



Projet de recherche
Système de Gestion de Base de Données
MySQL - SQLServer - Oracle

Tuteur : Gregory GALLI / Gabriel
MOPOLO
Diplôme : Master 1 MIAGE
Université de Nice/Sophia Antipolis
Année : 2020/2021

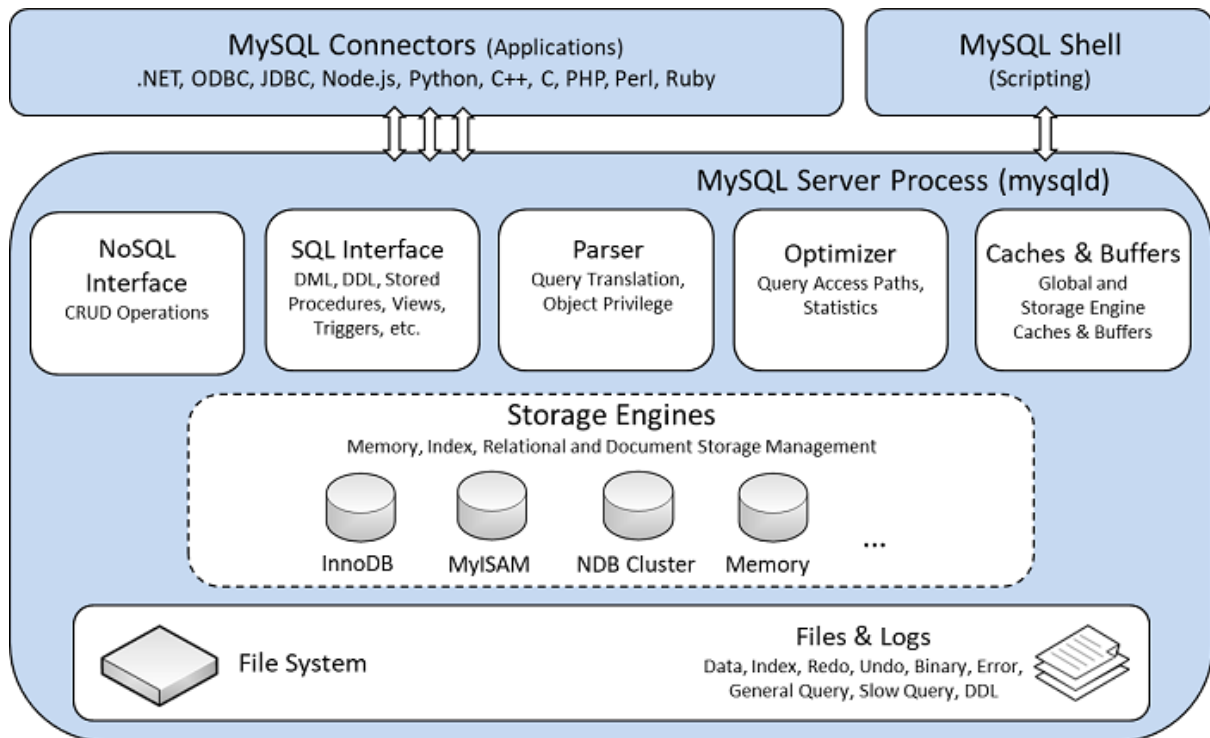
Nicolas Meyer

TABLE DES MATIÈRES

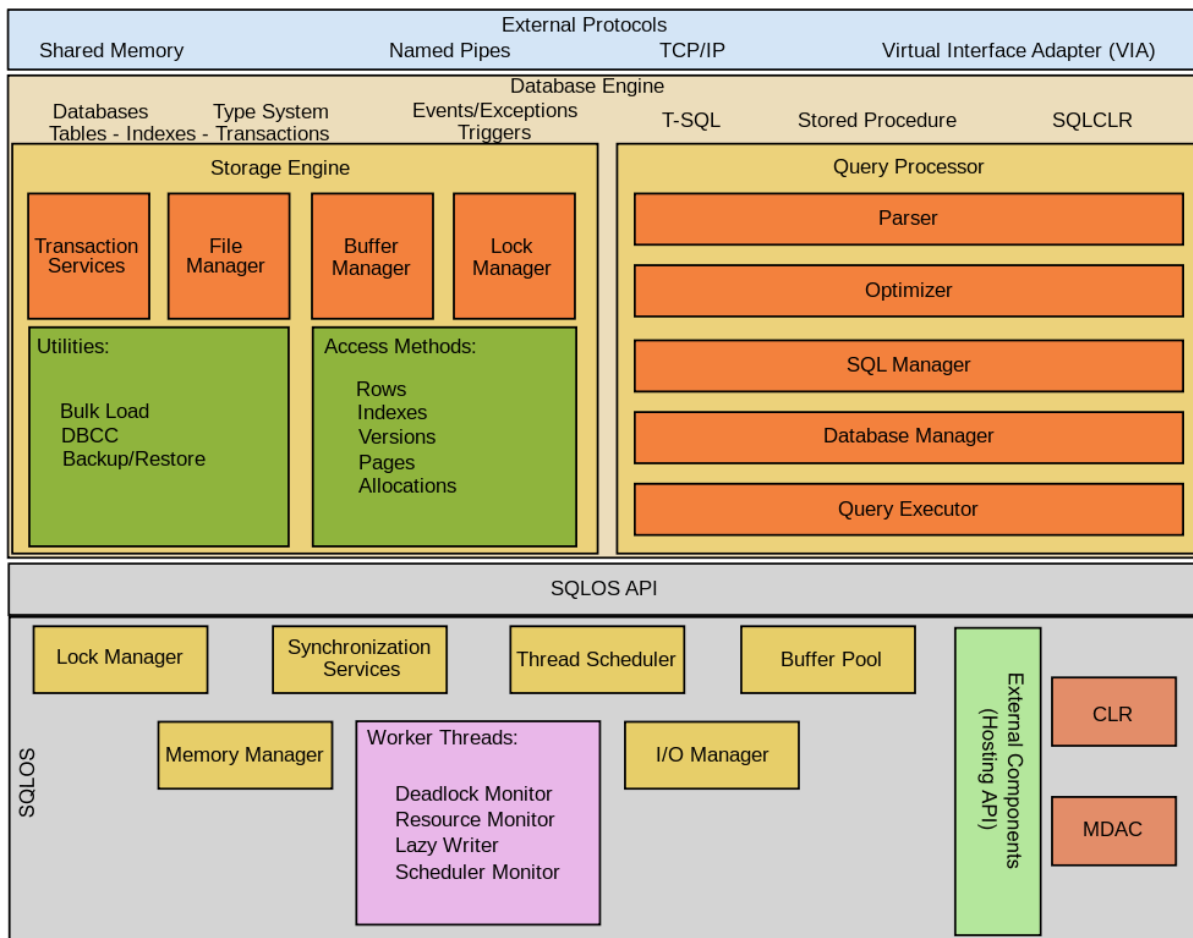
RECHERCHE DES COMPOSANTS DES SGBD	3
MYSQL	3
SQL SERVER	4
DICTIONNAIRE DE DONNÉES	4
TABLES SYSTÈMES	4
RECHERCHES DANS LE DICTIONNAIRE	6
Liste des tables, colonnes, contraintes et indexes	6
Liste des colonnes, contraintes, indexes d'une table	8
Liste des utilisateurs de la base de données	9
RECHERCHE DES TYPES DE FICHIERS	10
MYSQL	10
Identification et description de chaque type de fichier	10
Description et schémas.	11
SQL SERVER	12
Identification et description de chaque type de fichier	12
Description et schémas.	13
ORACLE - REQUÊTES.	13
Fichier de contrôle	13
Fichier de données	14
Fichier Redo Log	14
RECHERCHE SUR LES CACHES MÉMOIRES	15
MYSQL	15
Identification et description de la gestion de mémoire	15
Simulation	15
SQL SERVER	15
Identification et description de la gestion de mémoire	15
Simulation	16
ORACLE - ZONES MÉMOIRES.	16
TRANSACTIONS	17
REQUÊTES DE PREUVES DES ANOMALIES	17
Update Loss	17
Dirty Read	18
Inconsistency Read	19
Ghost update	20
PROPOSITION DE SOLUTIONS	20
BIBLIOGRAPHIE	20

