

# PROJET DE RECHERCHE SUR LES SGBD DU MARCHÉ – SYBASE

**Clara BENEDETTINI, TD2**

**Hugo RULLIÈRE, TD2**

**Lilian PETIT, TD2**

**Julien BLETRY, TD3**

**Arman HAKOBYAN, TD3**

**SMUAGE01-B – Fonctionnement d'un SGBD – Mr GALLI, Mr MOPOLO-MOKE**

**MIAGE Nice – Avril 2023**

## TABLE DES MATIÈRES

|   |           |
|---|-----------|
| <b>PREMIÈRE PARTIE – ÉVALUATION DU SGBD RELATIONNEL SYBASE</b>                      | <b>3</b>  |
| I – Identification du SGBD  | 3         |
| II – Installation et prise en main du SGBD  | 3         |
| III – Architecture fonctionnelle du SGBD choisi                                     | 5         |
| 3.1 - Dessins d'architecture  | 5         |
| 3.2 - Les caches mémoires et leurs rôles  | 5         |
| 3.3 - Les processus gravitant autour du cache mémoire et leur rôle                  | 6         |
| 3.4 - Gestion des connexions  | 6         |
| 3.5 - Processus d'exécution des requêtes  | 7         |
| IV – LE DICTIONNAIRE DE DONNÉES DU MOTEUR CHOISI                                    | 8         |
| 4.1 - Description du dictionnaire   | 8         |
| 4.2 - Liste de ces tables et leurs rôles  | 8         |
| 4.3 - Illustration avec des exemples et démos concrets.                             | 8         |
| V – ORGANISATION PHYSIQUE DU SGBD   | 9         |
| 5.1 - Les différents types de fichiers qui composent ce moteur et leur rôle         | 9         |
| 5.2 - Visualisation des informations sur ces fichiers en consultant le dictionnaire | 10        |
| VI – ORGANISATION LOGIQUE DES DONNÉES   | 10        |
| 6.1 - Comment sont organisées physiquement les données des tables                   | 10        |
| 6.2 - Comment sont organisées physiquement les données des index                    | 11        |
| 6.3 - Comment sont organisées physiquement les données d'annulation                 | 11        |
| 6.4 - Comment sont organisées physiquement les données temporaires                  | 11        |
| VII – GESTION DE LA CONCURRENCE D'ACCÈS   | 12        |
| 7.1 - Notion de transaction   | 12        |
| 7.2 - Support des propriétés ACID   | 12        |
| 7.3 - Technique de gestion de la concurrence d'accès                                | 12        |
| 7.4 - La sérialisation  | 13        |
| VIII – GESTION DES TRANSACTIONS DISTRIBUÉES   | 13        |
| 8.1 - Architecture  | 13        |
| 8.2 - Positionnement par rapport aux 12 règles de commit                            | 13        |
| IX – GESTION DE LA REPRISE SUR PANNE  | 13        |
| X – TECHNIQUES D'INDEXATION   | 14        |
| XI – OPTIMISATION DE REQUÊTES   | 14        |
| XII – SOURCES   | 14        |
| <b>DEUXIÈME PARTIE – PROJET DE DÉVELOPPEMENT DE TECHNIQUES D'INDEXATIONS</b>        | <b>15</b> |
| I – Réalisation du projet   | 15        |
| II – Rapport de recherche   | 15        |

## PREMIÈRE PARTIE – ÉVALUATION DU SGBD RELATIONNEL SYBASE

### I – Identification du SGBD

**Nom :** L'actuel nom de ce SGBD est SAP ASE, il était anciennement connu sous les noms Sybase SQL Server, Sybase DB, ou Sybase ASE.

**Date de parution :** La première version est sortie en 1987. La dernière version, la 16, est sortie en 2014.

**Créateur :** La société Sybase.

**Propriétaire :** En 2010, la société allemande SAP rachète Sybase, elle en est l'actuelle propriétaire.

**Modèles de données supportées :** le SGBD SAP ASE ne supporte que le modèle de données relationnel.

### II – Installation et prise en main du SGBD

Les systèmes d'exploitation (OS) pris en charge pour l'installation d'une base de données Sybase dépendent de la version de Sybase que vous souhaitez utiliser. Cependant, en général, Sybase prend en charge plusieurs systèmes d'exploitation, notamment :

- **Windows :** Sybase peut être installé sur différentes versions de Windows, notamment Windows Server et Windows Desktop, telles que Windows 10, Windows Server 2012, Windows Server 2016, etc.
- **Linux :** Sybase prend en charge plusieurs distributions Linux, telles que Red Hat Enterprise Linux, SUSE Linux Enterprise Server, CentOS, Oracle Linux, etc.
- **Unix :** Sybase peut être installé sur des systèmes Unix, tels que IBM AIX, HP-UX et Solaris.

