
DB2 SQL

Services

Objet

- **De nombreux services sont fournis au travers de DB2**
 - Pour la plupart en doublon des commandes système
 - Ou d'API
- **Leur nombre étant élevé et leur complexité variable**
 - Il s'agit ici d'établir une liste des fonctionnalités disponibles
- **Pour plus de détails sur une opération précise**
 - DB2 for i Services
https://www-01.ibm.com/support/knowledgecenter/ssw_ibm_i_73/rzajq/rzajqservicesdb2.htm?lang=fr
 - IBM i Services
https://www-01.ibm.com/support/knowledgecenter/ssw_ibm_i_73/rzajq/rzajqservicessys.htm?lang=fr

Synthèse

DB2 for i Services – Application services

QSYS2	DELIMIT_NAME	Fonction
QSYS2	OVERRIDE_TABLE	Procédure
QSYS2	PARSE_STATEMENT	Fonction table
SYSPROC	WLM_SET_CLIENT_INFO	Procédure

DB2 for i Services – Performance services

SYSTOOLS	ACT_ON_INDEX_ADVICE	Procédure
QSYS2	DATABASE_MONITOR_INFO	Vue
SYSTOOLS	HARVEST_INDEX_ADVICE	Procédure
SYSTOOLS	REMOVE_INDEXES	Procédure
QSYS2	RESET_TABLE_INDEX_STATISTICS	Procédure

Synthèse

DB2 for i Services – Utility services

QSYS2	CANCEL_SQL	Procédure
SYSTOOLS	CHECK_SYSCST	Procédure
SYSTOOLS	CHECK_SYSROUTINE	Procédure
QSYS2	DUMP_SQL_CURSORS	Procédure
QSYS2	FIND_AND_CANCEL_QSQSRVR_SQL	Procédure
QSYS2	FIND_QSQSRVR_JOBS	Procédure
QSYS2	GENERATE_SQL	Procédure
QSYS2	RESTART_IDENTITY	Procédure

Synthèse

IBM i Services – Application services

QSYS2	ENVIRONMENT_VARIABLE_INFO	Vue
QSYS2	QCMDEXC	Procédure
QSYS2	SERVICES_INFO	Table
QSYS2	SET_PASE_SHELL_INFO	Procédure

IBM i Services – Java services

QSYS2	JVM_INFO	Vue
QSYS2	SET_JVM	Procédure

IBM i Services – Journal services

QSYS2	DISPLAY_JOURNAL	Fonction table
QSYS2	JOURNAL_INFO	Vue

IBM i Services – Librarian services

QSYS2	LIBRARY_LIST_INFO	Vue
QSYS2	OBJECT_STATISTICS	Fonction table

Synthèse

IBM i Services – Message handling services

QSYS2	HISTORY_LOG_INFO	Fonction table
QSYS2	JOBLOG_INFO	Fonction table
QSYS2	MESSAGE_QUEUE_INFO	Vue
QSYS2	REPLY_LIST_INFO	Vue

IBM i Services – Product services

SYSTOOLS	LICENSE_EXPIRATION_CHECK	Procédure
QSYS2	LICENSE_INFO	Vue

IBM i Services – PTF services

SYSTOOLS	GROUP_PTF_CURRENCY	Vue
SYSTOOLS	GROUP_PTF_DETAILS	Vue
QSYS2	GROUP_PTF_INFO	Vue
QSYS2	PTF_INFO	Vue

Synthèse

IBM i Services – Security services

QSYS2	AUTHORITY_COLLECTION	Vue
QSYS2	AUTHORIZATION_LIST_INFO	Vue
QSYS2	AUTHORIZATION_LIST_USER_INFO	Vue
QSYS2	DRDA_AUTHENTICATION_ENTRY_INFO	Vue
QSYS2	FUNCTION_INFO	Vue
QSYS2	FUNCTION_USAGE	Vue
QSYS2	GROUP_PROFILE_ENTRIES	Vue
QSYS2	OBJECT_PRIVILEGES	Vue
SYSPROC	SET_COLUMN_ATTRIBUTE	Procédure
QSYS2	SQL_CHECK_AUTHORITY	Fonction
QSYS2	USER_INFO	Vue

Synthèse

IBM i Services – spool services

QSYS2	OUTPUT_QUEUE_ENTRIES	Fonction table
QSYS2	OUTPUT_QUEUE_ENTRIES	Vue

IBM i Services – storage services

QSYS2	MEDIA_LIBRARY_INFO	Vue
QSYS2	SYSDISKSTAT	Vue
QSYS2	SYSTMPSTG	Vue
QSYS2	USER_STORAGE	Vue

IBM i Services – System health services

QSYS2	SYSLIMTBL	Table
QSYS2	SYSLIMITS	Vue
SYSIBMADM	QIBM_SYSTEM_LIMITS_*	Variables globales

Synthèse

IBM i Services – TCP/IP services

SYSIBMADM	ENV_SYS_INFO	Vue
QSYS2	NETSTAT_INFO	Vue
QSYS2	NETSTAT_INTERFACE_INFO	Vue
QSYS2	NETSTAT_JOB_INFO	Vue
QSYS2	NETSTAT_ROUTE_INFO	Vue
QSYS2	SET_SERVER_SBS_ROUTING	Procédure
QSYS2	SERVER_SBS_ROUTING	Vue
QSYS2	TCPIP_INFO	Vue