I. RECHERCHE DES COMPOSANTS DES SGBD

A. MYSQL



B. SQL SERVER



II. DICTIONNAIRE DE DONNÉES

A. TABLES SYSTÈMES

Excel de la pré-cr  ation du dictionnaire de donn  e :

<https://docs.google.com/spreadsheets/d/1eKBmnoSKVSSUdToomieVNqvgImy252II41leg1ZUyd0/edit?usp=sharing>

Script SQL de cr  ation du dictionnaire de donn  e :

(<https://pastebin.com/k8UcgqBk>)

```
-- BY NICOLAS MEYER - M1 MIAGE TD 02 --

DROP TABLE META_TABLES CASCADE CONSTRAINTS;
CREATE TABLE META_TABLES (
    idTable NUMBER(4)
```

```

        CONSTRAINT pk_meta_tables PRIMARY KEY
        CONSTRAINT nn_meta_tables_idTable NOT NULL,
nomTable varchar2(50)
        CONSTRAINT nn_meta_tables_nomTable NOT NULL
        CONSTRAINT chk_meta_tables_nomTable CHECK (nomTable LIKE
'[a-zA-Z][a-zA-Z]%' ),
        tableOwner varchar2(50)
        CONSTRAINT nn_meta_tables_tableOwner NOT NULL
);

DROP TABLE META_COLUMNS CASCADE CONSTRAINTS;
CREATE TABLE META_COLUMNS (
        idColonne NUMBER(4)
        CONSTRAINT pk_meta_columns PRIMARY KEY
        CONSTRAINT nn_meta_columns_idColonne NOT NULL,
        idTable NUMBER(4)
        CONSTRAINT fk_meta_columns_tables REFERENCES
META_TABLES(idTable) ON DELETE CASCADE
        CONSTRAINT nn_meta_columns_idTable NOT NULL,
        nomColonne varchar2(50)
        CONSTRAINT nn_meta_columns_nomColonne NOT NULL
        CONSTRAINT chk_meta_columns_nomColonne CHECK (nomColonne
LIKE '[a-zA-Z][a-zA-Z]%' ),
        typeColonne varchar2(50)
        CONSTRAINT nn_meta_columns_typeColonne NOT NULL,
        tailleColonne NUMBER(10)
        CONSTRAINT nn_meta_columns_tailleColonne NOT NULL
);

DROP TABLE META_INDEXES CASCADE CONSTRAINTS;
CREATE TABLE META_INDEXES (
        idIndex NUMBER(4)
        CONSTRAINT pk_meta_indexes PRIMARY KEY
        CONSTRAINT nn_meta_indexes_idIndex NOT NULL,
        idTable NUMBER(4)
        CONSTRAINT fk_meta_indexes_tables REFERENCES
META_TABLES(idTable) ON DELETE CASCADE
        CONSTRAINT nn_meta_indexes_idTable NOT NULL,
        nomIndex varchar2(50)
        CONSTRAINT nn_meta_indexes_nomIndex NOT NULL
        CONSTRAINT chk_meta_indexes_nomIndex CHECK (nomIndex LIKE
'[a-zA-Z][a-zA-Z]%' ),
        typeIndex varchar2(50)
        CONSTRAINT nn_meta_indexes_typeIndex NOT NULL
);

DROP TABLE META_CONSTRAINTS CASCADE CONSTRAINTS;
CREATE TABLE META_CONSTRAINTS (

```

```

idConstraints NUMBER(4)
    CONSTRAINT pk_meta_constraints PRIMARY KEY
    CONSTRAINT nn_meta_constraints_idConstraints NOT NULL,
idTable NUMBER(4)
    CONSTRAINT fk_meta_constraints_tables REFERENCES
META_TABLES(idTable) ON DELETE CASCADE
    CONSTRAINT nn_meta_constraints_idTable NOT NULL,
idColonne NUMBER(4)
    CONSTRAINT fk_meta_constraints_columns REFERENCES
META_COLUMNS(idColonne) ON DELETE CASCADE
    CONSTRAINT nn_meta_constraints_idColonne NOT NULL,
nomContrainte varchar2(50)
    CONSTRAINT nn_meta_constraints_nomContrainte NOT NULL
    CONSTRAINT chk_meta_constraints_nomContrainte CHECK
(nomContrainte LIKE '[a-zA-Z][a-zA-Z]%' ),
typeContrainte varchar2(50)
    CONSTRAINT nn_meta_constraints_typeContrainte NOT NULL,
valeurContrainte varchar2(50)
    CONSTRAINT nn_meta_constraints_valeurContrainte NOT NULL,
statusContrainte varchar2(50)
    CONSTRAINT nn_meta_constraints_id NOT NULL
);