Durant notre projet de recherche, nous avons mis en place des bases de données Sybase sur les systèmes d'exploitations Windows 10 et 11. L'installation est facile et ne prends que quelques minutes.

On télécharge l'archive contenant l'installateur sur le site officiel de SAP : <https://www.sap.com/france/products/technology-platform/sybase-ase/trial.html>

On lance l'installateur et on clique sur suivant tout le long et on définit le mot de passe de l'administrateur pour avoir l'installation par défaut. Cette installation est recommandée et convient à la plupart des utilisateurs.

Une fois l'installation terminée, on lance le logiciel client **Interactive SQL**, installé en même temps que le SGBD Sybase. Ce logiciel nous permet d'interroger, de manipuler et d'administrer les données dans les bases de données Sybase et SQL Anywhere (un autre SGBD de SAP). Une fois le client ouvert, on saisit l'utilisateur et le mot de passe. On se connecte en tant qu'administrateur systèmes "sa".

Connect

**Connect to SAP Adaptive Server Enterprise**  
[Change database type](#)

Identify yourself to the server with user name and password.

User name:

Password:

Specify the server to connect to. Settings...

Select a server from the dropdown list, or enter host name and port number separated by ":" (for example, "ARONDIGHT:5000").

Server name:

Connection info: Host: ARONDIGHT Port: 5000 Details >>

Advanced >> Tools Connect Cancel Help

L'interface d'Interactive SQL est quelque peu archaïque et basique, elle se résume par un champ de saisie pour les requêtes SQL en haut, et un tableau affichant le résultat des requêtes en bas.

