnes IDENTITY, valeurs SEQUENCES
    - Valeurs par défaut, ROWID
    - Colonnes modifiées par un trigger
  - ▶ 

```
SELECT C1, C2, C3, C4, C5 FROM
INSERT (C1, C5) INTO TAB1
VALUES ('ABC', CURRENT DATE)
```
  
- FULLSELECT Scalaire
  - ▶ Autorise des 'Fullselect scalaires' là où sont autorisées des expressions
  - ▶ Un 'Fullselect scalaire' rend une seule ligne constituée d'une seule valeur
  - ▶ 

```
SELECT PRODUCT , PRICE FROM PRODUCTS
WHERE PRICE > 2 * (SELECT MIN(PRICE) FROM PRODUCTS)
```

## DB2 Version 8: améliorations SQL

### ■ SEQUENCES

- ▶ Permet d'avoir compteur qui s'incrémente via DB2
- ▶ Même fonctionnalité que les colonnes IDENTITY mais sans faire partie de la table (c'est un objet)
- ▶ CREATE SEQUENCE
- ▶ ALTER SEQUENCE
  - RESTART WITH valeur
  - INCREMENT BY
  - MINVALUE
  - MAXVALUE
  - CYCLE/NO CYCLE
  - CACHE/NO CACHE
- ▶ NEXTVAL: valeur suivante
- ▶ PREVVAL: valeur précédente

→ compatibilité , portabilité vers les autres SGBDs

## DB2 Version 8: améliorations SQL

- Curseurs Scrollables Dynamiques
  - ▶ Scrolling directement sur la table de base
  - ▶ Permet de voir les mises à jour faites par d'autres applications
  - ▶ DECLARE C1 SENSITIVE DYNAMIC CURSOR  
FOR SELECT COL1,COL2 FROM TAB1
  
- GROUP BY expression
  - ▶ SELECT COL1+COL2,COL3  
FROM TAB  
GROUP BY COL1+COL2
  
- Plusieurs clauses DISTINCT
  - ▶ SELECT COUNT(DISTINCT(C1)), AVG(DISTINCT(C2))  
FROM TAB
  
- Jointure entre Table UNICODE et Tables EBCDIC ou ASCII
  - ▶ plusieurs CCSIDs par ordre SQL



## DB2 Version 8: améliorations SQL

- Expression de Table Commune (Common Table Expression ou CTE)
  - ▶ L'expression de Table Commune est définie et utilisée dans un ordre SQL
  - ▶ Chaque Expression de Table Commune peut être référencée plusieurs fois par ordre SQL. Toutes les références partagent la même table résultante.
  - ▶ 

```
WITH RPL (PART,SUBPART,QUANTITY) AS  
  (SELECT R.PART, R.SUBPART, R.QUANTITY  
   FROM PARTLIST AS R WHERE R.PART='01')  
SELECT DISTINCT PART,SUBPART,QUANTITY  
FROM RPL  
ORDER BY PART,SUBPART,QUANTITY
```

→ compatibilité , portabilité vers les autres SGBDs



# DB2 Version 8: améliorations SQL

- SQL Récuratif

WITH

```
RPL (PART, SUBPART, QUANTITY) AS
```

```
(
```

## Select Initialisation

```
SELECT ROOT.PART, ROOT.SUBPART, ROOT.QUANTITY  
FROM PARTLIST ROOT  
WHERE ROOT.PART = '01'
```

**UNION ALL**

## Select Itératif

```
SELECT CHILD.PART, CHILD.SUBPART, CHILD.QUANTITY  
FROM RPL PARENT, PARTLIST CHILD  
WHERE PARENT.SUBPART = CHILD.PART
```

```
)
```

## Select Principal

```
SELECT PART, SUBPART, SUM(QUANTITY) AS QUANTITY  
FROM RPL  
GROUP BY PART, SUBPART
```

## DB2 Version 8: améliorations SQL

- Tables de requêtes matérialisées (Materialized Query Tables ou MQT)
  - ▶ dites aussi Tables Summary
  - ▶ Deux types de MQT
    - maintenue-par-le-système via ordre SQL REFRESH
    - maintenue-par-l'utilisateur via triggers, mises à jours batch, ..
  - ▶ Le DBA peut utiliser une MQT pour l'optimisation comme il utiliserait un index
  - ▶ L'optimiseur peut réécrire une requête soumise sur la table de base afin d'utiliser la MQT
  - ▶ La MQT peut être nommée directement dans l'ordre SQL
  - ▶ Information pré-calculée

