

Modernisation IBM i – Nouveautés 2014-2015

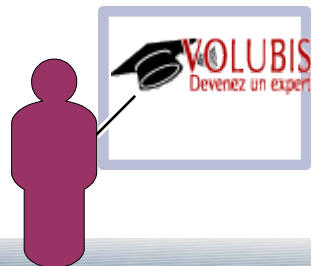
19 et 20 mai 2015 – IBM Client Center, Bois-Colombes

Volubis.fr

Conseil et formation sur OS/400, I5/OS puis IBM *i*
depuis 1994 !

Dans nos locaux, vos locaux ou par Internet

Christian Massé - cmasse@volubis.fr



Modernisation IBM i – Nouveautés 2014-2015

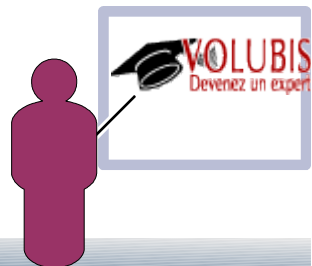
19 et 20 mai 2015 – IBM Client Center, Bois-Colombes

Volubis.fr

Base de connaissance depuis 1995 (plus de 500 cours)

Nouveau *Cours en ligne (accessibles en mode « replay »)*

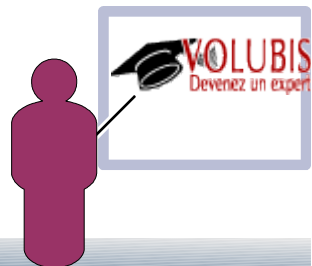
*Accédez à tout cela **gratuitement** pendant trois semaines cet été !*



Modernisation IBM i – Nouveautés 2014-2015

19 et 20 mai 2015 – IBM Client Center, Bois-Colombes

Session 8 : nouveautés SQL au fil de l'eau



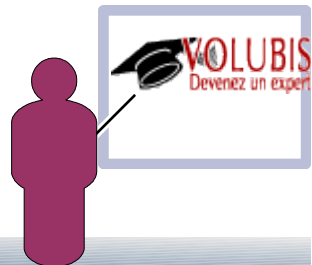
Petit retour en arrière...

OnDemand Tracking System Limits (résumé des épisodes précédents)

- **But du jeu, fournir des alertes sur les événements base de données pouvant impacter les limites du système, plutôt que de subir un arrêt de traitement ou des lenteurs**

**les limites DB2 sont aussi fournies dans le fichier du catalogue
SQL_SIZING**

Les limites sont suffisamment lointaines pour la plupart d'entre nous, mais le dépassement d'un seuil peut être une indication intéressante, voire le signe d'une activité anormale,

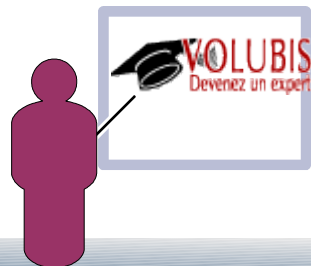


System Limits

OnDemand Tracking System Limits

- **Aux fonctions pouvant présenter une limite, on va associer :**
- **un Seuil**
la première fois que le seuil est dépassé on le signale
- **un Incrément**
quand un incrément, par rapport au seuil précédent est ensuite dépassé, on le signale également

les événements sont notifiés dans un fichier du catalogue système :
SYSLMTTBL



System Limits

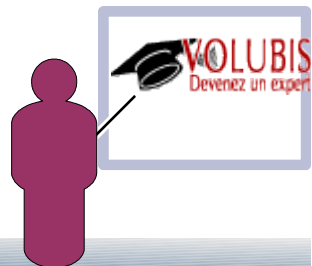
• Contrairement aux autres vues du catalogue SQL, vous pouvez associer à SYSLMTTBL un trigger (AFTER INSERT et/ou AFTER DELETE)