```

B. RECHERCHES DANS LE DICTIONNAIRE

1. Liste des tables, colonnes, contraintes et indexes

- **Tables** (Dans les DBA_*) :

```

SELECT * FROM DBA_TABLES
WHERE OWNER = 'MEYER1M2021' ;

```

	OWNER	TABLE_NAME	TABLESPACE_NAME	CLUSTER_NAME	IOT_NAME	STATUS	PCT_FREE
1	MEYER1M2021	PILOTE	USERS	(null)	(null)	VALID	10
2	MEYER1M2021	AVION	USERS	(null)	(null)	VALID	10
3	MEYER1M2021	VOL	USERS	(null)	(null)	VALID	10

- **Tables** (Dans la table créée) :

```

SELECT * FROM META_TABLES

```

- **Colonnes** (Dans les tables DBA_*) :

```

SELECT * FROM DBA_TAB_COLUMNS

```

```
WHERE OWNER = 'MEYER1M2021' ;
```

	OWNER	TABLE_NAME	COLUMN_NAME	DATA_TYPE	DATA_TYPE_MOD	DATA_TYPE_OWNER	DATA_LENGTH
1	MEYER1M2021	PILOTE	PL#	NUMBER	(null)	(null)	22
2	MEYER1M2021	PILOTE	PLNOM	VARCHAR2	(null)	(null)	12
3	MEYER1M2021	PILOTE	DNAISS	DATE	(null)	(null)	7
4	MEYER1M2021	PILOTE	ADR	VARCHAR2	(null)	(null)	20
5	MEYER1M2021	PILOTE	TEL	VARCHAR2	(null)	(null)	12
6	MEYER1M2021	PILOTE	SAL	NUMBER	(null)	(null)	22
7	MEYER1M2021	AVION	AV#	NUMBER	(null)	(null)	22
8	MEYER1M2021	AVION	AVTYPE	VARCHAR2	(null)	(null)	10
9	MEYER1M2021	AVION	CAP	NUMBER	(null)	(null)	22
10	MEYER1M2021	AVION	LOC	VARCHAR2	(null)	(null)	20

- **Colonnes** (Dans la table créée) :

```
SELECT * FROM META_COLUMNS
```

- **Contraintes** (Dans les DBA_*) :

```
SELECT * FROM DBA_CONSTRAINTS
WHERE OWNER = 'MEYER1M2021' ;
```

	OWNER	CONSTRAINT_NAME	CO...	TABLE_NAME	SEARCH_CONDITION
1	MEYER1M2021	VOL_FK_PILOTE	R	VOL	(null)
2	MEYER1M2021	FK_VOL_AVION#	R	VOL	(null)
3	MEYER1M2021	NL_VOL_PILOTE#	C	VOL	"PILOTE#" IS NOT NULL
4	MEYER1M2021	NL_VOL_AVION#	C	VOL	"AVION#" IS NOT NULL
5	MEYER1M2021	NL_VOL_HD	C	VOL	"HD" IS NOT NULL
6	MEYER1M2021	NL_VOL_HA	C	VOL	"HA" IS NOT NULL
7	MEYER1M2021	VOL_CHK_HA_HD	C	VOL	ha>hd
8	MEYER1M2021	NL_PILOTE_PLNOM	C	PILOTE	"PLNOM" IS NOT NULL
9	MEYER1M2021	NL_PILOTE_DNAISS	C	PILOTE	"DNAISS" IS NOT NULL
10	MEYER1M2021	NL_PILOTE_SAL	C	PILOTE	"SAL" IS NOT NULL
11	MEYER1M2021	CHK_PILOTE_SAL	C	PILOTE	sal < 70000.0
12	MEYER1M2021	NL_AVION_AVTYPE	C	AVION	"AVTYPE" IS NOT NULL
13	MEYER1M2021	NL_AVION_CAP	C	AVION	"CAP" IS NOT NULL
14	MEYER1M2021	NL_AVION_LOC	C	AVION	"LOC" IS NOT NULL
15	MEYER1M2021	CHK_AVION_TYPE	C	AVION	avtype in ('A300','A310','A320','B707','B7...

- **Contraintes** (Dans la table créée) :

```
SELECT * FROM META_CONSTRAINTS
```

- **Indexes** (Dans les DBA_*) :

```
SELECT * FROM DBA_INDEXES
WHERE OWNER = 'MEYER1M2021' ;
```

	OWNER	INDEX_NAME	INDEX_TYPE	TABLE_OWNER	TABLE_NAME	TABLE_TYPE	UNIQUENESS
1	MEYER1M2021	PK_PILOTE	NORMAL	MEYER1M2021	PILOTE	TABLE	UNIQUE
2	MEYER1M2021	UK_PILOTE_PLNOM	NORMAL	MEYER1M2021	PILOTE	TABLE	UNIQUE
3	MEYER1M2021	PK_AVION	NORMAL	MEYER1M2021	AVION	TABLE	UNIQUE
4	MEYER1M2021	PK_VOL	NORMAL	MEYER1M2021	VOL	TABLE	UNIQUE

- **Indexes** (Dans la table créée) :

```
SELECT * FROM META_INDEXES
```

2. Liste des colonnes, contraintes, indexes d'une table

- **Colonnes** (Dans les DBA_*) :

```
SELECT * FROM DBA_TAB_COLUMNS
WHERE OWNER = 'MEYER1M2021'
AND TABLE_NAME = 'VOL' ;
```

	OWNER	TABLE_NAME	COLUMN_NAME	DATA_TYPE	DATA_TYPE_MOD	DATA_TYPE_OWNER	DATA_LENGTH
1	MEYER1M2021	VOL	VOL#	NUMBER	(null)	(null)	22
2	MEYER1M2021	VOL	PILOTE#	NUMBER	(null)	(null)	22
3	MEYER1M2021	VOL	AVION#	NUMBER	(null)	(null)	22
4	MEYER1M2021	VOL	VD	VARCHAR2	(null)	(null)	20
5	MEYER1M2021	VOL	VA	VARCHAR2	(null)	(null)	20
6	MEYER1M2021	VOL	HD	NUMBER	(null)	(null)	22
7	MEYER1M2021	VOL	HA	NUMBER	(null)	(null)	22
8	MEYER1M2021	VOL	DAT	DATE	(null)	(null)	7