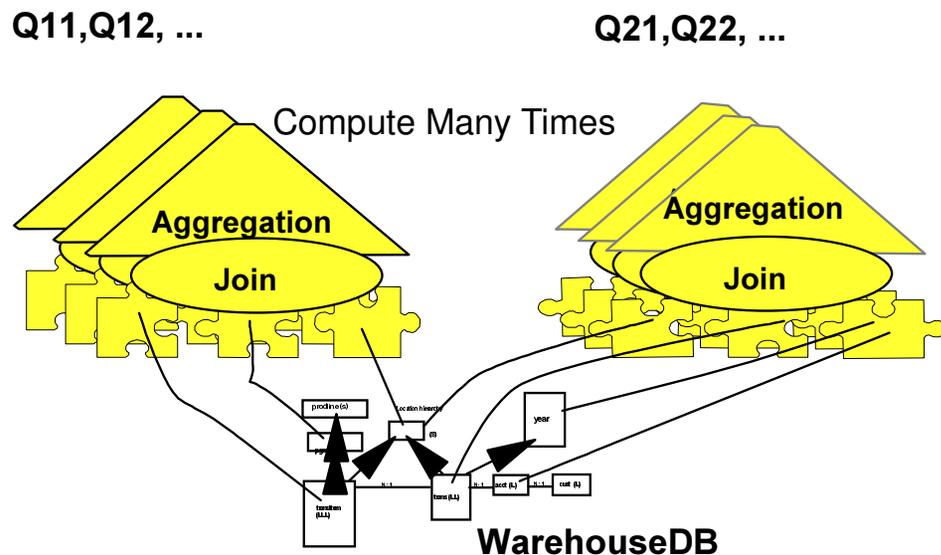
→ Meilleures performances



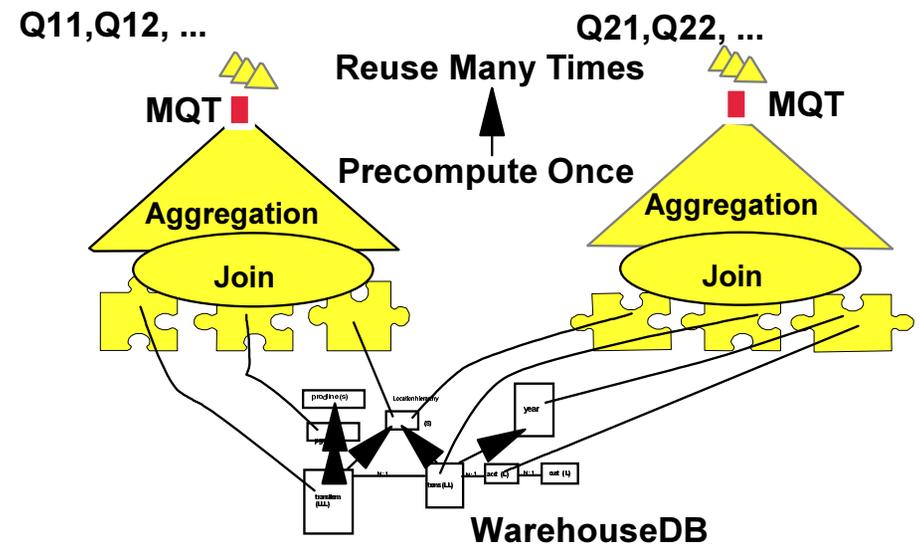
# DB2 Version 8: améliorations SQL

- Tables de requêtes matérialisées (MQT): utilisation

Sans les MQT



Avec les MQT



→ Meilleures performances pour requête décisionnelle



## DB2 Version 8: améliorations SQL

- Tables VOLATILE
  - ▶ Spécifié au moment de la création de la table
    - CREATE TABLE XYZ .... VOLATILE
  - ▶ Permet d'encourager l'utilisation d'un index pour une table ayant des données variables (nombre de lignes, forte plage de cardinalité)
    - Meilleures performances surtout pour SAP
  
- Performances SQL
  - ▶ Prédicat indexable pour types de données différents
    - exemples:
      - colonne en CHAR(3), littéral ou host variable en CHAR(4)
      - colonne en DECIMAL, host variable en FLOAT
    - intéressant pour les langages ne supportant pas tous les types de données SQL
      - exemple: C/C++ n'ont pas de type DECIMAL
  - ▶ Scan d'Index en arrière (backward)

## DB2 Version 8: évolution online du "schema"

- ALTER des caractéristiques des tables
  - ▶ longueur d'une colonne: CHAR(10) en CHAR(15)
  - ▶ type d'une colonne: CHAR en VARCHAR
  - ▶ modification entre types numériques (SMALLINT, INTEGER, DECIMAL, FLOAT, ...)
  - ▶ caractéristiques des colonnes IDENTITY
  