Définition de SYSLMTTBL

LAST_CHANGE_TIMESTAMP	TIMESTAMP	dernière modification
LIMIT_CATEGORY	SMALLINT	0 : Database 1 : Journal 2 : Sécurité 3 : Divers 4 : Gestion des travaux 5 : Système de fichier 6 : Sauvegarde/restauration 7 : Gestion de Cluster 8 : Communication
LIMIT_TYPE	SMALLINT	1 : Objet 2 : Travail 3 : Système 4 : ASP
LIMIT_ID	INTEGER	pour faire la jointure avec SQL_SIZING
JOB_NAME	VARCHAR(28)	coordonnées du travail
USER_NAME	VARCHAR(10)	Profil
CURRENT_VALUE	BIGINT	valeur rencontrée
SYSTEM_OBJECT-NAME	VARCHAR(30)	objet concerné ou NULL
SYSTEM_TABLE_MEMBER	VARCHAR(10)	membre ou NULL
OBJECT-TYPE	VARCHAR(7)	type d'objet
ASP_NUMBER	SMALLINT	N° d'ASP

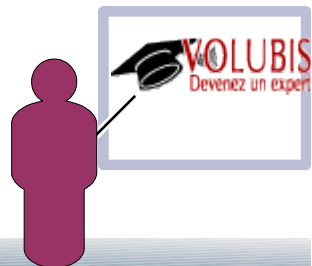
System Limits

- une vue est proposée **SYSLIMITS** faisant le lien entre **SYSLMTTBL**, **SQL_SIZING** et avec la fonction table **GET_JOB_INFO()**
- Il a un système de nettoyage automatique (**PRUNE**) sur **SYSLMTTBL**
 - EN V6, les 20 lignes les plus anciennes sont supprimées
 - EN V7, ce chiffre est paramétrable à l'aide de variables globales :
 - `CREATE VARIABLE SYSIBM.QIBM_SYSTEM_LIMITS_PRUNE_BY_ASP INTEGER DEFAULT 100`
 - `CREATE VARIABLE SYSIBM.QIBM_SYSTEM_LIMITS_PRUNE_BY_JOB INTEGER DEFAULT 50`
 - `CREATE VARIABLE SYSIBM.QIBM_SYSTEM_LIMITS_PRUNE_BY_OBJECT INTEGER DEFAULT 20`
 - `CREATE VARIABLE SYSIBM.QIBM_SYSTEM_LIMITS_PRUNE_BY_SYSTEM INTEGER DEFAULT 100`



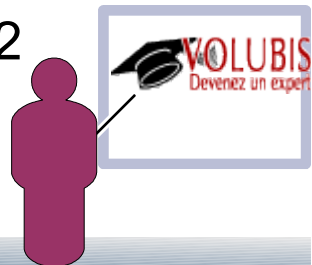
System Limits

- Les fonctions de la phase 1 (7.1 / TR6)
 - Système
 - nbr maxi de jobs (seuil à 1000 / incrément à 400)
 - ASP
 - nbr maxi de spool par ASP (10000 / 5000)
 - Objets
 - nbr maxi de membres (100 / 100)
 - Travaux
 - nbr maxi de lignes verrouillées (10000 / 100000)
 - nbr maxi de ligne dans une transaction (10000 / 100000)



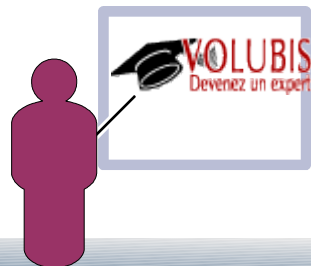
System Limits

- Les fonctions de la phase 2 (7.1 / TR7)
- base de données
 - nbr maxi de lignes dans un membre
 - nbr maxi de lignes actives
 - nbr maxi de lignes supprimées
 - nbr maxi de lignes en excédent (notion d'incrément)
 - nbr maxi de variables à taille variable
(sont gérés comme des extensions)
 - nbr maxi d'index autour d'une table
- journal
 - taille maxi (1 To) d'un récepteur
 - nbr maxi d'objets attachés à un journal *MAX10M
 - nbr maxi d'objets attachés à un journal *MAX250K
 - plus grand n° de séquence d'un journal *MAXOPT3
 - plus grand n° de séquence d'un journal *MAXOPT1, *MAXOPT2
- système de fichiers
 - nbr maxi d'objets attachés à une bibliothèque