Synthèse

IBM i Services – Work management services

QSYS2	ACTIVE_JOB_INFO	Fonction table
QSYS2	GET_JOB_INFO	Fonction table
QSYS2	JOB_INFO	Fonction table
QSYS2	MEMORY_POOL	Fonction table
QSYS2	MEMORY_POOL_INFO	Vue
QSYS2	OBJECT_LOCK_INFO	Vue
QSYS2	RECORD_LOCK_INFO	Vue
QSYS2	SCHEDULED_JOB_INFO	Vue
QSYS2	SYSTEM_STATUS	Fonction table
QSYS2	SYSTEM_STATUS_INFO	Vue
QSYS2	SYSTEM_VALUE_INFO	Vue

IBM i Services

Application services

ENVIRONMENT_VARIABLE_INFO

➤ Vue QSYS2.ENVIRONMENT_VARIABLE_INFO

- Retourne les informations sur les variables d'environnement
- Exemple

```
SELECT ENVIRONMENT_VARIABLE_NAME,  
       ENVIRONMENT_VARIABLE_VALUE  
FROM QSYS2.ENVIRONMENT_VARIABLE_INFO  
WHERE ENVIRONMENT_VARIABLE_TYPE = 'SYSTEM'
```

ENVIRONMENT_VARIABLE_NAME	ENVIRONMENT_VARIABLE_VALUE
QIBM_MULTI_THREADED	Y
QIBM_SNDSPFLF_OPT	TCP

QCMDEXC

➤ Procédure QSYS2.QCMDEXC

- Syntaxe

```
>>-QCMDEXC--(--CL-command-string--)------><
```

- Exemple

```
CALL QSYS2.QCMDEXC('ADDLIBLE PRODLIB2');
```

SERVICES_INFO

➤ Table QSYS2.SERVICES_INFO

- Liste l'ensemble des services disponibles au travers de DB2
Description, Group PTF nécessaire ...

- Exemple

```
SELECT *  
FROM QSYS2.SERVICES_INFO
```

SERVICE_CATEGORY	SERVICE_SCHEMA_NAME	SERVICE_NAME	SQL_OBJECT_TYPE
PTF	QSYS2	PTF_INFO	VIEW
PTF	QSYS2	GROUP_PTF_INFO	VIEW
PTF	SYSTOOLS	GROUP_PTF_CURRENCY	VIEW
PTF	SYSTOOLS	GROUP_PTF_DETAILS	VIEW
SECURITY	QSYS2	USER_INFO	VIEW
SECURITY	QSYS2	FUNCTION_INFO	VIEW
SECURITY	QSYS2	FUNCTION_USAGE	VIEW

Java services

JVM_INFO

➤ Vue **QSYS2.JVM_INFO**

- Retourne les informations sur les travaux avec une JVM active
- Exemple

```
SELECT *  
FROM QSYS2.JVM_INFO  
ORDER BY CURRENT_HEAP_SIZE DESC
```

SET_JVM

➤ Procédure QSYS2.SET_JVM

- Permet d'effectuer une action sur un travaux JVM actif

- Syntaxe

```
>>-SET_JVM--(--job_name--,--action--)-----><
```

- Actions possibles

GC_ENABLE_VERBOSE

GC_DISABLE_VERBOSE

GENERATE_HEAP_DUMP

GENERATE_SYSTEM_DUMP

GENERATE_JAVA_DUMP

- Exemple

```
CALL QSYS2.SET_JVM('121376/QWEBADMIN/ADMIN4',  
                  'GC_ENABLE_VERBOSE') ;
```

Journal services

DISPLAY_JOURNAL

➤ UDF Table QSYS2.DISPLAY_JOURNAL

- Retourne les entrées de journal

```
>>-DISPLAY_JOURNAL--(----->
>--+-----+--journal_library--->
    '-JOURNAL_LIBRARY-- => -'
>--,--+-----+--journal_name----->
    '-JOURNAL_NAME-- => -'
>---+---paramètres optionnels-----><
```

- Exemples

```
SELECT *
FROM TABLE ( DISPLAY_JOURNAL( 'TESTLIB', 'QSQJRN')) AS JT;

SELECT journal_code, journal_entry_type,
       object, object_type, X.*
FROM TABLE ( QSYS2.Display_Journal(
    'PRODDATA', 'QSQJRN', -- Journal library and name
    OBJECT_LIBRARY=>'PRODDATA', OBJECT_NAME=>'SALES',
    OBJECT_OBJTYPE=>'*FILE', OBJECT_MEMBER=>'SALES' ) ) AS X
WHERE journal_entry_type in ('DL', 'PT', 'PX', 'UP') AND
      "CURRENT_USER" = 'SUPERUSER'
ORDER BY entry_timestamp DESC
```

JOURNAL_INFO

➤ Vue QSYS2.JOURNAL_INFO

- Affiche les attributs du journal
Cf API Retrieve Journal Information

- Exemples

```
SELECT JOURNAL_NAME, JOURNAL_LIBRARY
       MAXIMUM_REMOTE_JOURNALS_ENTRIES_BEHIND,
       MAXIMUM_REMOTE_JOURNALS_TIME_BEHIND,
       MAXIMUM_REMOTE_JOURNALS_RETRANSMISSIONS
FROM QSYS2.JOURNAL_INFO,
WHERE MAXIMUM_REMOTE_JOURNALS_ENTRIES_BEHIND > 0
ORDER BY MAXIMUM_REMOTE_JOURNALS_ENTRIES_BEHIND DESC
```

```
SELECT *
FROM QSYS2.JOURNAL_INFO
WHERE JOURNAL_TYPE = '*REMOTE' AND
      JOURNAL_STATE <> '*ACTIVE'
ORDER BY JOURNAL_LIBRARY, JOURNAL_NAME
```

Librarian services

LIBRARY_LIST_INFO

- **Vue QSYS2.LIBRARY_LIST_INFO**
 - Retourne la liste de bibliothèques en cours
 - Exemple

```
SELECT * FROM QSYS2.LIBRARY_LIST_INFO
```