- Quelques impacts d'un ALTER sur une colonne
  - ▶ Tablespace positionné en état AREO (Advisory REORG pending)
    - données toujours accessibles mais risque de dégradation de performance
    - données inchangées sauf lors d'un INSERT ou UPDATE
  - ▶ Plans et Packages référençant la colonne invalidés
  - ▶ Valeurs de Runstats sur colonnes invalidées

→ Modification des objets plus simple car dynamique



## DB2 Version 8: évolution online du "schema"

- Modifications des caractéristiques des index
  - ▶ Ajout d'une colonne dans un index
    - ajout à la fin de l'index
    - impacts: index positionné en état RBDP
    - restriction: interdit pour les Index Partitionnés, pour les index primaires
  - ▶ Suppression (DROP) d'un index partitionné
  - ▶ Modification de l'attribut CLUSTER
    - ALTER INDEX index1 NOT CLUSTER
    - ALTER INDEX index2 CLUSTER
  - ▶ Colonne en VARCHAR plus "paddées" à la longueur maximale
    - Pour les anciens index: ALTER INDEX PADDED/NOT PADDED
      - index positionné en état RBDP
    - Pour les nouveaux index: dépend d'un paramètre du ZPARM
      - PADIX = NO (défaut)

→ Disponibilité

## DB2 Version 8: évolution online du "schema"

- Modification du nombre de partitions
  - ▶ ajout d'une partition en fin de table
    - arrêt du tablespace et de l'index de partitionnement nécessaire
  - ▶ Via ALTER TABLE ... ADD PARTITION ENDING AT (zzzz)
  
- Rotation des partitions
  - ▶ Via ALTER TABLE ... ROTATE PARTITION ...
    - indication de la clé maximale de la nouvelle partition
  - ▶ Les données de la première partition sont supprimées via DELETE
  - ▶ La nouvelle partition devient la partition 1
    - ordre logique ne correspond plus à l'ordre physique
  - ▶ Disponibilité immédiate (pas de REORG nécessaire)

Part 2 fev 1998

Part 59 nov 2002

Part 60 dec 2002

Part 1 janv 2003

→ Modification des objets plus simple car dynamique



## DB2 Version 8: Tablespaces Partitionnés

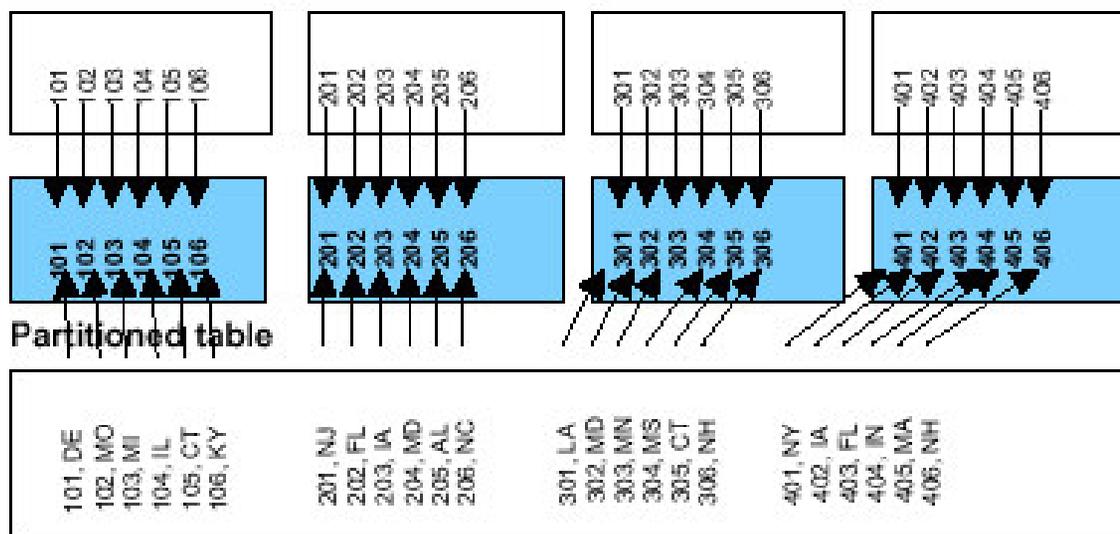
- Plus d'index obligatoire pour le partitionnement!
- Définition des clés de partitionnement dans la table
  - ▶ CREATE TABLE CUSTOMER (  
    ACCOUNT\_NUM          INTEGER,  
    CUST\_NAME            CHAR(30),  
    ...)  
    PARTITION BY (ACCOUNT\_NUM)  
    (PART 1 ENDING (199),  
    PART 2 ENDING (299),  
    ...  
    PART n ENDING (xxx))
  - ▶ La définition des objets est complète