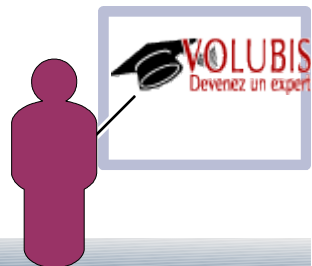
System Limits

- Les fonctions de la phase 3 (7.2 / TR1)
 - nbr maxi de fichiers IFS attachés à un répertoire
 - nbr maxi de répertoires attachés à un répertoire
 - nbr maxi d'objets IFS attachés à l'ASP *SYSBAS
 - nbr maxi d'objets IFS attachés à un ASP utilisateur
 - nbr maxi de documents attachés à un folder (QDLS)
 - nbr maxi de documents attachés à l'ASP *SYSBAS
 - nbr maxi de documents attachés à un ASP utilisateur
 - taille maxi d'un fichier de l'IFS
 - taille maxi d'un document (QDLS)
-
- Chaque seuil est identifié par un ID (LIMIT_ID) ,
tous les nouveaux sont dans la plage 18402 / 18410



System Limits

- **Les fonctions de la phase 4 (7.2 / TR2)**
- **Taille maxi d'un index *MAX4GB (15400)**
 - Limite : 4.294.967.296
 - Seuil : 838.860.800
 - Incrément : 167.772.160
- **Taille maxi d'un index *MAX1TB (15401)**
 - Limite : 1.869.166.411.776
 - Seuil : 8.388.608.00
 - Incrément : 8.388.608.00
- **Taille maxi d'un index EVI (15403)**
 - Limite : 2.199.023.255.552
 - Seuil : 1.677.721.600
 - Incrément : 8.388.608.00



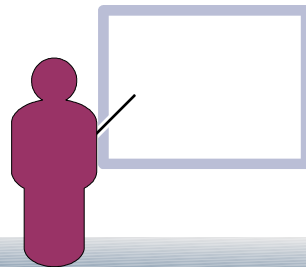
Registres client

Les registres clients (Client ProgramID, Client APPLNAME, etc...) sont désormais aussi renseignés pour :

- QUERY
- QM
- STRSQL
- RUNSQLSTM

Ils le sont aussi en SF99701, Level 29 pour DDM/DRDA

- Client PROGRAMID 'DRDA'
- Client APPLNAME 'xxxVVrrMM'
 - Xxx représente le signature du client 'QSQ' (IBM i) / 'SQL' (DB2 Luw) etc...
 - VvRRmm , la version ('QSQ070100' = IBM i 7.1)
- Client USERID profil utilisateur client
- Client WORKSTNNAME le nom de la machine cliente
- Client ACCNTG code accounting coté client



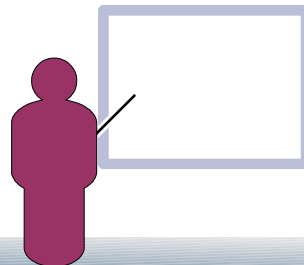
Registres client

Depuis la TR1, ces registres clients sont accessibles par API

- o l'API **QUSRJOBI** permet de retrouver certaines informations système dans le format JOBI0900 (7.2 uniquement)

192	C0	BINARY(4)	Offset to client accounting string -> registre CLIENT_ACCSTR
196	C4	BINARY(4)	Length of client accounting string
200	C8	BINARY(4)	Offset to client application name -> registre CLIENT_APPLNAME
204	CC	BINARY(4)	Length to client application name
208	D0	BINARY(4)	Offset to client program identifier -> registre CLIENT_PROGRAMMID
212	D4	BINARY(4)	Length to client program identifier
216	D8	BINARY(4)	Offset to client user identifier -> registre CLIENT_USERID
220	DC	BINARY(4)	Length of client user identifier
224	E0	BINARY(4)	Offset to client workstation name -> registre CLIENT_WORKSTNNAME
228	E4	BINARY(4)	Length of client workstation name

264	108	CHAR(1)	Client IP address for Server job type (1 : IPV4, 2 IPV6)
265	109	CHAR(45)	Client IP address for Server job (adresse IP)
388	184	BINARY(4)	Client port number (port IP)
392	188	BINARY(4)	Offset to client host name
396	18C	BINARY(4)	Length of client host name