Message handling services

HISTORY_LOG_INFO

➤ UDF Table QSYS2.HISTORY_LOG_INFO

- Retourne une ligne par message dans l'intervalle défini par START_TIME et END_TIME
- Identique à DSPLOG ou API QMHOLHST
- Syntaxe (paramètres optionnels)

```
>>-HISTORY_LOG_INFO--(-'-START_TIME-- => -'  
                        '-END_TIME-- => -'-)
```

START_TIME et END_TIME sont des horodatages

Valeurs par défaut :

```
START_TIME : CURRENT DATE - 1 DAY
```

```
END_TIME   : CURRENT TIMESTAMP
```

- Exemples

```
SELECT * FROM TABLE(QSYS2.HISTORY_LOG_INFO()) X;
```

```
SELECT * FROM TABLE(QSYS2.HISTORY_LOG_INFO(CURRENT TIMESTAMP  
- 1 DAY)) X;
```

```
SELECT * FROM TABLE(QSYS2.HISTORY_LOG_INFO(LAST_IPL_TIME,  
CURRENT TIMESTAMP)) A;
```

JOBLOG_INFO

➤ UDF Table QSYS2.JOBLOG_INFO

- Retourne les messages de la joblog d'un travail actif

```
>>-JOBLOG_INFO--(--job-name--)-----><
```

- Exemples

```
SELECT *
```

```
FROM TABLE(QSYS2.JOBLOG_INFO('347117/Quser/Qzdasoinit')) A
```

```
SELECT MESSAGE_TEXT
```

```
FROM TABLE(QSYS2.JOBLOG_INFO('817029/QUSER/QPADEV0004')) A
```

```
WHERE A.MESSAGE_TYPE = 'REQUEST'
```

```
ORDER BY ORDINAL_POSITION DESC
```

```
FETCH FIRST 1 ROW ONLY
```

MESSAGE_QUEUE_INFO

➤ Vue QSYS2.MESSAGE_QUEUE_INFO

- Contient une ligne par file de message
- Equivalent à la commande DSPMSG ou à l'API QMHRCVM
- Nécessite les droits

*USE sur la *MSGQ

*EXECUTE sur *LIB

- Exemple

```
SELECT * FROM QSYS2.MESSAGE_QUEUE_INFO;
```

```
SELECT message_timestamp, message_text, from_user  
FROM qsys2.message_queue_info  
WHERE message_type = 'INQUIRY'  
ORDER BY message_timestamp DESC;
```

REPLY_LIST_INFO

➤ Vue **QSYS2.REPLY_LIST_INFO**

- Contient les informations de la liste de réponse pour le travail en cours

- Exemple

```
SELECT * FROM QSYS2.REPLY_LIST_INFO
```

Product services

LICENSE_EXPIRATION_CHECK

➤ Procédure **SYSTOOLS.LICENSE_EXPIRATION_CHECK**

- Envoie un message dans *SYSOPR pour chaque licence de produit installé qui arrivera à expiration dans le nombre de jours indiqués

- Syntaxe

```
>>-LICENSE_EXPIRATION_CHECK--(----->
>--+-----+--)-><
  '-+-----+--expiration-interval-'
    '-EXPIRATION_INTERVAL-- => -'
```

- Avec

expiration-interval

Nombre de jours

Défaut : 30 jours

- Exemple

```
CALL SYSTOOLS.LICENSE_EXPIRATION_CHECK(10);
```


LICENSE_INFO

➤ Vue **QSYS2.LICENSE_INFO**

- Contient les informations de licences pour tous les produits et dispositifs

Cf WRKLICINF

- Exemple

```
SELECT *  
FROM QSYS2.LICENSE_INFO  
WHERE LICENSE_EXPIRATION <= CURRENT DATE + 14 DAYS ;
```

PTF services

GROUP_PTF_CURRENCY

➤ Vue **SYSTOOLS.GROUP_PTF_CURRENCY**

- Permet de comparer le niveau des GROUP PTF installés avec ceux disponibles chez IBM

- Exemple

```
SELECT *  
FROM SYSTOOLS.GROUP_PTF_CURRENCY  
ORDER BY ptf_group_level_available -  
         ptf_group_level_installed DESC
```

GROUP_PTF_DETAILS

➤ Vue **SYSTOOLS.GROUP_PTF_DETAILS**

- Idem **GROUP_PTF_CURRENCY** mais plus détaillée
- Exemple

```
SELECT *  
FROM SYSTOOLS.GROUP_PTF_DETAILS  
WHERE PTF_STATUS <> 'PTF APPLIED'  
ORDER BY PTF_GROUP_NAME
```

GROUP_PTF_INFO

➤ Vue **QSYS2.GROUP_PTF_INFO**

- Fournit des informations sur les GROUP PTF du serveur

- Exemple

```
SELECT MAX(PTF_GROUP_LEVEL) AS CUM_LEVEL  
FROM QSYS2.GROUP_PTF_INFO  
WHERE PTF_GROUP_NAME IN ('SF99610', 'SF99710') AND  
       PTF_GROUP_STATUS = 'INSTALLED'
```

PTF_INFO

➤ Vue **QSYS2.PTF_INFO**

- Fournit des informations sur les PTF du serveur

- Exemples

```
SELECT PTF_IDENTIFIER, PTF_IPL_ACTION, A.*  
FROM QSYS2.PTF_INFO A  
WHERE PTF_IPL_ACTION <> 'NONE'
```

```
SELECT PTF_IDENTIFIER, PTF_PRODUCT_DESCRIPTION, A.*  
FROM QSYS2.PTF_INFO A  
WHERE PTF_LOADED_STATUS = 'LOADED'  
ORDER BY PTF_PRODUCT_ID
```