# DB2 Version 8: Tablespaces Partitionnés

- Plusieurs types d'index:
  - ▶ Index Partitionné (Partitioned Index ou PI)
    - index physiquement partitionné
  - ▶ Index de Partitionnement (Partitioning Index)
    - index dont la(les) colonne(s) les plus à gauche correspond(ent) à la (les) colonne(s) de partitionnement de la

Partitioned Partitioning index part\_ix\_1



◀ Index de Partitionnement Partitionné

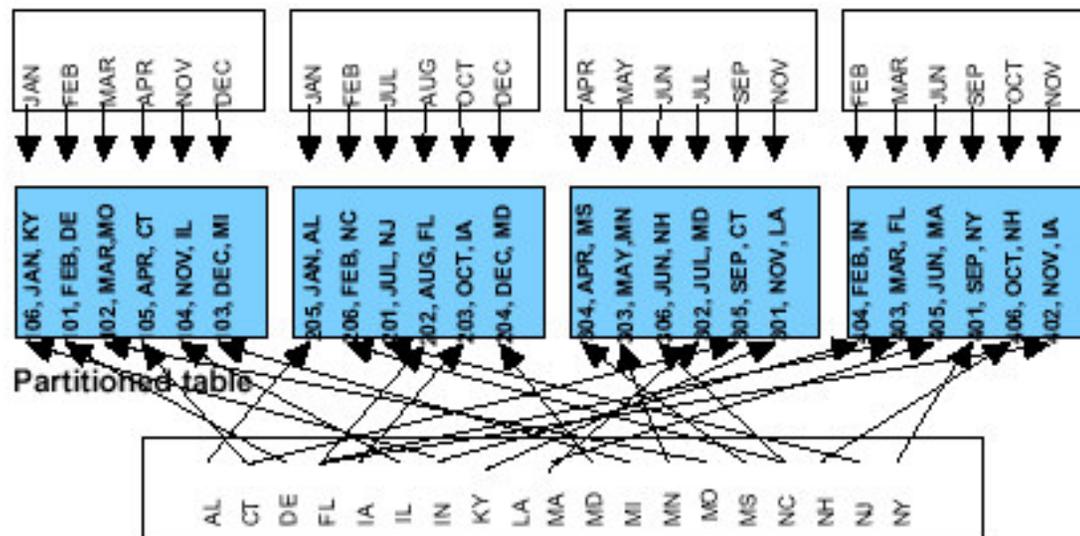
◀ Index de Partitionnement Non-Partitionné

Non-partitioned Partitioning index part\_ix\_2

# DB2 Version 8: Tablespaces Partitionnés

- D'autres types d'index
  - ▶ Index Secondaires: tout index qui n'est pas un index de partitionnement
    - DPSI = Data Partitioned Secondary Index
      - index secondaire physiquement partitionné
    - NPSI = Non Partitioned Secondary Index
      - index secondaire non-partitionné