TEST (sa) / master (dbo) - Interactive SQL

File Edit SQL Data Favorites Tools Window Help

master

SQL Statements

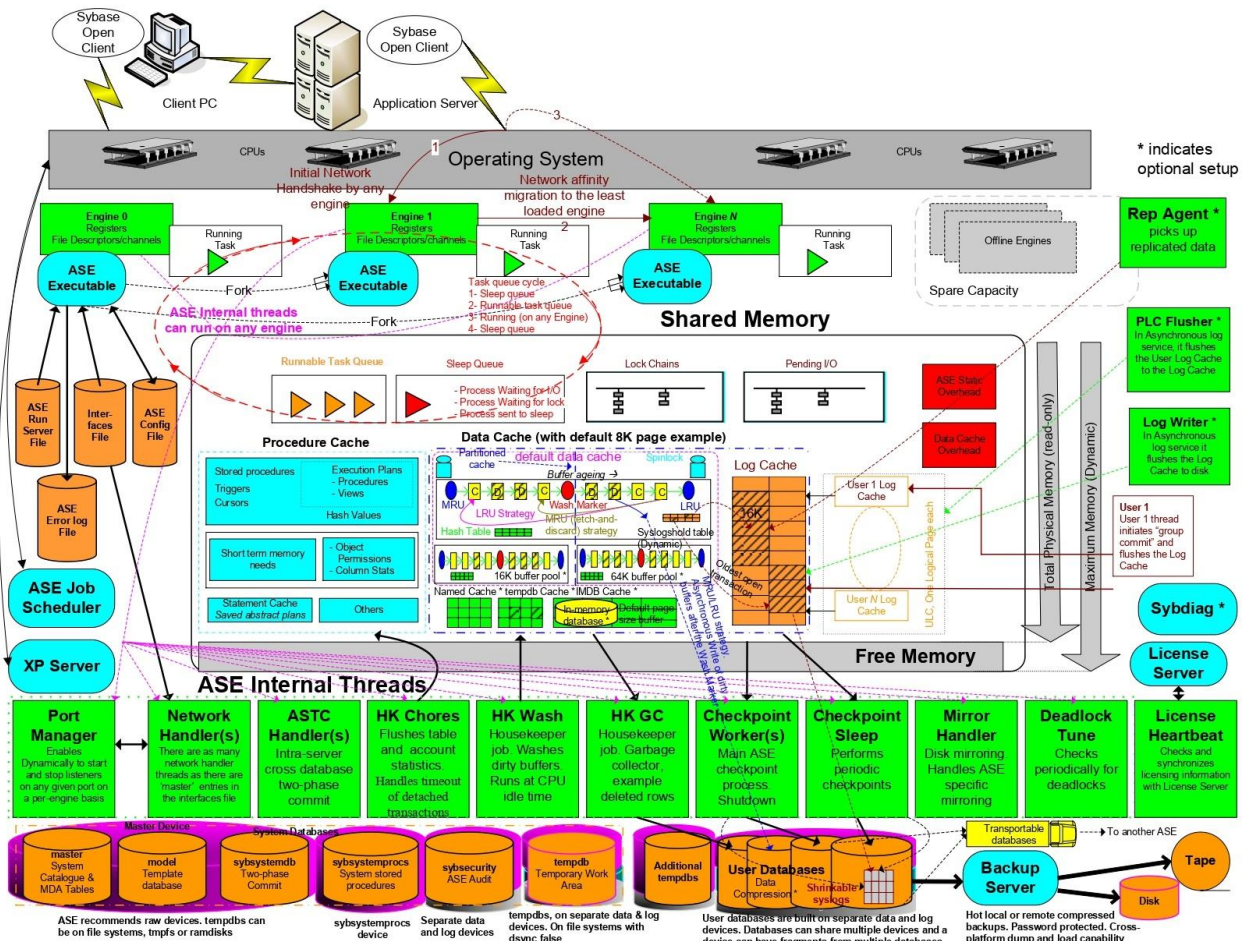
Results

Messages

Line 1 Column 1

### III – Architecture fonctionnelle du SGBD choisi

#### 3.1 - Dessins d'architecture



#### 3.2 - Les caches mémoires et leurs rôles

Dans un SGBD Sybase, les caches mémoires sont utilisés pour stocker temporairement des données fréquemment utilisées pour améliorer les performances. Ces caches peuvent être optimisés pour s'adapter au mieux à environnement précis.

Plusieurs caches mémoires existes, les voici :

- **Cache de données :** Utilisé pour stocker les pages de données souvent utilisées en mémoire vive. Récupérées du disque lorsqu'elles sont nécessaires et stockées dans le cache de données pour un accès plus rapide à l'avenir. Ce cache permet de réduire les temps d'accès aux disques, qui sont beaucoup plus lents que la mémoire vive.

- **Cache de procédures stockées** : Utilisé pour stocker en mémoire les plans d'exécution des procédures stockées. Ce cache permet d'éviter la compilation à chaque exécution de la procédure stockée, ce qui améliore considérablement les performances de l'application.
- **Cache de requêtes** : Utilisé pour stocker en mémoire les résultats de requêtes fréquemment exécutées. Ce cache permet d'accélérer les temps de réponse des requêtes en évitant de devoir exécuter à nouveau la même requête chaque fois qu'elle est appelée.

### 3.3 - Les processus gravitant autour du cache mémoire et leur rôle

Dans un SGBD Sybase, plusieurs processus gravitent autour du cache mémoire pour gérer son utilisation et son fonctionnement. Ces processus sont :

- **Le gestionnaire de cache** : Responsable de la gestion du cache de données en mémoire vive. Surveille l'utilisation du cache et décide quelles pages doivent être stockées en mémoire ou retirées du cache pour faire de la place à de nouvelles pages. Surveille également les performances du cache pour s'assurer qu'il fonctionne de manière optimale.
- **Le processus de plan d'exécution de la procédure stockée** : Génère et stock en mémoire le plan d'exécution de la procédure. Ce plan d'exécution est stocké dans le cache de procédures stockées et est utilisé pour exécuter la procédure à l'avenir. Si un nouveau plan d'exécution doit être généré, le processus est appelé à nouveau.
- **Le processus d'optimisation de requête** : Responsable de la génération des plans d'exécution pour les requêtes qui sont exécutées dans la base de données. Utilise des algorithmes sophistiqués pour trouver le plan d'exécution le plus efficace pour chaque requête.
- **Le processus de récupération de page** : Appelé quand nécessaire pour récupérer la page à partir du disque et la stocker en mémoire. Surveille par ailleurs l'utilisation du cache de données et décide quelles pages doivent être stockées en mémoire ou retirées du cache pour faire de la place à de nouvelles pages.

### 3.4 - Gestion des connexions

Dans un SGBD Sybase, les principaux aspects de la gestion des connexions sont :

- **Connexion initiale** : Lorsqu'un utilisateur se connecte à une instance d'un SGBD Sybase, une session est créée avec un identifiant de session unique. La session est associée à un thread dans le système et toutes les requêtes et les transactions effectuées par l'utilisateur sont exécutées dans cette session.
- **Limitation des connexions** : Permet de limiter le nombre de connexions simultanées au système en configurant une limite maximale. Cela permet d'éviter une surcharge du système et de garantir que les ressources du système sont utilisées de manière optimale.