Security services

AUTHORITY_COLLECTION

➤ Vue QSYS2.AUTHORITY_COLLECTION

- Retourne les informations de la collecte de droits

- Exemple

```
SELECT AUTHORIZATION_NAME, AUTHORITY_COLLECTION_ACTIVE  
FROM QSYS2.USER_INFO  
WHERE AUTHORITY_COLLECTION_REPOSITORY_EXISTS='YES';
```

```
SELECT *  
FROM QSYS2.AUTHORITY_COLLECTION  
WHERE USER_NAME = 'USER1';
```


AUTHORIZATION_LIST_INFO

➤ Vue **QSYS2.AUTHORIZATION_LIST_INFO**

- Retourne la liste de tous les objets protégés par une liste d'autorisations

Objets de QSYS et fichiers de l'IFS

- Nécessite

*READ sur la liste, ou *ALLOBJ, QIBM_DB_SECADM via CHGFCNUSG

- Exemple

```
SELECT *  
FROM QSYS2.AUTHORIZATION_LIST_INFO  
WHERE AUTHORIZATION_LIST = 'APP1';
```

AUTHORIZATION_LIST_USER_INFO

- **Vue QSYS2.AUTHORIZATION_LIST_USER_INFO**
 - Retourne la liste des listes d'autorisations ainsi que des autorités contenues (profils et droits associés dans la liste)
 - Equivalent à DSPAUTL
 - Exemple

```
SELECT *
FROM QSYS2.AUTHORIZATION_LIST_USER_INFO
WHERE AUTHORIZATION_NAME = '*PUBLIC' ;
```

DRDA_AUTHENTICATION_ENTRY_INFO

➤ Vue **QSYS2.DRDA_AUTHENTICATION_ENTRY_INFO**

- Retourne les authentifications utilisateurs pour DRDA
Celles ajoutées par **ADDSVRAUTE**

- Exemple

```
SELECT DISTINCT(AUTHORIZATION_NAME)  
FROM QSYS2.DRDA_AUTHENTICATION_ENTRY_INFO
```

FUNCTION_INFO

- **Vue QSYS2.FUNCTION_INFO**
 - Retourne les informations des fonctions
Celles de WRKFCNUSG
 - Exemple

```
SELECT * FROM
QSYS2.FUNCTION_INFO
ORDER BY FUNCTION_ID
```

FUNCTION_USAGE

➤ Vue **QSYS2.FUNCTION_USAGE**

- Retourne la configuration détaillée des fonctions
Celles de WRKFCNUSG

- Exemple

```
SELECT *  
FROM QSYS2.FUNCTION_USAGE  
ORDER BY FUNCTION_ID, USER_NAME
```

GROUP_PROFILE_ENTRIES

➤ Vue **QSYS2.GROUP_PROFILE_ENTRIES**

- Retourne la composition des groupes d'utilisateurs

- Exemple

```
SELECT *  
FROM GROUP_PROFILE_ENTRIES
```

OBJECT_PRIVILEGES

➤ Vue **QSYS2.OBJECT_PRIVILEGES**

- Retourne la liste de chaque privilège utilisateur pour chaque objet du système
- Equivalent à la commande DSPOBJAUT
- Exemple

```
SELECT *  
FROM QSYS2.OBJECT_PRIVILEGES  
WHERE SYSTEM_OBJECT_SCHEMA = 'QSYS' AND  
      OBJECT_TYPE = '*USRPRF' AND  
      AUTHORIZATION_NAME = '*PUBLIC' AND  
      OBJECT_AUTHORITY <> '*EXCLUDE';
```

SET_COLUMN_ATTRIBUTE

➤ procédure **SYSPROC.SET_COLUMN_ATTRIBUTE**

- Permet de positionner l'attribut SECURE sur une colonne, de sorte que sa valeur n'apparaisse pas dans les moniteur DB ou le cache des plans SQL

```
>>-SET_COLUMN_ATTRIBUTE--(--schema-name-- , --table-name-- , -  
-column-name-- , --attribute--)-><
```

Avec attribute SECURE NO ou YES

- Exemple

```
CALL SYSPROC.SET_COLUMN_ATTRIBUTE(  
    'LIB1', 'ORDERS', 'CCNBR', 'SECURE YES' ) ;
```


SQL_CHECK_AUTHORITY

➤ UDF QSYS2.SQL_CHECK_AUTHORITY

- Permet de tester que l'utilisateur en cours est autorisé ou non à un fichier

```
>>-SQL_CHECK_AUTHORITY--(--library-name-- , --file-name--)><
```

- Valeur de retour
 - 0 l'utilisateur n'est pas autorisé
 - 1 l'utilisateur est autorisé

USER_INFO

➤ Vue **QSYS2.USER_INFO**

- Contient les informations des profils utilisateurs
- Exemple

```
SELECT *  
FROM QSYS2.USER_INFO  
WHERE SIGN_ON_ATTEMPTS_NOT_VALID > 0
```

Spool services

OUTPUT_QUEUE_ENTRIES

➤ UDF table **QSYS2.OUTPUT_QUEUE_ENTRIES**

- Retourne une ligne par entrée de file d'attente en sortie
- Exemple

```
SELECT *  
FROM TABLE(QSYS2.OUTPUT_QUEUE_ENTRIES(  
    '*LIBL', 'QEZJOBLOG', '*NO')) A  
ORDER BY SIZE DESC  
FETCH FIRST 100 ROWS ONLY
```