Data Partitioned Secondary Index (DPSI) -- data\_part\_si\_1



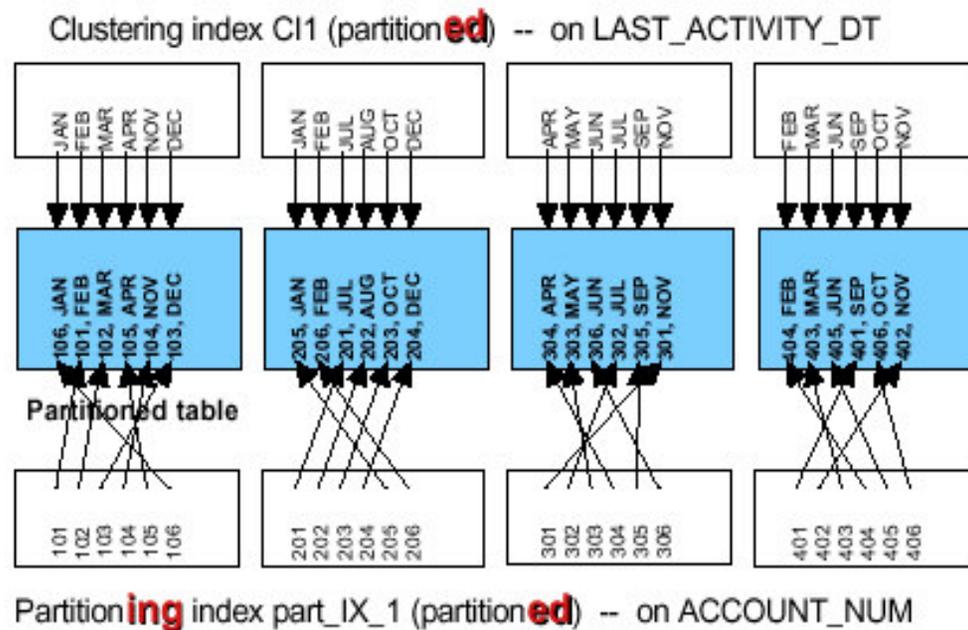
▶ Index Secondaire Partitionné

▶ Index Secondaire Non-Partitionné

Non-Partitioned Secondary Index (NPSI) -- non\_part\_si\_2

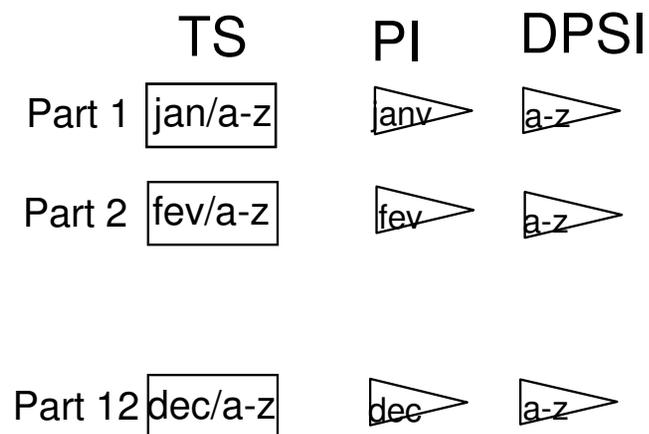
## DB2 Version 8: Tablespaces Partitionnés

- Index Cluster
  - ▶ Cela peut être n'importe quel index
    - L'index Cluster n'est plus forcément l'Index de Partitionnement
  - ▶ Peut être modifié via ALTER INDEX
  
- Ordre des lignes (pour INSERT et REORG)
  - ▶ La colonne de partitionnement détermine la partition dans laquelle sera la ligne
  - ▶ L'index cluster détermine l'emplacement à l'intérieur de la partition



## DB2 Version 8: Tablespaces Partitionnés

- DPSI = index secondaire physiquement partitionné
  - ▶ Nombre de partitions du DPSI = nombre de partitions de la table
  - ▶ Les clés de la partition 'n' du DPSI réfèrent seulement aux lignes de la partition 'n' de la table
  - ▶ Exemple:
    - Partition des données par mois
    - Clustering par nom à l'intérieur de chaque partition de l'index DPSI



- ▶ CREATE INDEX ...

```
...
PARTITIONED
...
```



## DB2 Version 8 : améliorations opérationnelles

- Encore plus de paramètres du ZPARM modifiables dynamiquement
  - ▶ SYSADM, SYSOPR, CACHEDYN, ...
- Redémarrage automatique des utilitaires
- Améliorations du REORG Online
  - ▶ Possibilité de DISCARD
  - ▶ Plus de phase BUILD2 avec les nouveaux index (DPSI)
- LOAD/UNLOAD:
  - ▶ input et output délimité
- RUNSTATS:
  - ▶ statistiques de distribution non uniforme sur les colonnes non-indexées



## DB2 Version 8 : améliorations opérationnelles

- BACKUP et RESTAURATION au niveau de l'ensemble du système
  - ▶ Prérequis:
    - z/OS V1R5 (service DFSMSHsm)
      - Nouvel objet HSM: COPYPOOL
      - Nouvel objet HSM: COPY TARGET Storagegroup
    - Unités de contrôles supportant les APIs Flashcopy ESS
    - Datasets DB2 sur des volumes gérés par SMS
  - ▶ COPYPOOL
    - Nouvelle construction SMS - Ensemble de Storage Group SMS (256 maximum)
    - Attribut Version (85 Maximum)
      - pour spécifier le nombre de copies à maintenir sur disque
    - Chaque sous-système DB2 a 2 Copypools
      - DATA COPYPOOL (DSN\$location\_name\$DB)
      - LOG COPYPOOL (DSN\$location\_name\$LG)



## DB2 Version 8 : améliorations opérationnelles

- BACKUP au niveau de l'ensemble du système
  - ▶ Utilitaire BACKUP SYSTEM: sauvegarde volume rapide
    - Pas de point de cohérence (QUIESCE) nécessaire
    - Pas d'interruption mais DB2 suspend les opérations de création, suppression, renommage et les extensions
    - Gestion de 85 versions de backup sur disques (limite dépendante des règles SMS)
    - Information de backup contenues dans le BSDS et DFSMSHsm
    - Enregistre le « Recover Based Log Point (RBLP) dans DBD01
  
  - ▶ FULL BACKUP ou DATA ONLY
    - BACKUP SYSTEM - FULL
      - Sauvegarde des Databases et des Logs (Actives et BSDS)
      - Définition de COPYPOOL « Database » et « Log »
    - BACKUP SYSTEM - DATA ONLY
      - Sauvegarde des Databases
      - Définition de COPYPOOL « Database »