- **Colonnes** (Dans la table créée) :

```
SELECT * FROM META_TABLES
INNER JOIN META_COLUMNS ON META_TABLES.idTable =
META_COLUMNS.idTable
AND META_TABLES.nomTable = 'VOL' ;
```

- **Contraintes** (Dans les DBA_*) :

```
SELECT * FROM DBA_CONSTRAINTS
WHERE OWNER = 'MEYER1M2021'
AND TABLE_NAME = 'VOL' ;
```


OWNER	CONSTRAINT_NAME	C...	TABLE_NAME	SEARCH_CONDITION	SEARCH_CONDITION_VC	R_OWNER
1 MEYER1M2021	NL_VOL_PILOTE#	C	VOL	"PILOTE#" IS NOT NULL	"PILOTE#" IS NOT NULL	(null)
2 MEYER1M2021	NL_VOL_AVION#	C	VOL	"AVION#" IS NOT NULL	"AVION#" IS NOT NULL	(null)
3 MEYER1M2021	NL_VOL_HD	C	VOL	"HD" IS NOT NULL	"HD" IS NOT NULL	(null)
4 MEYER1M2021	NL_VOL_HA	C	VOL	"HA" IS NOT NULL	"HA" IS NOT NULL	(null)
5 MEYER1M2021	VOL_CHK_HA_HD	C	VOL	ha>hd	ha>hd	(null)
6 MEYER1M2021	PK_VOL	P	VOL	(null)	(null)	(null)
7 MEYER1M2021	VOL_FK_PILOTE	R	VOL	(null)	(null)	MEYER1M2021
8 MEYER1M2021	FK_VOL_AVION#	R	VOL	(null)	(null)	MEYER1M2021

- **Contraintes** (Dans la table créée) :

```
SELECT * FROM META_TABLES
INNER JOIN META_CONSTRAINTS ON META_TABLES.idTable =
META_CONSTRAINTS.idTable
AND META_TABLES.nomTable = 'VOL' ;
```

- **Indexes** (Dans les DBA_*) :

```
SELECT * FROM DBA_INDEXES
WHERE OWNER = 'MEYER1M2021'
AND TABLE_NAME = 'VOL' ;
```

OWNER	INDEX_NAME	INDEX_TYPE	TABLE_OWNER	TABLE_NAME	TABLE_TYPE	UNIQUENESS	COMPRESSION	PREFIX_LENGTH
1 MEYER1M2021	PK_VOL	NORMAL	MEYER1M2021	VOL	TABLE	UNIQUE	DISABLED	(null)

- **Indexes** (Dans la table créée) :

```
SELECT * FROM META_TABLES
INNER JOIN META_INDEXES ON META_TABLES.idTable =
META_INDEXES.idTable
AND META_TABLES.nomTable = 'VOL' ;
```

3. Liste des utilisateurs de la base de données

- **Tous les utilisateurs** :

```
SELECT * FROM DBA_USERS
```

	USERNAME	USER_ID	PASSWORD	ACCOUNT_STATUS	LOCK_DATE	EXPIRY_DATE	DEFAULT_TABLESPACE
100	KHALIFA1M2021	217	(null)	OPEN	(null)	09/10/21	USERS
101	MAUGARD1M2021	218	(null)	OPEN	(null)	09/10/21	USERS
102	MEJRI1M2021	219	(null)	OPEN	(null)	09/10/21	USERS
103	MERCURI1M2021	220	(null)	OPEN	(null)	09/10/21	USERS
104	MEYER1M2021	221	(null)	OPEN	(null)	09/10/21	USERS
105	SANTAMARIA1M2021	222	(null)	OPEN	(null)	09/10/21	USERS
106	TCHERGUIZOVA1M2021	223	(null)	OPEN	(null)	09/10/21	USERS
107	TENEMAFFO1M2021	224	(null)	OPEN	(null)	09/10/21	USERS
108	BAUMANN1M2021	225	(null)	OPEN	(null)	09/10/21	USERS
109	BELKARFA1M2021	226	(null)	OPEN	(null)	09/10/21	USERS
110	BENABDELJELIL1M2021	227	(null)	OPEN	(null)	09/10/21	USERS
111	BENABDELKRIM1M2021	228	(null)	OPEN	(null)	09/10/21	USERS
112	BEYOSMAN1M2021	229	(null)	OPEN	(null)	09/10/21	USERS
113	BONTEMPS1M2021	230	(null)	OPEN	(null)	09/10/21	USERS

- Un utilisateur :

```
SELECT * FROM DBA_USERS
WHERE USERNAME = 'MEYER1M2021'
```

	USERNAME	USER_ID	PASSWORD	ACCOUNT_STATUS	LOCK_DATE	EXPIRY_DATE	DEFAULT_TABLESPACE
1	MEYER1M2021	221	(null)	OPEN	(null)	09/10/21	USERS