OUTPUT_QUEUE_ENTRIES

➤ Vue QSYS2.OUTPUT_QUEUE_ENTRIES

- Retourne une ligne par entrée de file d'attente en sortie
- Exemple

```
SELECT *  
FROM QSYS2.OUTPUT_QUEUE_ENTRIES  
WHERE OUTPUT_QUEUE_NAME = 'QEZJOBLOG'  
ORDER BY SIZE DESC  
FETCH FIRST 100 ROWS ONLY
```

Storage services

MEDIA_LIBRARY_INFO

➤ Vue **QSYS2.MEDIA_LIBRARY_INFO**

- Retourne les informations de la commande **WRKMLBSTS** (**Work with Media Library Status**)

- Exemple

```
SELECT * FROM QSYS2.MEDIA_LIBRARY_INFO
```

SYSDISKSTAT

➤ Vue **QSYS2.SYSDISKSTAT**

- Retourne les informations sur les disques
- Exemple

```
SELECT *  
FROM QSYS2.SYSDISKSTAT
```

Pour les disques SSD :

```
SELECT *  
FROM QSYS2.SYSDISKSTAT  
WHERE UNIT_TYPE = 1
```


SYSTMPSTG

➤ Vue **QSYS2.SYSTMPSTG**

- Retourne les informations sur le stockage temporaire, par pool et par travail

- Exemple

```
SELECT *  
FROM QSYS2.SYSTMPSTG
```

USER_STORAGE

➤ Vue **QSYS2.USER_STORAGE**

- Retourne les informations sur le stockage utilisé par les objets dont le propriétaire est le profil utilisateur indiqué

- Exemple

```
SELECT *  
FROM QSYS2.USER_STORAGE  
WHERE USER_NAME = 'NB'
```

System health services

SYSLIMTBL

➤ Table QSYS2.SYSLIMTBL

- Contient les informations sur les limites systèmes qui sont approchées

LIMIT_CATEGORTY :	0	Database
	1	Journal
	2	Security
	3	Miscellaneous
	4	Work management
	5	File system
	6	Save/restore
	7	Cluster
	8	Communications
LIMIT_TYPE :	1	Object
	2	Job
	3	System
	4	ASP

- Exemple

```
SELECT * FROM QSYS2.SYSLIMTBL
```

SYSLIMITS

➤ Vue **QSYS2.SYSLIMITS**

- Contient les informations sur les limites systèmes qui sont approchées

Elle est construite sur **QSYS2.SYSLMTBL**

Contient des informations sur les travaux si encore actifs

- Exemple

```
SELECT *
```

```
FROM QSYS2.SYSLIMITS
```

QIBM_SYSTEM_LIMITS_*

➤ Variables globales **SYSIBMADM.QIBM_SYSTEM_LIMITS_***

- Permet de limiter le nombre de lignes dans **QSYS2.SYSLIMTBL**, par type de limite

- Nom des variables possibles et valeurs par défaut

QIBM_SYSTEM_LIMITS_PRUNE_BY_ASP : 100

QIBM_SYSTEM_LIMITS_PRUNE_BY_JOB : 50

QIBM_SYSTEM_LIMITS_PRUNE_BY_OBJECT : 20

QIBM_SYSTEM_LIMITS_PRUNE_BY_SYSTEM : 100

QIBM_SYSTEM_LIMITS_SAVE_HIGH_POINTS_BY_ASP : 25

QIBM_SYSTEM_LIMITS_SAVE_HIGH_POINTS_BY_JOB : 5

QIBM_SYSTEM_LIMITS_SAVE_HIGH_POINTS_BY_OBJECT : 5

QIBM_SYSTEM_LIMITS_SAVE_HIGH_POINTS_BY_SYSTEM : 25

- Exemples

```
CREATE OR REPLACE VARIABLE  
SYSIBMADM.QIBM_SYSTEM_LIMITS_PRUNE_BY_SYSTEM INTEGER DEFAULT  
50 ;
```

```
VALUES SYSIBMADM.QIBM_SYSTEM_LIMITS_PRUNE_BY_SYSTEM ;
```

TCP/IP services

ENV_SYS_INFO

- **Vue `SYSIBMADM.ENV_SYS_INFO`**
 - Retourne des informations du serveur
 - Version
 - Nom
 - Nombre de CPU et mémoire
 - Exemple

```
SELECT *  
FROM SYSIBMADM.ENV_SYS_INFO
```


NETSTAT_INFO

- **Vue QSYS2.NETSTAT_INFO**
 - Retourne les informations de connexion IPv4 et IPv6
Une ligne par connexion
 - Exemple

```
SELECT *  
FROM QSYS2.NETSTAT_INFO
```

NETSTAT_INTERFACE_INFO

➤ Vue QSYS2.NETSTAT_INTERFACE_INFO

- Retourne les informations de connexion IPv4 et IPv6 par interfaces

- Exemple

```
SELECT *  
FROM QSYS2.NETSTAT_INTERFACE_INFO
```

NETSTAT_JOB_INFO

➤ Vue **QSYS2.NETSTAT_JOB_INFO**

- Retourne les informations des travaux qui utilisent des connexions IPv4 et IPv6 par interfaces

- Exemple

```
SELECT *  
FROM QSYS2.NETSTAT_JOB_INFO  
WHERE CONNECTION_TYPE = 'IPV4'
```

NETSTAT_ROUTE_INFO

➤ Vue `QSYS2.NETSTAT_ROUTE_INFO`

- Retourne les informations sur les routes IPv4 et IPv6
- Exemple

```
SELECT *  
FROM QSYS2.NETSTAT_ROUTE_INFO
```

SET_SERVER_SBS_ROUTING

➤ Procédure QSYS2.SET_SERVER_SBS_ROUTING

- Permet de configurer certains serveurs pour utiliser des sous-systèmes alternatifs, en fonction du profil ou profil de groupe