- **Pooling de connexions** : Permet de prendre en charge le pooling de connexions pour réutiliser les connexions existantes plutôt que d'en créer de nouvelles à chaque fois qu'un utilisateur se connecte. Le pooling de connexions permet de réduire le temps nécessaire pour établir une connexion et d'améliorer les performances du système.
- **Gestion des transactions** : Permet de gérer les transactions pour garantir l'intégrité des données. Si une transaction échoue, le SGBD effectue un rollback pour annuler tous les changements apportés à la base de données. Si une transaction est réussie, le SGBD effectue un commit pour valider les changements dans la base de données.
- **Déconnexion** : Lorsqu'un utilisateur se déconnecte d'une instance d'un SGBD Sybase, la session associée est fermée et toutes les ressources utilisées par la session sont libérées. Si une transaction est en cours, le SGBD effectue un rollback pour annuler tous les changements apportés à la base de données.

### 3.5 - Processus d'exécution des requêtes

Le processus des requêtes dans un SGBD Sybase suit plusieurs étapes pour traiter une requête et renvoyer les résultats à l'utilisateur. Les étapes suivies sont :

- **Analyse syntaxique** : Lorsqu'une requête est soumise, le SGBD commence par analyser la syntaxe de la requête pour s'assurer qu'elle est correcte. Si la syntaxe de la requête est incorrecte, une erreur est renvoyée à l'utilisateur et la requête n'est pas exécutée.
- **Analyse sémantique** : Si la syntaxe de la requête est correcte, le SGBD effectue une analyse sémantique pour s'assurer que la requête est logiquement correcte. Par exemple, il vérifie que les tables et les colonnes référencées existent dans la base de données et que les types de données sont compatibles.
- **Optimisation de requête** : Une fois que la requête est validée, le SGBD commence le processus d'optimisation de requête pour déterminer le plan d'exécution optimal. Le plan d'exécution est un ensemble d'instructions qui spécifient comment récupérer les données de la base de données pour répondre à la requête.
- **Exécution de la requête** : Une fois que le plan d'exécution a été généré, le SGBD commence à exécuter la requête. Les données sont récupérées à partir de la base de données.
- **Renvoi des résultats** : Une fois que les résultats ont été générés, le SGBD les renvoie à l'utilisateur.
- **Fermeture de la connexion** : Une fois que les résultats ont été renvoyés, le SGBD ferme la connexion associée à la session.



## IV – LE DICTIONNAIRE DE DONNÉES DU MOTEUR CHOISI

### 4.1 - Description du dictionnaire

Dans un SGBD Sybase, le dictionnaire de données est une collection de tables système qui contient des informations sur la structure de la base de données et les objets qui y sont stockés. Ce dictionnaire contient les tables systèmes, les informations sur les objets, les métadonnées et les accès.

### 4.2 - Liste de ces tables et leurs rôles

Dans un SGBD Sybase, les tables système sont :

- **sysobjects** : Contient une liste de tous les objets de la base de données, tels que les tables, les vues, les index, les procédures stockées et les utilisateurs.
- **syscolumns** : Contient des informations sur les colonnes de toutes les tables de la base de données, telles que les noms, les types de données, les tailles et les valeurs par défaut.
- **sysindexes** : Contient des informations sur les index de toutes les tables de la base de données, telles que les noms, les types, les clés et les options de configuration.
- **sysconstraints** : Contient des informations sur les contraintes de toutes les tables de la base de données.
- **sysusers** : Contient des informations sur les utilisateurs et les rôles de sécurité de la base de données, tels que les noms, les mots de passe, les autorisations et les propriétaires.
- **sysprocedures** : Contient des informations sur toutes les procédures stockées de la base de données, telles que les noms, les définitions, les paramètres et les propriétaires.
- **syscomments** : Contient le code source des procédures stockées, des déclencheurs et d'autres objets de la base de données.
- **sysdepends** : Contient des informations sur les dépendances entre les objets de la base de données, tels que les relations entre les tables et les procédures stockées.
- **systypes** : Contient des informations sur les types de données pris en charge par la base de données, tels que les noms, les tailles, les précisions et les échelles.
- **sysdatabases** : Contient des informations sur toutes les bases de données du serveur, telles que les noms, les tailles, les propriétaires et les options de configuration.
- **syslogins** : Contient des informations sur tous les utilisateurs et les comptes de connexion au serveur.