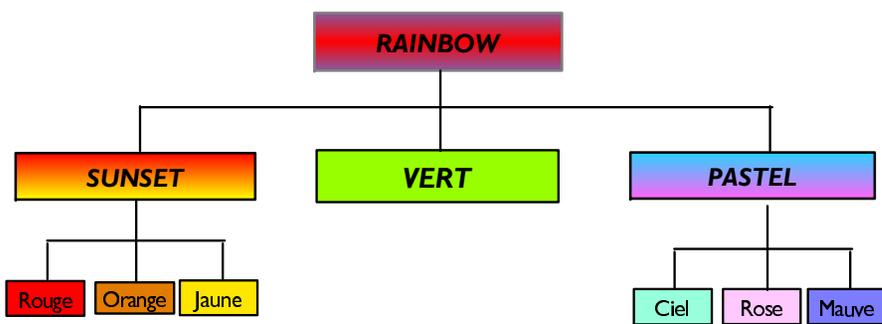
## DB2 Version 8 : améliorations opérationnelles

- RESTAURATION au niveau de l'ensemble du système
  - ▶ Restauration au PIT du BACKUP
    - Pas d'utilisation de l'utilitaire RESTORE SYSTEM
    - Utilisation des copies du BACKUP SYSTEM FULL pour restaurer les Copypool « Database » et « Log »
  - ▶ Utilitaire RESTORE SYSTEM
    - A utiliser pour restaurer à un point dans le temps donné (différent de celui du backup)
    - Utilisation des copies prises lors d'un BACKUP SYSTEM FULL ou DATAONLY
      - RESTORE SYSTEM ne restaure PAS les Logs (donc les copies d'un BACKUP DATAONLY sont suffisantes)
    - Deux phases
      - Phase RESTORE
        - Restaure les volumes Databases à partir de la Version de BACKUP précédent le PIT choisi
      - Phase LOG APPLY
        - Applique les enregistrements Logs pour restaurer les databases jusqu'au point PIT choisi



## DB2 Version 8 : améliorations opérationnelles

- Sécurité Multi-niveaux par ligne (Row Granularity Multilevel Security)
  - ▶ Colonne dans la table définie AS SECURITY LABEL
    - chaque ligne a un label de sécurité spécifique
    - le label de sécurité est obtenu de RACF
  - ▶ Hiérarchie établie dans le gestionnaire de sécurité externe (RACF)



SECURITY LABEL	COL1	COL2	COL3
RAINBOW	56	7	76
RAINBOW	24	56	65
CIEL	113	456	56
MAUVE	3	4	5
JAUNE	24	38	45
PASTEL	56	46	45

Un utilisateur avec une autorisation RAINBOW aura le droit d'accéder à toutes les lignes

Un utilisateur avec une autorisation PASTEL aura le droit d'accéder aux lignes PASTEL, CIEL, ROSE, MAUVE

- ▶ Prérequis: z/OS V1R5 et Security Server (RACF)

## DB2 Version 8: améliorations Data Sharing

- "Batching" des écritures et castouts GBP
  - ▶ Ecriture/castout de plusieurs pages en une seule opération CF
  - ▶ Prérequis: z/OS V1R4 et CFLEVEL=12
- Pages modifiées écrites dans le GBP à la phase 1 au lieu de la phase 2.
- Contention globale réduite pour les verrous L-locks sur les Tablespaces
  - ▶ Réduction des contentions niveau XES entre les membres
- Restart Light pour résolution des Threads Indoubt

→ Meilleures performances



## DB2 Version 8: améliorations réseau

- Réplication:
  - ▶ Capture basée sur du multi-threading
  - ▶ Utilisation de la fonction Cross-Loader par la fonction APPLY
  - ▶ Nouvelles fonctions d'administration
    - plus simple à utiliser, plus performante
- Java:
  - ▶ Driver de type 2 et de type 4
  - ▶ Support de JDBC/SQLJ 3.0
    - meilleure intégration avec WebSphere et Java
- XML Publishing
  - ▶ Fonctions construites SQL permettant la génération de données XML à partir de données relationnelles
    - réduction des développements applicatifs pour générer des données XML permettant l'échange d'informations



## DB2 Version 8: support UNICODE

- Quelques Mythes
    - ▶ on doit convertir toutes ses données en Unicode
    - ▶ le catalogue et les données vont doubler de taille
    - ▶ ceux qui ne veulent que des données en EBCDIC ne sont pas impactés
  