- Syntaxe

```
>>-SET_SERVER_SBS_ROUTING--(----->
>--+-----+---authorization_name----->
    '-AUTHORIZATION_NAME-- => -'
>--,--+-----+---server_name----->
    '-SERVER_NAME-- => -'
>--,--+-----+---subsystem_name--)------<
    '-SUBSYSTEM_NAME-- => -'
```

- Exemple

```
CALL QSYS2.SET_SERVER_SBS_ROUTING(
    'SLFUSER', 'QRWTSRVR', 'ADHOCSBS')
```

SERVER_SBS_ROUTING

➤ Vue **QSYS2.SERVER_SBS_ROUTING**

- Retourne les informations de routage des sous-systèmes serveurs

- Exemple

```
SELECT *  
FROM QSYS2.SERVER_SBS_ROUTING
```

TCPIP_INFO

➤ Vue **QSYS2.TCPIP_INFO**

- Retourne les informations TCP/IP de la connexion courante

- Exemple

```
SELECT *  
FROM QSYS2.TCPIP_INFO
```

Work management services

ACTIVE_JOB_INFO

➤ UDF table QSYS2.ACTIVE_JOB_INFO

- Retourne une ligne par travail actif
- Syntaxe

```
>>-ACTIVE_JOB_INFO--(----->
                        '-RESET_STATISTICS-- => '
                        'SUBSYSTEM_LIST_FILTER-- => '
                        'JOB_NAME_FILTER-- => '
                        'CURRENT_USER_LIST_FILTER-- => '
>---)-----<<
```

- Exemple : travaux QZDASOINIT qui consomment le plus d'IO

```
SELECT JOB_NAME, AUTHORIZATION_NAME,
       ELAPSED_TOTAL_DISK_IO_COUNT, ELAPSED_CPU_PERCENTAGE
FROM TABLE(QSYS2.ACTIVE_JOB_INFO( JOB_NAME_FILTER =>
'QZDASOINIT',
                                SUBSYSTEM_LIST_FILTER => 'QUSRWRK')) X
ORDER BY ELAPSED_TOTAL_DISK_IO_COUNT DESC
FETCH FIRST 10 ROWS ONLY;
```

GET_JOB_INFO

➤ UDF table QSYS2.GET_JOB_INFO

- Retourne une ligne contenant les informations d'un travail spécifié
- Syntaxe

```
>>-GET_JOB_INFO--(--job-name--)-----><
```

- Exemple

```
SELECT *  
FROM TABLE(QSYS2.GET_JOB_INFO(  
            '347117/Quser/Qzdasoinit')) A
```

JOB_INFO

➤ UDF table QSYS2.JOB_INFO

- Nécessite 7.3 TR1 ou 7.2 TR5
- Retourne une ligne par travail correspondant aux critères fournis,
Équivalent aux commandes : WRKUSRJOB, WRKSBSJOB, WRKSBMJOB ou API QUSLJOB
- Syntaxe (tous paramètres facultatifs)

```
>>-JOB_INFO--(- ' -JOB_STATUS_FILTER-- => -'  
              '-JOB_TYPE_FILTER-- => - '  
              '-JOB_SUBSYSTEM_FILTER-- => - '  
              '-JOB_USER_FILTER-- => - '  
              '-JOB_SUBMITTER_FILTER-- => -')--><
```

- Exemples

Tous les jobs interactifs

```
SELECT * FROM TABLE(QSYS2.JOB_INFO(JOB_TYPE_FILTER =>  
'*INTERACT')) X;
```

Travaux que j'ai soumis dans QBATCH

```
SELECT * FROM  
TABLE(QSYS2.JOB_INFO(JOB_SUBMITTING_SOURCE_FILTER => '*JOB',  
JOB_SUBSYSTEM_FILTER => 'QBATCH')) X;
```

MEMORY_POOL

➤ UDF table QSYS2.MEMORY_POOL

- Retourne une ligne pour chaque pool mémoire
- Syntaxe

```
>>-MEMORY_POOL--(----->
>--+-----+--)------<
'--+-----+--reset_statistics-'
'-RESET_STATISTICS-- => -'
```

- Exemple

```
SELECT *
FROM TABLE(QSYS2.MEMORY_POOL(RESET_STATISTICS=>'YES')) X;
```

MEMORY_POOL_INFO

➤ Vue `QSYS2.MEMORY_POOL_INFO`

- Retourne une ligne par pool actif

Informations de la commande `WRKSYSSTS` (Work System Status)

- Exemple

```
SELECT *  
FROM QSYS2.MEMORY_POOL_INFO;
```

OBJECT_LOCK_INFO

➤ Vue **QSYS2.OBJECT_LOCK_INFO**

- Retourne une ligne par verrou

- Exemple

```
SELECT *  
FROM QSYS2.OBJECT_LOCK_INFO  
WHERE SYSTEM_OBJECT_NAME = 'SALES'
```

RECORD_LOCK_INFO

➤ Vue QSYS2.RECORD_LOCK_INFO

- Retourne une ligne pour chaque ligne verrouillée

- Exemple

```
SELECT JOB_NAME, COUNT(*) AS ROWS_UPDATING
FROM QSYS2.RECORD_LOCK_INFO
WHERE FILE_NAME = 'SALES' AND
      LIBRARY_NAME = 'TOYSTORE' AND
      LOCK_STATE = 'UPDATE'
GROUP BY JOB_NAME
ORDER BY ROWS_UPDATING DESC
```

SCHEDULED_JOB_INFO

➤ Vue **QSYS2.SCHEDULED_JOB_INFO**

- Retourne une ligne par travail planifié

Informations de la commande **WRKJOBSCDE** (Work with Job Schedule Entries)