### 4.3 - Illustration avec des exemples et démos concrets.

Dans un SGBD Sybase, on peut par exemple lister toutes les tables avec la requête SQL suivante :  
**SELECT \* FROM sysobjects WHERE type='U'**



| Results                 |               |     |      |          |         |          |          |          |                         |                         |        |        |        |       |        |       |          |         |          |
|-------------------------|---------------|-----|------|----------|---------|----------|----------|----------|-------------------------|-------------------------|--------|--------|--------|-------|--------|-------|----------|---------|----------|
| name                    | id            | uid | type | userstat | sysstat | indexdel | schemact | sysstat2 | crdate                  | exdate                  | deltrg | instrg | updtgr | setrg | ckfrst | cache | audflags | objpare | versions |
| 1/spt_values            | 992,003,534   | 1U  |      | 0        | 83      | 1        | 0        | 73,728   | Mar 1, 2023 12:16:44 AM | Mar 1, 2023 12:16:44 AM | 0      | 0      | 0      | 0     | 0      | 0     | 0        | 0       | 0 (NULL) |
| 2/spt_monitor           | 1,024,003,648 | 1U  |      | 0        | 67      | 0        | 0        | 74,240   | Mar 1, 2023 12:16:45 AM | Mar 1, 2023 12:16:45 AM | 0      | 0      | 0      | 0     | 0      | 0     | 0        | 0       | 0 (NULL) |
| 3/spt_limit_types       | 1,072,003,819 | 1U  |      | 0        | 67      | 0        | 0        | 74,240   | Mar 1, 2023 12:16:46 AM | Mar 1, 2023 12:16:46 AM | 0      | 0      | 0      | 0     | 0      | 0     | 0        | 0       | 0 (NULL) |
| 4/sybkensedbg           | 1,120,003,890 | 1U  |      | 0        | 67      | 0        | 0        | 74,240   | Mar 1, 2023 12:16:46 AM | Mar 1, 2023 12:16:46 AM | 0      | 0      | 0      | 0     | 0      | 0     | 0        | 0       | 0 (NULL) |
| 5/spt_gdbc_table_types  | 1,152,004,104 | 1U  |      | 0        | 67      | 0        | 0        | 74,240   | Mar 1, 2023 12:16:46 AM | Mar 1, 2023 12:16:46 AM | 0      | 0      | 0      | 0     | 0      | 0     | 0        | 0       | 0 (NULL) |
| 6/spt_gdbc_mda          | 1,184,004,218 | 1U  |      | 0        | 99      | 1        | 0        | 73,728   | Mar 1, 2023 12:16:46 AM | Mar 1, 2023 12:16:46 AM | 0      | 0      | 0      | 0     | 0      | 0     | 0        | 0       | 0 (NULL) |
| 7/spt_gdbc_conversion   | 1,232,004,389 | 1U  |      | 0        | 67      | 0        | 0        | 74,240   | Mar 1, 2023 12:16:46 AM | Mar 1, 2023 12:16:46 AM | 0      | 0      | 0      | 0     | 0      | 0     | 0        | 0       | 0 (NULL) |
| 8/jdbc_function_escapes | 1,264,004,503 | 1U  |      | 0        | 67      | 0        | 0        | 73,728   | Mar 1, 2023 12:16:46 AM | Mar 1, 2023 12:16:46 AM | 0      | 0      | 0      | 0     | 0      | 0     | 0        | 0       | 0 (NULL) |
| 9/monCPC                | 1,180,528,208 | 1U  |      | 0        | 67      | 0        | 0        | 77,312   | Mar 1, 2023 12:16:51 AM | Mar 1, 2023 12:16:51 AM | 0      | 0      | 0      | 0     | 0      | 0     | 0        | 0       | 0 (NULL) |
| 10/monTables            | 1,664,005,928 | 1U  |      | 0        | 67      | 0        | 0        | 76,800   | Mar 1, 2023 12:16:50 AM | Mar 1, 2023 12:16:50 AM | 0      | 0      | 0      | 0     | 0      | 0     | 0        | 0       | 0 (NULL) |

Autre exemple, on peut lister toutes les vues avec la requête SQL suivante : **SELECT \* FROM sysobjects WHERE type='V'**

Autre exemple, on peut lister toutes les contraintes sur une table précise avec la requête SQL suivant : **SELECT \* FROM sysconstraints WHERE id=OBJECT\_ID('table\_name')**

## V – ORGANISATION PHYSIQUE DU SGBD

### 5.1 - Les différents types de fichiers qui composent ce moteur et leur rôle

Le moteur d'un SGBD Sybase est composé de plusieurs types de fichiers qui sont utilisés pour stocker les données, les index et les informations système. Ces types de fichiers sont :