III. RECHERCHE DES TYPES DE FICHIERS

A. MYSQL

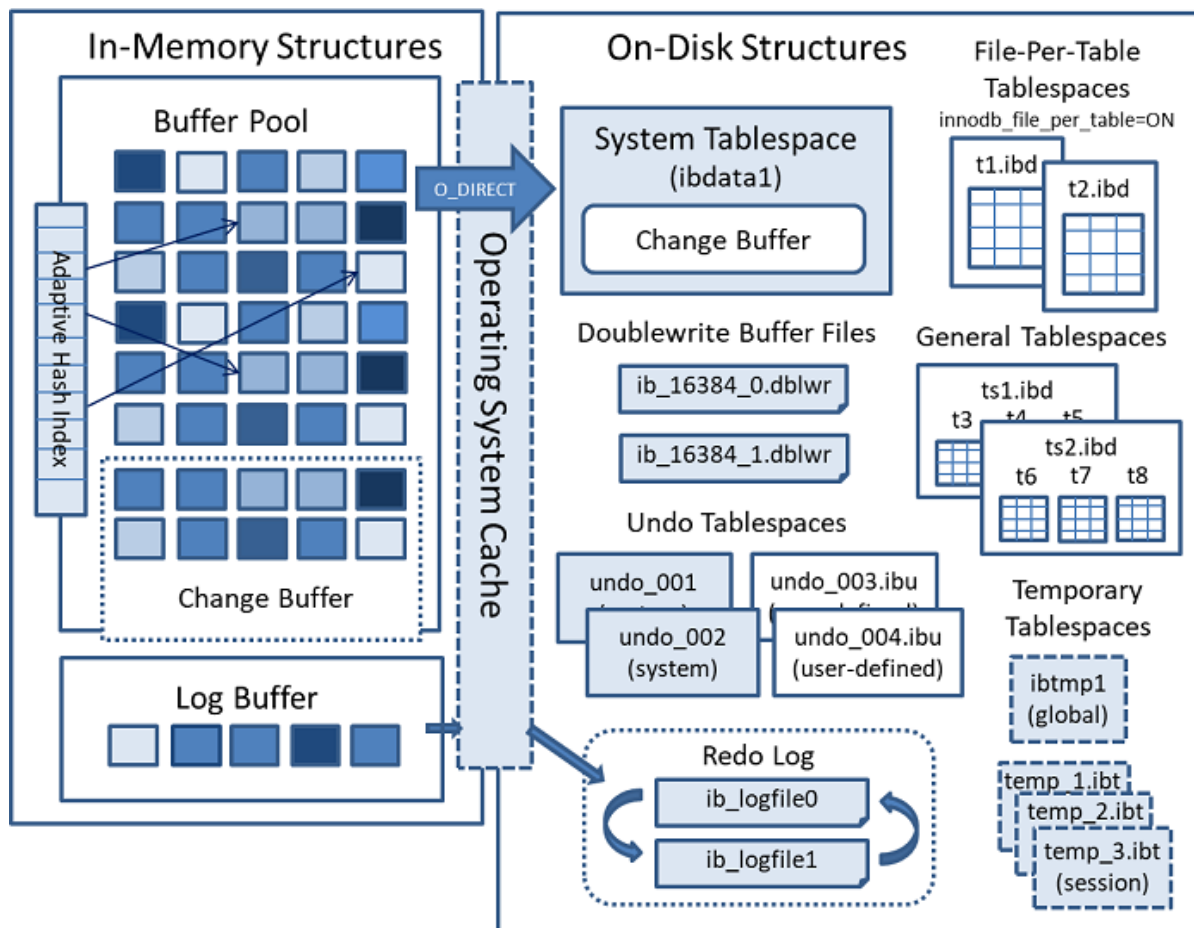
1. Identification et description de chaque type de fichier

Sur MySQL, il existe tout un tas d'extension de fichier afin d'aider au fonctionnement du SGBD. Voici la liste non-exhaustive des types de fichiers présents dans l'infrastructure MySQL :

- ***.frm** : Il s'agit de l'extension de fichier qui contient le schéma ou la définition de la table.
- ***.myd** : Il s'agit de l'extension du fichier qui contient les données de la table MyISAM.
- ***.myi** : Il s'agit de l'extension du fichier qui contient les index de la table MyISAM.
- **my.cnf** : C'est un fichier de configuration de base de données MySQL. Il s'agit du fichier de configuration principal du serveur MySQL.
- **db.opt** : Chaque fois qu'une base de données est créée ou modifiée à l'aide des commandes MySQL, les caractéristiques de la base de données sont stockées dans un fichier texte, "db.opt".
- ***.ibd** : Ce sont les fichiers qui stockent les données et l'index des tables MySQL InnoDB. Ce type de fichier est créé ou utilisé par le logiciel MySQL InnoDB et lui est associé.

- ***.sock** : Toutes les connexions à la base de données MySQL sont gérées par un fichier spécial appelé fichier socket. Ce fichier de socket, à savoir mysqld.sock, est créé automatiquement par le service MySQL, ce qui facilite la communication entre les différents processus.
- **.pid** : L'ID de processus du serveur MySQL est écrit dans ce fichier. La valeur par défaut sera le nom d'hôte du serveur MySQL.
- ***.db** : Ce sont les fichiers avec les extensions «.db» qui stockent les données et les index du moteur de stockage BerkeleyDB.
- ***.log** : Il existe plusieurs fichiers de type .log ; A savoir les erreurs systèmes, les erreurs des requêtes, des erreurs binaires qui sont relatives à la création de données ou de table orchestré par MySQL et les “slow query” qui sont présentes pour améliorer les performances.
- ***.index** : Pour surveiller les fichiers journaux binaires utilisés, un fichier d'index binaire est créé, contenant les noms de tous les fichiers journaux binaires.
- **.TMD** : Il s'agit du fichier de base de données intermédiaire créé par le serveur MySQL créé lors des opérations de réparation. Ce fichier contient des informations sur les récupérations de bases de données.
- ***.TRG et *.TRN** : Les fichiers TRG sont des fichiers de paramétrage qui sont déclenchés et les fichiers TRN sont des fichiers d'espace de noms également déclenchés. Dans le serveur MySQL, chaque fois que des triggers (ou “déclencheur” en français) sont définis, les définitions sont stockées dans ces fichiers.
- ***.ARZ, *.ARM et .ARN** : Les fichiers de données de table et de métadonnées de table ont respectivement les extensions .ARZ et .ARM. Le fichier .ARN est le fichier d'optimisation pendant le processus d'optimisation.