- Exemple

```
SELECT *  
FROM QSYS2.SCHEDULED_JOB_INFO  
WHERE STATUS IN ('HELD', 'SAVED')  
ORDER BY SCHEDULED_BY;
```


SYSTEM_STATUS

➤ UDF table QSY2.SYSTEM_STATUS

- Retourne une ligne contenant des informations sur la partition courante

Informations des commandes WRKSYSSTS (Work with System Status) et WRKSYSACT (Work with System Activity)

- Exemple

```
SELECT *  
FROM TABLE(  
    QSYS2.SYSTEM_STATUS(RESET_STATISTICS=>'YES')) X;
```

SYSTEM_STATUS_INFO

➤ Vue QSYS2.SYSTEM_STATUS_INFO

- Retourne une ligne contenant des informations sur la partition courante

Informations des commandes **WRKSYSSTS** (Work with System Status) et **WRKSYSACT** (Work with System Activity)

Pour réinitialiser les statistiques, utiliser la fonction **SYSTEM_STATUS**

- Exemple

```
SELECT *  
FROM QSYS2.SYSTEM_STATUS_INFO;
```

SYSTEM_VALUE_INFO

➤ Vue QSY2.SYSTEM_VALUE_INFO

- Retourne la liste des valeurs systèmes et leur valeur
- Exemple

```
SELECT *  
FROM QSYS2.SYSTEM_VALUE_INFO  
WHERE SYSTEM_VALUE_NAME LIKE '%MAX%'
```

DB2 for i Services

Application services

DELIMIT_NAME

➤ Fonction QSY2.DELIMIT_NAME

- Retourne un nom avec les délimiteurs, si nécessaire
- Syntaxe

```
>>-DELIMIT_NAME--(--name--)------><
```

- Exemple

```
VALUES DELIMIT_NAME('ABC'),  
        DELIMIT_NAME('abc'),  
        DELIMIT_NAME('test"name'),  
        DELIMIT_NAME('test' 'name2'),  
        DELIMIT_NAME('NEW')
```

Produit

ABC

"abc"

"test""name"

"test' name2"

"NEW"

OVERRIDE_TABLE

➤ Procédure QSYS2.OVERRIDE_TABLE

- Indique la taille des blocs pour une table
- Syntaxe

```
>>-OVERRIDE_TABLE--(--schema-name-- , --table-name-- , --  
blocking-size--)-><
```

- Exemples

```
CALL QSYS2.OVERRIDE_TABLE('CORPDATA', 'EMP', '*BUF256KB');
```

Ou pour supprimer

```
CALL QSYS2.OVERRIDE_TABLE('CORPDATA', 'EMP', 0);
```

PARSE_STATEMENT

➤ Fonction table QSYS2.PARSE_STATEMENT

- Retourne une liste d'objets et de nom de colonnes utilisés dans une instruction SQL
- Syntaxe

```
>>---PARSE_STATEMENT----(-+-----+---SQL-statement-->
                                '-SQL_STATEMENT-- => -'
'-,+-----+---naming-'
        '-NAMING-- => -'
'-,+-----+---decimal-point-'
        '-DECIMAL_POINT-- => -'
'-,+-----+---SQL-string-delimiter-'>--)-><
        '-SQL_STRING_DELIMITER-- => -'
```

- Avec les paramètres

NAMING : *SYS (défaut) ou *SQL

DECIMAL_POINT : *PERIOD / . (défaut) ou *COMMA / ,

SQL_STRING_DELIMITER : *APOSTSQL / ' ou *QUOTESQL / "

- Exemples

TODO

WLM_SET_CLIENT_INFO

➤ Procédure **SYSPROCS.WLM_SET_CLIENT_INFO**

- Permet de donner une valeur à un registre spécial SQL
- Syntaxe

```
>>-WLM_SET_CLIENT_INFO--(--client_userid--,----->  
>--client_wrkstnname--,--client_applname--,--client_acctng--,--->  
>--client_programid--)-----><
```

Performance services

ACT_ON_INDEX_ADVICE

➤ Procédure **SYSTOOLS.ACT_ON_INDEX_ADVICE**

- Permet de créer les index recommandés pour une table
- Syntaxe

```
>>-ACT_ON_INDEX_ADVICE--(  
  --schema-name--,  
  --table-name--,  
  --times_advised--,  
  --mti_used--,  
  --average_estimate--)-><
```

- Exemple : création des index recommandés plus de 1.000 fois dans pour des tables de la bibliothèque PRODLIB

```
CALL SYSTOOLS.ACT_ON_INDEX_ADVICE(  
  'PRODLIB',NULL,NULL,1000,NULL)
```

DATABASE_MONITOR_INFO

➤ Vue **QSYS2.DATABASE_MONITOR_INFO**

- Retourne les informations sur les moniteurs de base de données, ainsi que les moniteurs d'événement de cache de plan SQL

- Exemple

```
SELECT MONITOR_ID, MONITOR_LIBRARY, MONITOR_FILE  
FROM QSYS2.DATABASE_MONITOR_INFO  
WHERE MONITOR_STATUS = 'ACTIVE' AND  
       MONITOR_TYPE = 'PUBLIC'
```

HARVEST_INDEX_ADVICE

➤ Procédure **SYSTOOLS.HARVEST_INDEX_ADVICE**

- Permet de générer les scripts SQL de création des index recommandés
- Syntaxe

```
>>-HARVEST_INDEX_ADVICE--(--schema-name--,--table-name--,----->  
>--times_advised--,--mti_used--,--average_estimate--,--output-library--,--  
output-file--)-><
```