- **Fichiers de données** : Utilisés pour stocker les données de la base de données, telles que les tables, les vues et les procédures stockées. Les fichiers de données sont généralement organisés en groupes de fichiers pour faciliter la gestion de l'espace disque. Les données sont stockées dans des pages de 2 Ko ou 4 Ko, qui peuvent être lues et écrites par le moteur de la base de données.
- **Fichiers journaux** : Utilisés pour enregistrer les transactions effectuées dans la base de données. Les fichiers journaux sont utilisés pour la récupération après une panne ou un arrêt non planifié. Les transactions sont écrites dans le fichier journal avant d'être écrites dans les fichiers de données, assurant ainsi que la base de données peut être récupérée jusqu'à un point de récupération cohérent.
- **Fichiers d'index** : Utilisés pour stocker les index des tables de la base de données. Les index sont utilisés pour accélérer les recherches dans la base de données. Les fichiers d'index sont organisés en groupes de fichiers et stockent les pages d'index, qui contiennent des informations sur les données stockées dans les fichiers de données.
- **Fichiers de sauvegarde** : Utilisés pour stocker une copie de la base de données à un moment donné. Les fichiers de sauvegarde peuvent être utilisés pour restaurer la base de données en cas de défaillance du système ou de perte de données.
- **Fichiers temporaires** : Utilisés pour stocker des données temporaires lors de l'exécution des requêtes SQL. Les fichiers temporaires sont généralement stockés dans un répertoire dédié sur le disque et sont supprimés après leur utilisation.
- **Fichiers de contrôle** : Utilisés pour stocker des informations sur la structure de la base de données, tels que le nombre de fichiers de données et de journaux, les tailles de bloc, les propriétés des objets de la base de données, ...

- **Fichiers de configuration** : Utilisés pour stocker des informations de configuration pour le serveur de base de données, telles que les paramètres de mémoire, les limites de connexions, les chemins de fichiers de données et autres.

## 5.2 - Visualisation des informations sur ces fichiers en consultant le dictionnaire

Dans un SGBD Sybase, il est possible de visualiser des informations sur les fichiers du moteur en consultant le dictionnaire de données. Les tables contenant ces informations sont :

- **sysdatabases** : Contient des informations sur les bases de données, y compris les noms des fichiers de données et de journaux.
- **sysdevices** : Contient des informations sur les dispositifs de stockage, tels que les disques, les bandes et les fichiers.
- **sysusages** : Contient des informations sur les utilisations des fichiers de données et de journaux, y compris leur emplacement, leur taille et leur état.
- **sysfilegroups** : Contient des informations sur les groupes de fichiers de données, tels que leur nom, leur état et les fichiers qu'ils contiennent.
- **sysfilelist** : Contient une liste de tous les fichiers de données et de journaux utilisés par chaque base de données.
- **sysaltfiles** : Contient des informations sur les fichiers de sauvegarde, tels que leur nom, leur emplacement et leur état.
- **sysbackupdev** : Contient des informations sur les dispositifs de stockage utilisés pour les sauvegardes, tels que les disques, les bandes et les fichiers.
- **sysconfigures** : Contient des informations sur les paramètres de configuration du serveur, tels que la mémoire allouée, le nombre de connexions et les tailles de fichiers.

## VI – ORGANISATION LOGIQUE DES DONNÉES

### 6.1 - Comment sont organisées physiquement les données des tables

Dans un SGBD Sybase, les données des tables sont stockées sur des pages de données organisés de manière séquentielle et numérotées en fonction de l'emplacement dans le disque. Chaque page de données a une taille fixe de 2 ou 4 Ko. Les pages de données sont organisées en unités de stockage logiques appelées "segments".

Il existe différents types de segments :

- Les segments de données : Stockent les données de la table et sont organisés en pages de données.
- Les segments d'index : Stockent les index de la table et sont également organisés en pages de données.
- Les segments mixtes : Contiennent à la fois des données et des index.

## **6.2 - Comment sont organisées physiquement les données des index**

Dans un SGBD Sybase, les index sont stockés dans des pages de données similaires à celles utilisées pour stocker les données de la table. Ces pages sont aussi organisées de la même manière que pour les pages de données des tables.

## **6.3 - Comment sont organisées physiquement les données d'annulation**

Dans un SGBD Sybase, les données d'annulation sont stockées dans un fichier de journal. Le journal des transactions enregistre toutes les modifications apportées à la base de données et permet de restaurer la base de données en cas de panne du système ou d'erreur humaine. Le fichier de journal est organisé en pages de journal, qui sont des unités de stockage logiques semblables aux pages de données. Les pages de journal sont numérotées séquentiellement pour refléter l'ordre dans lequel les transactions ont été enregistrées. Lorsqu'une transaction est exécutée, le SGBD enregistre les modifications dans le fichier de journal avant de les appliquer aux pages de données de la table.