DB2 for i service

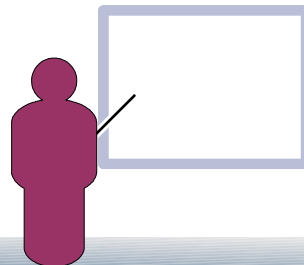
·Autre notion ancienne (maintenant) mais qui évolue en permanence : SQL as a service ou IBM i service : accéder aux informations IBM i en SQL

EN TR1, deux nouvelles variables globales (7.2 uniquement)

JOB_NAME retourne les coordonnées du job en cours

values JOB_NAME
00001
041417/QUSER/QZDASOINIT

SERVER_MODE_JOB_NAME retourne les coordonnées du job serveur
(quand la requête est exécutée en mode serveur : QSQSRVR)



DB2 for i service

·Autre notion ancienne (maintenant) mais qui évolue en permanence : SQL as a service ou IBM i service : accéder aux informations IBM i en SQL

vue SYSTOOLS.GROUP_PTF_CURRENCY

Affiche vos groupes PTF, va chercher les niveaux actuels sur le Net (HTTPGETBLOB + XMLTABLE) pour vous indiquer si vous êtes à jour.

```
PTF_GROUP_CURRENCY          PTF_GROUP_ID  PTF_GROUP_TITLE
INSTALLED LEVEL IS CURRENT  SF99480       720 WebSphere App Server V8.0
INSTALLED LEVEL IS CURRENT  SF99481       720 WebSphere App Server V8.5
INSTALLED LEVEL IS CURRENT  SF99702       720 DB2 for IBM i
INSTALLED LEVEL IS CURRENT  SF99713       720 IBM HTTP Server for i
INSTALLED LEVEL IS CURRENT  SF99715       720 Backup Recovery Solutions
INSTALLED LEVEL IS CURRENT  SF99716       720 Java
INSTALLED LEVEL IS CURRENT  SF99717       720 Technology Refresh
INSTALLED LEVEL IS CURRENT  SF99718       720 Group Security
INSTALLED LEVEL IS CURRENT  SF99719       720 Group Hiper
INSTALLED LEVEL IS CURRENT  SF99720       Current Cumulative PTF Media Documentation
INSTALLED LEVEL IS CURRENT  SF99721       720 All PTF Groups except Cumulative PTF Package
INSTALLED LEVEL IS CURRENT  SF99747       720 DB2 Web Query for i V2.1.0
INSTALLED LEVEL IS CURRENT  SF99766       720 Print PTFs
INSTALLED LEVEL IS CURRENT  SF99775       720 Hardware and Related PTFs
INSTALLED LEVEL IS CURRENT  SF99776       720 High Availability for IBM i
***** Fin de données *****
```

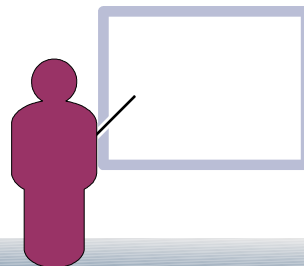
DB2 for i service

·Autre notion ancienne (maintenant) mais qui évolue en permanence : SQL as a service ou IBM i service : accéder aux informations IBM i en SQL

vue SYSTMPSTG

permet d'accéder à la mémoire temporaire utilisée (job et système)

BUCKET_NUMBER	JOB_NAME	BUCKET_CURRENT_SIZE
65.905	ADMIN2	218.832.896
65.692	ADMIN4	183.697.408
65.688	ADMIN1	163.434.496
65.695	ADMIN5	162.545.664
65.676	QUSRDIR	153.526.272
65.570	QTCPWRK	151.486.464
65.904	ADMIN3	145.272.832
65.605	QYPSPFCOL	104.927.232



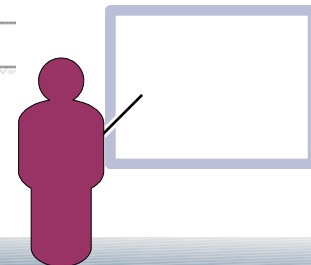
DB2 for i service

·Autre notion ancienne (maintenant) mais qui évolue en permanence : SQL as a service ou IBM i service : accéder aux informations IBM i en SQL

fonction table JOBLOG_INFO, retourne l'historique d'un travail

```
SELECT * FROM  
TABLE(QSYS2.JOBLOG_INFO(QSYS2.job_name)) AS J
```