REMOVE_INDEXES

➤ Procédure **SYSTOOLS.REMOVE_INDEXES**

- Permet de supprimer les index inutilisés depuis une certaine durée
- Syntaxe

```
>>-REMOVE_INDEXES--(--schema-name-- ,--times_used-- ,  
--index-age--)-><
```

- Exemples

```
CALL SYSTOOLS.REMOVE_INDEXES('MYLIB', 1, '1 MONTH')
```

Ou

```
CALL SYSTOOLS.REMOVE_INDEXES(NULL, 100, '14 DAYS')
```

RESET_TABLE_INDEX_STATISTICS

➤ Procédure QSYS2.RESET_TABLE_INDEX_STATISTICS

- Permet de réinitialiser les statistiques d'index pour une table
- Syntaxe

```
>>-RESET_TABLE_INDEX_STATISTICS--(--+-----+--schema-name--,->
                                     '-SCHEMA_NAME-- => -'
>--+-----+--table-name----->
   '-TABLE_NAME-- => -'
>--+-----+--delete-advice-')-----<
   '-,-+-----+--delete-advice-'
   '-DELETE_ADVICE-- => -'
```

- Exemples

```
CALL qsys2.Reset_Table_Index_Statistics('MJATST', 'AMON2')
```

Ou

```
CALL qsys2.Reset_Table_Index_Statistics('MJATST', 'AMON%')
```

Utility services

CANCEL_SQL

➤ Procédure QSYS2.CANCEL_SQL

- Demande l'annulation d'une instruction SQL en cours dans un travail spécifié

- Syntaxe

```
>>-CANCEL_SQL--(--job-name--)-----><
```

- Exemple

```
CALL QSYS2.CANCEL_SQL('483456/QUSER/QZDASOINIT')
```

CHECK_SYSCST

➤ Procédure QSYS2.CHECK_SYSCST

- Compare le contenu de la table QSYS2.SYSCONSTRAINTS entre deux systèmes, pour une bibliothèque

- Syntaxe

```
>>-CHECK_SYSCST--(--remote-rdb-name--,--schema-name--+-----+---)-><  
                                                    '-,--avoid-result-set-'
```

- Exemple

```
CALL SYSTOOLS.CHECK_SYSCST('LP01UT18', 'CORPDB_EX')
```

CHECK_SYSROUTINE

➤ Procédure QSYS2.CHECK_SYSROUTINE

- Compare le contenu de la table QSYS2.SYSROUTINES entre deux systèmes, pour une bibliothèque

- Syntaxe

```
>>-CHECK_SYSROUTINE--(--remote-rdb-name-- ,--schema-name--+-----+--)-><  
                                     '-,--avoid-result-set-'
```

- Exemple

```
CALL SYSTOOLS.CHECK_SYSROUTINE('LP01UT18', 'CORPDB_EX')
```

DUMP_SQL_CURSORS

➤ Procédure QSYS2.DUMP_SQL_CURSORS

- Liste les curseurs ouverts dans un travail
- Syntaxe

```
>>-DUMP_SQL_CURSORS--(--job-name--,
                        --library-name--,
                        --table-name--,
                        --output-option--)-><
```

- Exemple

```
CALL QSYS2.DUMP_SQL_CURSORS('*', 'QGPL', 'SQLCSR1', 3);
```

FIND_AND_CANCEL_QSQRVR_SQL

➤ Procédure QSYS2.FIND_AND_CANCEL_QSQRVR_SQL

- Permet de terminer les travaux SQL serveur QSQRVR pour un travail utilisateur

Utilise les procédures FIND_QSQRVR_JOBS et CANCEL_SQL

- Syntaxe

```
>>-FIND_AND_CANCEL_QSQRVR_SQL--(--job-name--)------><
```

- Exemple

```
CALL QSYS2.FIND_AND_CANCEL_QSQRVR_SQL(  
    '564321/APPUSER/APPJOBNAME')
```

FIND_QSQRVR_JOBS

➤ Procédure QSYS2.FIND_QSQRVR_JOBS

- Retourne des informations sur les travaux SQL serveur QSQRVR pour un travail utilisateur

- Syntaxe

```
>>-FIND_QSQRVR_JOBS--(--job-name--)------><
```

GENERATE_SQL

➤ Procédure QSYS2.GENERATE_SQL

- Génère le script SQL de création d'un objet base de donnée
- Syntaxe

```
>>-GENERATE_SQL--(----->
>--+-----+--object_name----->
  '-DATABASE_OBJECT_NAME-- => -'
>--,--+-----+--object_library---->
  '-DATABASE_OBJECT_LIBRARY_NAME-- => -'
>--,--+-----+--object_type----->
  '-DATABASE_OBJECT_TYPE-- => -'
>--+-----+-->
  '-,--+-----+--source-file-name-'
  '-DATABASE_SOURCE_FILE_NAME-- => -'
... )
```

- Exemples

```
CALL QSYS2.GENERATE_SQL('%', 'SAMPLE_CORPDB', 'TABLE', REPLACE_OPTION =>
'0');
```

Ou

```
CALL QSYS2.GENERATE_SQL('X%', 'SAMPLE_CORPDB', 'INDEX', 'GENFILE',
'DDLSOURCE', 'INDEXSRC', REPLACE_OPTION => '0');
```

RESTART_IDENTITY

➤ Procédure QSYS2.RESTART_IDENTITY

- Permet d'indiquer que les colonnes identité d'une table doivent adopter la valeur suivante des colonne identité d'une autre table

- Syntaxe

```
>>-RESTART_IDENTITY--(--source-schema--,  
                        --source-table--,  
                        --target-schema--,  
                        --target-table--)-><
```

- Exemple

```
CALL QSYS2.RESTART_IDENTITY(  
    'OLDLIB', 'OLDTABLE', 'NEWLIB', 'NEWTABLE')
```