  - Ce qui se passe en DB2 V8
    - ▶ Catalogue converti en Unicode
      - affecte ordre de tri
    - ▶ Interprétation ("Parsing") Unicode
      - ordres SQL des DBRMs et SYSSTMTs en unicode
    - ▶ Précompilateur: paramètre NEWFUN
- Meilleure intégration avec les technologies Java et les ISVs

# DB2 Version 8: support UNICODE

- UNICODE
    - ▶ encodage UTF-8 (Unicode Transformation Format in 8 bits)
      - utilise de 1 à 4 octets pour représenter un caractère
    - ▶ encodage UTF-16 pour GRAPHIC/VARGRAPHIC
  
  - Impacts:
    - ▶ Caractères stockés sur plus de place
      - Caractères standards (A,B,C, ...) : toujours sur 1 octet
      - Caractères spécifiques (é, à, ç, ...): sur 2 octets
    - ▶ Ordre de tri différent
      - semblable à l'ASCII
  
  - Catalogue converti en Unicode
    - ▶ au moment de la migration
- Meilleure intégration avec les technologies Java et les ISVs



## DB2 Version 8: support UNICODE

- Interprétation des ordres SQL
  - ▶ DB2 convertit l'ordre SQL en Unicode pour le traiter
    - SQL statique: par le précompilateur
    - SQL dynamique: pendant le PREPARE/EXECUTE IMMEDIATE
  
- Préparation des programmes
  - ▶ paramètre NEWFUN du précompilateur
    - NEWFUN(NO):
      - ordres SQL en EBCDIC dans les DBRMs
      - syntaxe V8 refusée
    - NEWFUN(YES):
      - ordres SQL en UNICODE dans les DBRMs
      - syntaxe V8 autorisée



## DB2 Version 8: support UNICODE

- Conversion
  - ▶ Local
    - en général, pas de conversion pour les applications locales sauf si:
      - données en ASCII ou UNICODE
      - Option du Bind (Application Encoding)
  - ▶ Eloigné
    - automatiquement quand nécessaire via DRDA
- Services de conversion
  - ▶ Support z/OS pour Unicode: Conversion Services
    - configuration nécessaire: images de conversion à construire



## DB2 Version 8: support UNICODE et Catalogue

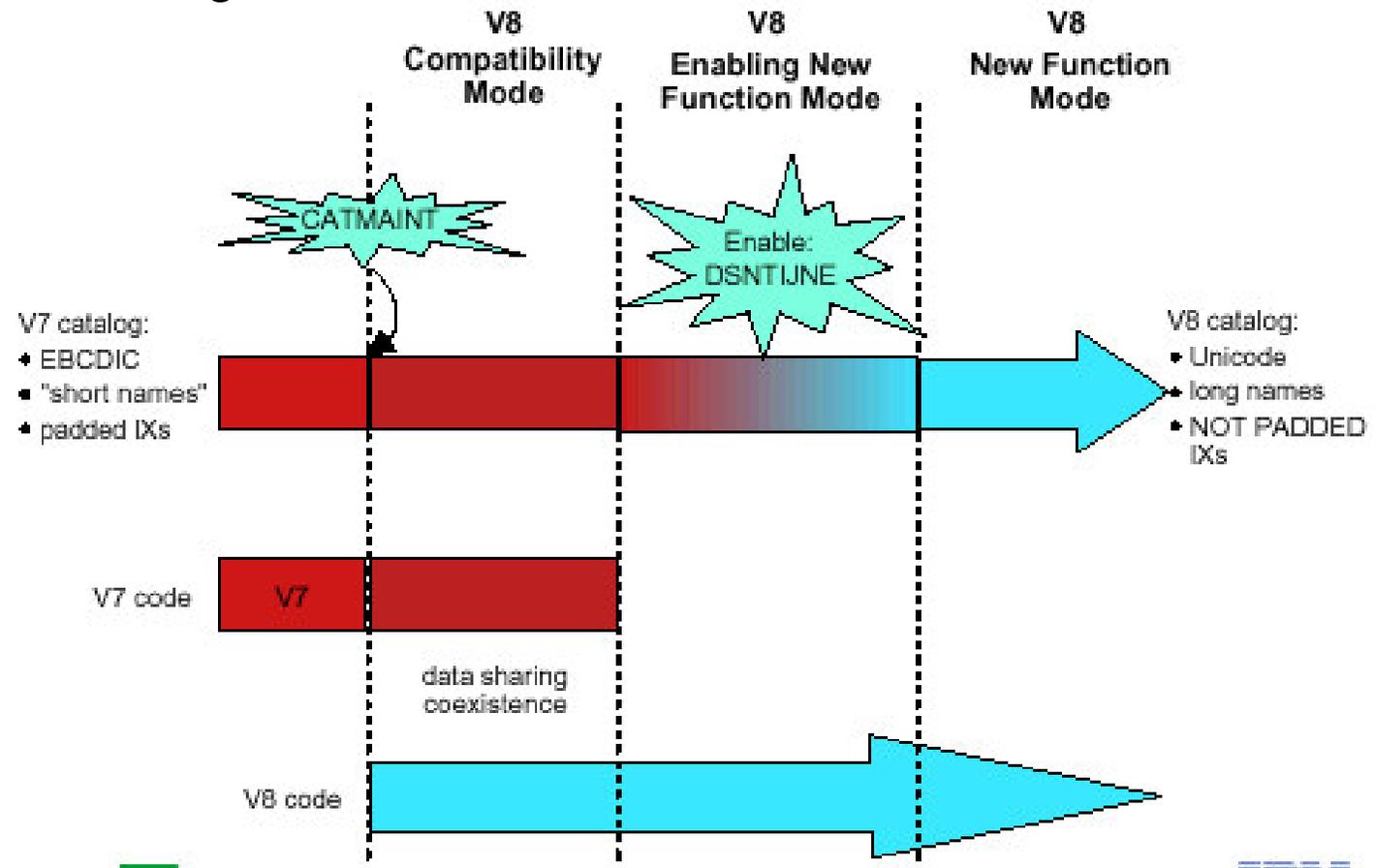
- CATALOGUE DB2
  - ▶ REORG Online possible
  - ▶ Stocké en Unicode
  - ▶ Support des longs noms
  