2. Description et schémas.



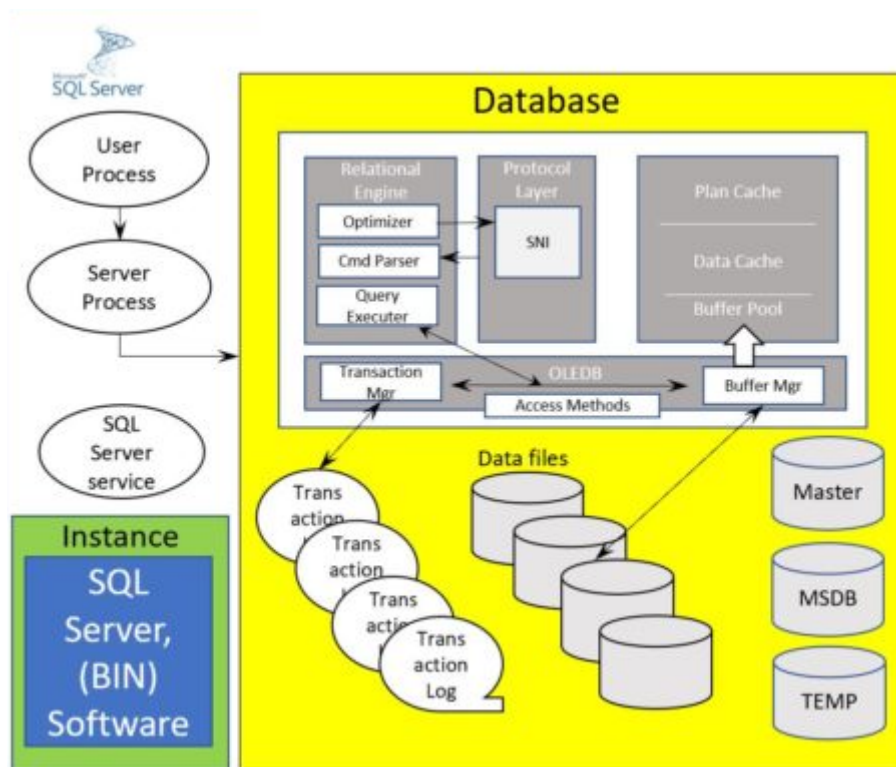
B. SQL SERVER

1. Identification et description de chaque type de fichier

En fonction des exigences et de l'utilisation, une base de données Microsoft SQL Server est en outre classée en trois types de fichiers de données. Il est possible de changer l'extension mais il existe des extensions "par défaut" :

- ***.mdf** : Chaque base de données n'a qu'un seul fichier de données principal à partir duquel tous les autres fichiers de la base de données démarrent. Les emplacements de tous les fichiers de données de la base de données sont enregistrés non seulement dans la base de données principale, mais également dans le fichier de données principal.
- ***.ndf** : Ce type de fichier de données inclut tous les fichiers de données autres que les fichiers de données primaires (donc les *.mdf). Une base de données peut avoir des *.mdf et des *.ndf, plusieurs fichiers de données secondaires ou aucun d'entre eux.
- ***.ldf** : Ce type de fichier est crucial pour le processus de restauration de la base de données. Toutes les informations du journal sont stockées dans le fichier journal. Chaque base de données doit avoir au moins un fichier journal.

2. Description et schémas.



C. ORACLE - REQUÊTES.

1. Fichier de contrôle

- Contenu du fichier de contrôle :

```
SELECT * FROM V$DATABASE
```

	DBID	NAME	CREATED	RESETLOGS_...	RESETLOGS_TIME	PRIOR_R...	PRIOR_RESETLOGS_TIME	LOG_MODE
1	1783921266	BDMBDS	18/07/18	2282175	23/01/20	1477662	18/07/18	NOARCHIVELO

- Nom des fichiers de contrôle :

```
SELECT * FROM V$CONTROLFILE
```

	STATUS	NAME	IS_RECOVERY_DEST_FILE	BLOCK_SIZE	FILE_SIZE_BKLS	CON_ID
1	(null)	/u02/app/oracle/oradata/BDMBDS/control01.ctl	NO	16384	2174	0
2	(null)	/u03/app/oracle/fast_recovery_area/BDMBDS/control02.ctl	NO	16384	2174	0

- Paramètre d'initialisation :

```
SELECT * FROM V$PARAMETER
```

NUM	NAME	TYPE	VALUE	DISPLAY_VA...	DEFAULT_VALUE	ISDEFAULT	ISSES_MODIFIABLE
1	52 lock_name_space	2	(null)	(null)	NULL	TRUE	FALSE
2	53 processes	3	300	300	0	FALSE	FALSE
3	54 sessions	3	472	472	4294967295	TRUE	FALSE
4	56 timed_statistics	1	TRUE	TRUE	FALSE	TRUE	TRUE
5	57 timed_os_statistics	3	0	0	0	TRUE	TRUE
6	58 resource_limit	1	TRUE	TRUE	TRUE	TRUE	FALSE
7	59 license_max_sessions	3	0	0	0	TRUE	FALSE
8	60 license_sessions_warning	3	0	0	0	TRUE	FALSE
9	68 long_module_action	1	TRUE	TRUE	TRUE	TRUE	FALSE
10	90 standby_db_preserve_states	2	NONE	NONE	NONE	TRUE	FALSE
11	94 instance_abort_delay_time	3	0	0	0	TRUE	FALSE
12	112 cpu_count	3	2	2	0	TRUE	FALSE
13	118 instance_groups	2	(null)	(null)	NULL	TRUE	FALSE
14	127 event	2	(null)	(null)	NONE	TRUE	FALSE
15	142 sga_max_size	6	2768240640	2640M	0	TRUE	FALSE
16	149 use_large_pages	2	TRUE	TRUE	TRUE	TRUE	FALSE

2. Fichier de données

- Nom des fichiers de données :

```
SELECT * FROM V$DATAFILE
```

FILE#	CREATION_...	CREATION_TIME	TS#	RFILE#	STATUS	ENABLED	CHECKPOINT_CHANGE#	CHECKPOINT_TIME
1	26	32936513 11/07/20	0	1	SYSTEM	READ WRITE	135919752	15/04/21
2	27	32936518 11/07/20	1	4	ONLINE	READ WRITE	135919752	15/04/21
3	28	32936522 11/07/20	2	9	ONLINE	READ WRITE	135919752	15/04/21
4	29	32936527 11/07/20	5	13	ONLINE	READ WRITE	135919752	15/04/21
5	170	74982636 16/11/20	5	170	ONLINE	READ WRITE	135919752	15/04/21
6	171	74982653 16/11/20	1	171	ONLINE	READ WRITE	135919752	15/04/21
7	190	86661340 19/12/20	55	190	ONLINE	READ WRITE	135919752	15/04/21
8	191	86661367 19/12/20	56	191	ONLINE	READ WRITE	135919752	15/04/21
9	192	86661438 19/12/20	57	192	ONLINE	READ WRITE	135919752	15/04/21

3. Fichier Redo Log

- LogFile :

```
SELECT * FROM V$LOGFILE
```

GR...	STATUS	TYPE	MEMBER	IS_R...	CON_ID
1	3 (null)	ONLINE	/u04/app/oracle/redo/redo03.log	NO	0
2	2 (null)	ONLINE	/u04/app/oracle/redo/redo02.log	NO	0
3	1 (null)	ONLINE	/u04/app/oracle/redo/redo01.log	NO	0
4	1 (null)	ONLINE	/u04/app/oracle/redo/redo11.log	NO	0
5	2 (null)	ONLINE	/u04/app/oracle/redo/redo12.log	NO	0
6	3 (null)	ONLINE	/u04/app/oracle/redo/redo13.log	NO	0
7	4 (null)	ONLINE	/u04/app/oracle/redo/redo04.log	NO	0
8	4 (null)	ONLINE	/u04/app/oracle/redo/redo14.log	NO	0