Ces données d'annulation sont gérées par le gestionnaire de transactions du SGBD, qui s'assure que toutes les modifications apportées à la base de données sont enregistrées dans le journal des transactions. Le gestionnaire de transactions garantit également la cohérence des transactions en s'assurant que chaque transaction est complètement exécutée avant de valider ses modifications.

## **6.4 - Comment sont organisées physiquement les données temporaires**

Dans un SGBD Sybase, les données temporaires sont stockées dans des segments de la base de données appelés segments de travail temporaire. Qui sont des espaces de stockage réservés pour les opérations de traitement temporaire, telles que les opérations de tri et de jointure. Le SGBD crée automatiquement ces segments de travail temporaire lorsque des opérations nécessitent de stocker des données temporaires en mémoire ou sur disque. Les segments de travail temporaire peuvent être configurés pour être stockés dans la RAM ou sur disque en fonction de la configuration du système et de la taille des données temporaires.

## VII – GESTION DE LA CONCURRENCE D'ACCÈS

### 7.1 - Notion de transaction

Dans un SGBD Sybase, une transaction est une séquence d'opérations qui sont exécutées de manière cohérente et atomique. Cela signifie que toutes les opérations d'une transaction doivent être exécutées avec succès ou que la transaction entière doit être annulée.

Rien de bien nouveau, la notion de transaction dans un SGBD Sybase est exactement la même que dans les autres SGBD du marché.

### 7.2 - Support des propriétés ACID

Le SGBD Sybase supporte toutes les propriétés ACID (Atomicité, Cohérence, Isolation et Durabilité) pour garantir l'intégrité et la fiabilité des transactions dans la base de données.

### 7.3 - Technique de gestion de la concurrence d'accès

La gestion de la concurrence d'accès dans un SGBD Sybase est réalisée à l'aide de verrous ou locks, qui sont placés sur les ressources (tables, lignes, pages, etc.) pour empêcher les conflits d'accès entre différentes transactions. Les verrous sont utilisés pour gérer la concurrence d'accès en contrôlant l'accès concurrent aux données.

Sybase utilise une approche de **verrouillage bimodal** pour gérer la concurrence d'accès. Dans cette approche, deux modes de verrouillage sont utilisés : le verrouillage en lecture (**shared lock S**) et le verrouillage en écriture (**exclusive lock X**).

Il existe également d'autres types de verrous, qui sont :

- **Page Lock** : Utilisé pour verrouiller une page entière dans une table. Cela permet à une transaction de verrouiller efficacement un grand nombre de lignes en une seule opération.
- **Index Lock** : Utilisé pour verrouiller un index plutôt qu'une table entière. Cela permet à une transaction de verrouiller efficacement un grand nombre de lignes en utilisant un index.
- **Resource Lock** : Utilisé pour verrouiller des ressources autres que des tables et des index, telles que des procédures stockées ou des séquences.
- **Update (U) Lock** : Utilisé pour bloquer les autres transactions de modifier les données verrouillées, mais permet aux autres transactions d'accéder à ces données en lecture seule.
- **Intent Shared (IS) Lock** : Utilisé pour indiquer l'intention de prendre des verrous partagés sur les sous-ressources d'une ressource.

- **Intent Exclusive (IX) Lock** : Utilisé pour indiquer l'intention de prendre des verrous exclusifs sur les sous-ressources d'une ressource.
- **Shared with Intent Exclusive (SIX) Lock** : Utilisé pour acquérir un verrou partagé sur une ressource, tout en indiquant son intention de prendre un verrou exclusif sur une sous-ressource.
- **Exclusive Intent Shared (EIS) Lock** : Utilisé pour indiquer l'intention de prendre un verrou exclusif sur une sous-ressource, tout en maintenant un verrou partagé sur la ressource parente.

#### 7.4 - La sérialisation

Dans un SGBD Sybase, la sérialisation est réalisée à travers l'utilisation de verrous sur les pages de données. Lorsqu'une transaction demande l'accès en lecture ou en écriture à une page de données, le système de gestion de verrous place un verrou exclusif sur cette page pour empêcher l'accès à toute autre transaction. Lorsque la transaction a fini d'utiliser cette page, le verrou est libéré pour permettre à d'autres transactions d'accéder à la page.

### VIII – GESTION DES TRANSACTIONS DISTRIBUÉES

#### 8.1 - Architecture

Sybase prend en charge les transactions distribuées, qui permettent à une transaction de mettre à jour plusieurs bases de données sur différents serveurs. Cette fonctionnalité est prise en charge par le gestionnaire de transactions distribuées (Distributed Transaction Manager - DTM) de Sybase, qui coordonne les transactions entre les différentes bases de données.