- Plage de données différentes
  - ▶ `SELECT NAME FROM SYSIBM.SYSTABLES  
WHERE NAME < 'T0' AND NAME > 'TA'`
    - V7: retourne toutes les tables commençant par T et suivies d'une lettre
    - V8: ne retourne aucune ligne !
  
- Ordre de tri différent
  - ▶ `SELECT NAME FROM SYSIBM.SYSTABLES  
WHERE NAME LIKE 'T%'  
ORDER BY NAME`
    - V7 (EBCDIC): retourne TA, TB, T1, T2
    - V8 (UNICODE): retourne T1, T2, TA, TB

## DB2 Version 8: Migration

- La migration se fait en plusieurs étapes
- Migration en Mode "Compatibility"
  - ▶ CATMAINT
  - ▶ système DB2 en version 8 MAIS pas d'utilisation des nouvelles fonctionnalités
  - ▶ coexistence Data Sharing
  - ▶ fallback possible en V7
- Migration en Mode "Enabling New Function"
  - ▶ CATENFM: conversion du Catalogue et de la Directory
    - établissement du mode "Enabling New Function"
    - plusieurs étapes pour chaque tablespace
    - à la fin, établissement du mode "New Function"
  - ▶ fallback vers Mode "Compatibility"
- Mode "New Function"
  - ▶ La migration est terminée: les fonctionnalités de DB2 V8 sont disponibles

# DB2 Version 8: Migration

## ■ Etapes de migration



## En résumé, DB2 Version 8:

- ***Casse les limites actuelles:***
  - ▶ mémoire, longs noms, ordres SQL, partitions, ...
- ***Améliore la disponibilité et les performances***
  - ▶ modifications dynamiques des caractéristiques des objets
  - ▶ redémarrage automatique et meilleures performances des utilitaires
  - ▶ Sauvegarde et Backup au niveau du système
- ***Améliore les possibilités SQL pour support des ERPs et compatibilité vers les autres SGBDs***
  - ▶ Insert et Fetch multi-lignes
  - ▶ Curseurs scrollables dynamiques
  - ▶ SEQUENCES
- ***S'intègre avec les nouvelles techniques Java et WebSphere***



## Informations complémentaires

### ■ Brochures DB2 V8

- ▶ Toutes les brochures DB2 V8
  - <http://www-306.ibm.com/software/data/db2/zos/v8books.html>
- ▶ dont le DB2 V8 Release Planning Guide
  - <http://publib.boulder.ibm.com/cgi-bin/bookmgr/BOOKS/dsnrgj11/CCONTENTS>

### ■ Redbooks

- ▶ DB2 V8 for z/OS Technical Preview (SG24-6871)
  - <http://www.redbooks.ibm.com/abstracts/sg246871.html?Open>
- ▶ DB2 UDB for z/OS Version 8: Everything You Ever Wanted to Know, ... and More (SG24-6079)
  - <http://www.redbooks.ibm.com/abstracts/sg246079.html?Open>

### ■ Cours "Delta DB2 V8" - DB908FR (3 jours)

- ▶ 06 au 08 avril
- ▶ 27 au 29 juin
- ▶ 03 au 05 octobre
- ▶ 28 au 30 novembre