- Log :

```
SELECT * FROM V$LOG
```

	G...	T...	SEQ...	BYTES	BLOCKSIZE	ME...	AR...	STATUS	FIRST_CHANGE#	FIRST_TIME
1	1	1	3989	1073741824	512	2 NO		INACTIVE	135379611	14/04/21
2	2	1	3991	1073741824	512	2 NO		CURRENT	135919752	15/04/21
3	3	1	3988	1073741824	512	2 NO		INACTIVE	135118221	13/04/21
4	4	1	3990	1073741824	512	2 NO		INACTIVE	135600181	14/04/21

IV. RECHERCHE SUR LES CACHES MÉMOIRES

A. MYSQL

1. Identification et description de la gestion de mémoire

MySQL alloue des buffers (ou “tampons” en français) et des caches pour améliorer les performances des opérations de base de données. La configuration par défaut est conçue pour permettre à un serveur MySQL de démarrer sur une machine virtuelle disposant d'environ 512 Mo de RAM.

Il est possible d'améliorer les performances de MySQL en augmentant les valeurs de certaines variables système liées au cache et à la mémoire tampon.

Il est également possible de modifier la configuration par défaut pour exécuter MySQL sur des systèmes avec une mémoire limitée.

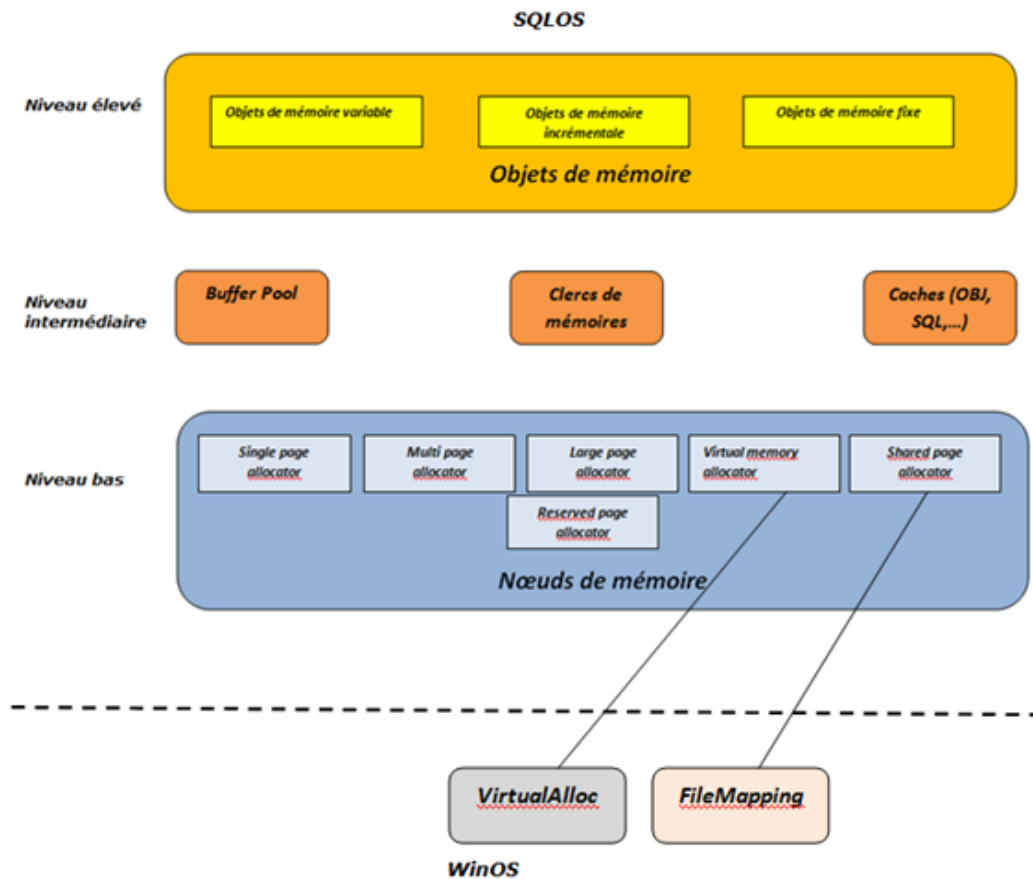
2. Simulation

B. SQL SERVER

1. Identification et description de la gestion de mémoire

SQLOS gère la mémoire de façon hiérarchisée selon trois niveaux. La base de la hiérarchie est constituée des nœuds de mémoire. Le niveau suivant est constitué des clerks de mémoire, des caches mémoire et des pools de mémoires. Le niveau supérieur comprend les objets de mémoire.

Les nœuds de mémoire sont les principaux allocateurs de ressources des composants SQL Server. Ils sont constitués d'allocateurs de pages (single page allocator, multi-page allocator, large page allocator et reserved page allocator[4]), d'un allocateur de mémoire virtuelle (virtual memory allocator, qui utilise l'API Windows VirtualAlloc pour l'allocation de pages de mémoires hors-buffer pool) et d'un allocateur de mémoire partagée (shared memory allocator, qui utilise l'API Windows FileMapping pour l'allocation de pages de mémoire partagées).



2. Simulation

C. ORACLE - ZONES MÉMOIRES.

- SGA :

```
SELECT * FROM V$SGA
```

	NAME	VALUE	CON_ID
1	Fixed Size	8932744	0
2	Variable Size	1795162112	0
3	Database Buffers	889192448	0
4	Redo Buffers	74952704	0

- PGA :

```
SELECT * FROM V$PGASTAT
```


NAME	VALUE	UNIT	CON_ID
1 aggregate PGA target parameter	0	bytes	0
2 aggregate PGA auto target	0	bytes	0
3 global memory bound	183756800	bytes	0
4 total PGA inuse	43560960	bytes	0
5 total PGA allocated	91724800	bytes	0
6 maximum PGA allocated	542478336	bytes	0
7 total freeable PGA memory	34603008	bytes	0
8 MGA allocated (under PGA)	0	bytes	0
9 maximum MGA allocated	0	bytes	0
10 process count	10	(null)	0
11 max processes count	11	(null)	0
12 PGA memory freed back to OS	49932206080	bytes	0
13 total PGA used for auto workareas	0	bytes	0
14 maximum PGA used for auto workareas	51662848	bytes	0
15 total PGA used for manual workareas	0	bytes	0
16 maximum PGA used for manual workareas	1067008	bytes	0
17 over allocation count	0	(null)	0
18 bytes processed	175018428416	bytes	0
19 extra bytes read/written	0	bytes	0
20 cache hit percentage	100	percent	0
21 recompute count (total)	3203305	(null)	0