#### 8.2 - Positionnement par rapport aux 12 règles de commit

Le SGBD Sybase respecte les 12 règles de Chris J. Date.

### IX – GESTION DE LA REPRISE SUR PANNE

Dans un SGBD Sybase, la reprise sur panne est automatiquement assurée par le système. Même s'il est possible de réaliser une reprise sur panne manuelle, il est conseillé de laisser cette tâche au système.

## **X – TECHNIQUES D'INDEXATION**

Dans un SGBD Sybase, nous pouvons créer des index sur les tables soit en utilisant la commande "create index", soit en utilisant les contraintes d'intégrité unique ou clé primaire de la commande "create table". Cependant, l'indexation par les contraintes d'intégrité ont leurs limites.

## **XI – OPTIMISATION DE REQUÊTES**

Un SGBD Sybase propose plusieurs modes d'optimisation, nous permettant de choisir la stratégie d'optimisation la plus adaptée à notre environnement. À noter que l'optimisation des requêtes n'est pas activée par défaut. L'optimiseur de requêtes se chargeant de l'optimisation est auto-réglable et utilise rarement des worktables pour la matérialisation entre les étapes des opérations. Toutefois, l'optimiseur de requêtes peut utiliser plus de worktables lorsqu'il détermine que les opérations de hachage et de fusion sont plus efficaces.

## **XII – SOURCES**

Pour notre recherche, nous avons principalement utilisé la documentation en ligne proposée par la société SAP. Accessible avec le lien suivant :

<https://infocenter.sybase.com/help/index.jsp?topic=/com.sybase.infocenter.help.ase.16.0/doc/html/ti/tle.html>

## DEUXIÈME PARTIE – PROJET DE DÉVELOPPEMENT DE TECHNIQUES D'INDEXATIONS

### I – Réalisation du projet

Pour mettre en place une recherche séquentielle, nous avons écrit une fonction qui parcourt les éléments d'une collection les uns après les autres jusqu'à ce que l'élément recherché soit trouvé ou que la fin de la collection soit atteinte. Notre collection est ici un fichier, qui est donc parcouru ligne par ligne.

Pour ce qui est de la recherche par index, nous avons écrit une fonction qui récupère l'élément recherché dans un HashMap, en utilisant une fonction de hachage pour calculer l'indice de stockage correspondant à la clé donnée en entrée.

Le projet de développement peut être récupéré sur le repository GitHub suivant :  
<https://github.com/Master1-MIAGE-UCA/projet-de-recherche-fsgbd-groupe-4-sybase>

### II – Rapport de recherche

Pour notre recherche, 3 jeux de données ont été utilisés. Contenant respectivement 1000, 10 000 et 50 000 lignes. Nous avons recherché pour chaque jeu de données, par index et de manière séquentielle, 100 valeurs tirées aléatoirement. Voici les temps en nanosecondes que nous avons enregistrées :

| Recherche    | Lignes | Temps Min | Temps Max | Temps Moyen |
|--------------|--------|-----------|-----------|-------------|
| Index        | 1000   | 1000      | 13200     | 1929        |
| Index        | 10000  | 1000      | 11800     | 1981        |
| Index        | 50000  | 1600      | 11600     | 2954        |
| Recherche    | Lignes | Temps Min | Temps Max | Temps Moyen |
| Séquentielle | 1000   | 77000     | 3284800   | 338231      |
| Séquentielle | 10000  | 82100     | 3404100   | 915884      |
| Séquentielle | 50000  | 156300    | 12151900  | 2898707     |

Nous pouvons constater, à la suite de cette étude, que la recherche par index est bien plus rapide que la recherche séquentielle.



Nous avons également recherché plusieurs fois, par index et de manière séquentielle, 100 valeurs tirées aléatoirement. Voici les temps en nanosecondes que nous avons enregistrées :

| Recherche    | Essai | Lignes | Temps Min | Temps Max | Temps Moyen |
|--------------|-------|--------|-----------|-----------|-------------|
| Index        | 1     | 1000   | 500       | 18600     | 1346        |
| Index        | 2     | 1000   | 600       | 3800      | 1224        |
| Index        | 3     | 1000   | 600       | 3200      | 1075        |
| Recherche    | Essai | Lignes | Temps Min | Temps Max | Temps Moyen |
| Séquentielle | 1     | 1000   | 68100     | 4492000   | 335651      |
| Séquentielle | 2     | 1000   | 59400     | 470600    | 146737      |
| Séquentielle | 3     | 1000   | 77300     | 598000    | 166033      |

Nous pouvons constater, à la suite de cette étude, que la recherche par index est bien plus constante en temps de recherche que la recherche séquentielle.