V. TRANSACTIONS

A. REQUÊTES DE PREUVES DES ANOMALIES

1. Update Loss

TRANSACTION 1	TRANSACTION 2
<pre>SELECT PILOTE.SAL FROM PILOTE WHERE PLNOM = 'Mopololo' -- Resultat : 17000.6 --</pre>	<pre>SELECT PILOTE.SAL FROM PILOTE WHERE PLNOM = 'Mopololo'; -- Resultat : 17000.6 -- UPDATE PILOTE SET SAL = 1500 WHERE PLNOM = 'MOPOLO';</pre>

```

UPDATE PILOTE SET SAL = 1000
WHERE PLNOM = 'MOPOLO';

SELECT PILOTE.SAL FROM PILOTE
WHERE PLNOM = 'Mopolo';
-- Resultat : 1000 --

COMMIT;

```

```

SELECT PILOTE.SAL FROM PILOTE
WHERE PLNOM = 'Mopolo';
-- Resultat : 1500 --

COMMIT;

```

Nous constatons que l'update de la **deuxième transaction** a totalement été supprimée, non prise en compte. La valeur, en transaction finale, est que le nouveau salaire du pilote "Mopolo" est de 1000 et non 1500.

2. Dirty Read

TRANSACTION 1	TRANSACTION 2
<pre> SELECT PILOTE.SAL FROM PILOTE WHERE PLNOM = 'Mopolo' -- Resultat : 17000.6 -- UPDATE PILOTE SET SAL = 1000 WHERE PLNOM = 'MOPOLO'; </pre>	<pre> SELECT PILOTE.SAL FROM PILOTE WHERE PLNOM = 'Mopolo'; -- Resultat : 17000.6 -- UPDATE PILOTE SET SAL = 1500 WHERE PLNOM = 'MOPOLO'; SELECT PILOTE.SAL FROM PILOTE WHERE PLNOM = 'Mopolo'; -- Resultat : 1500 -- COMMIT; </pre>

```
ROLLBACK;

SELECT PILOTE.SAL FROM PILOTE
WHERE PLNOM = 'Mopolo';
-- Resultat : 17000.6 --
```

Nous constatons dans l’affichage de la donnée à la fin de la **première transaction**, le pilote “Mopolo” à la valeur de son salaire à “17000.6” alors qu’elle a été modifiée lors de la **seconde transaction**. Donc la modification de la deuxième transaction n’a jamais été prise en compte.

3. Inconsistency Read

TRANSACTION 1	TRANSACTION 2
<pre>SELECT PILOTE.SAL FROM PILOTE WHERE PLNOM = 'Mopolo' -- Resultat : 17000.6 -- SELECT PILOTE.SAL FROM PILOTE WHERE PLNOM = 'Mopolo'; -- Resultat : 17000.6 -- COMMIT;</pre>	<pre>SELECT PILOTE.SAL FROM PILOTE WHERE PLNOM = 'Mopolo'; -- Resultat : 17000.6 -- UPDATE PILOTE SET SAL = 1500 WHERE PLNOM = 'MOPOLO'; SELECT PILOTE.SAL FROM PILOTE WHERE PLNOM = 'Mopolo'; -- Resultat : 1500 -- COMMIT;</pre>

Nous constatons dans l’affichage de la donnée à la fin de la **première transaction**, le pilote “Mopolo” à la valeur de son salaire à “17000.6” alors qu’elle a été modifiée lors de la **seconde transaction**.

4. Ghost update

TRANSACTION 1	TRANSACTION 2
<pre>SELECT * FROM PILOTE WHERE PLNOM = 'Mopololo' SELECT * FROM PILOTE WHERE PLNOM = 'Mopololo'; INSERT INTO PILOTE VALUES ('22', 'Mopololo', '16-DEC-1945', 'Nice', '987654321', 20000.0); COMMIT;</pre>	<pre>SELECT * FROM PILOTE WHERE PLNOM = 'Mopololo'; INSERT INTO PILOTE VALUES ('21', 'Mopololo', '16-DEC-1945', 'Marseille', '123456789', 16874.0); COMMIT;</pre>

B. PROPOSITION DE SOLUTIONS

Afin de pouvoir sécuriser les transactions et éviter les erreurs de chargements de données et de modifications sur des données erronées, il est possible d'ajouter des verrous sur les colonnes ou les tables.

Ces verrous permettent à une transaction d'arrêter le traitement en cours si celle-ci est en cours de changement selon les différentes couches en cas si le verrouillage se fait à granules variables.

Nous allons donc appliquer un verrou sur la donnée en cours de modification afin que d'autres transactions ne puissent pas effectuer d'opération.

C. DISTRIBUEE

<https://pastebin.com/Xw9tNXze>

VI. BIBLIOGRAPHIE

1. <https://dev.mysql.com/doc/refman/8.0/en/pluggable-storage-overview.html> (partie I, A)
2. <https://secureanycld.com/sql-server-2016-standard-edition-on-cloud-technical-support-cloud-help-azure-aws-opensource-cognosys/> (partie I, B)
3. <https://perso.liris.cnrs.fr/nicolas.lumineau/teaching/M2ASBD/CoursAdmin.pdf> (partie III, C)
4. <https://docs.microsoft.com/fr-fr/sql/relational-databases/performance-monitor/monitor-memory-usage?view=sql-server-ver15> (partie IV, B)
5. <https://mcherif.wordpress.com/2012/12/04/sql-server-un-peu-de-theorie-sur-la-memoire/> (partie IV, B)
6. <https://www.geeksforgeeks.org/mysql-database-files/> (partie III, A)
7. <https://sqlbak.com/academy/database-files> (partie III, B)
8. <https://dbakevlar.com/2019/03/oracle-vs-sql-server-architecture/> (partie III, B - Schéma)
9. <http://man.hubwiz.com/docset/MySQL.docset/Contents/Resources/Documents/innodb-storage-engine.html> (partie III, A - Schéma)