ORDINAL_POSITI...	MESSAGE_...	MESSAGE_TYPE	MESSA...	SEVERITY	MESSAGE_TIMESTAMP
1	CPF1124	INFORMATIONAL		0	2014-11-17 14:50:49.737602
2	CPF1301	INFORMATIONAL		30	2014-11-17 14:50:53.780581
3	CPIAD02	INFORMATIONAL		0	2014-11-17 14:50:53.780776
4	SQL799C	INFORMATIONAL		10	2014-11-17 14:50:54.016747
5	CPI321A	INFORMATIONAL		0	2014-11-17 14:50:55.686222
6	CPI321A	INFORMATIONAL		0	2014-11-17 14:50:55.686235
7	CPI321A	INFORMATIONAL		0	2014-11-17 14:50:55.686243
8	CPI321A	INFORMATIONAL		0	2014-11-17 14:50:55.686250
9	CPI321A	INFORMATIONAL		0	2014-11-17 14:50:55.686264
10	CPI321A	INFORMATIONAL		0	2014-11-17 14:50:55.686272
11	CPI2101	INFORMATIONAL		0	2014-11-17 14:50:55.987344
12	CPC2130	COMPLETION		0	2014-11-17 14:50:55.987452



DB2 for i service

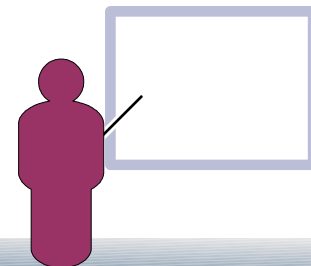
·Autre notion ancienne (maintenant) mais qui évolue en permanence : SQL as a service ou IBM i service : accéder aux informations IBM i en SQL

vue LIBRARY_LIST_INFO, retourne *LIBL du travail en cours

```
CL: addlibl BDVIN1;
```

```
select * from qsys2.LIBRARY_LIST_INFO;
```

ORDINAL_POSITION	SCHEMA_NAME	SYSTEM_SCHEMA_NAME	TYPE	IASP...	TEXT_DESCRIPTION
1	QSYS	QSYS	SYSTEM	0	Bibliothèque système
2	QSYS2	QSYS2	SYSTEM	0	Bibliothèque système pour CPI
3	QHLP SYS	QHLP SYS	SYSTEM	0	-
4	QUSR SYS	QUSR SYS	SYSTEM	0	System Library for Users
5	QIWS	QIWS	PRODUCT	0	-
6	AF4W	AF4W	CURRENT	0	AF400 : bibliothèque de travail (CM)
7	BDVIN1	BDVIN1	USER	0	Collection créée par SQL
8	QG PL	QG PL	USER	0	General Purpose Library
9	QTEMP	QTEMP	USER	0	-



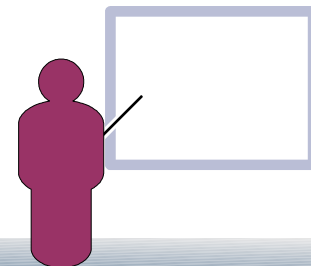
DB2 for i service

·Autre notion ancienne (maintenant) mais qui évolue en permanence : SQL as a service ou IBM i service : accéder aux informations IBM i en SQL

vue REPLY_LIST_INFO retourne la liste de réponse système

```
select * from qsys2.REPLY_LIST_INFO;
```

SEQUENCE_NUMBER	MESSAGE_ID	MESSAGE_REPLY	COMPARISON_DATA	COMPARISON_DATA_OFFSET	DUMP_JOB
1	CPA8424	G	-	-	-NO
10	CPA0700	D	-	-	-YES
15	CPA7025	I	-	-	-NO
20	RPG0000	D	-	-	-YES
30	CBE0000	D	-	-	-YES
40	PLI0000	D	-	-	-YES
955	CPA5737	C	Q1PLIN Q1PCTL	6	NO
3282	CPA32B2	I	iSeries Navigator	-	-NO



DB2 for i service

·Autre notion ancienne (maintenant) mais qui évolue en permanence : SQL as a service ou IBM i service : accéder aux informations IBM i en SQL

vue JOURNAL_INFO retourne la liste des journaux et leurs attributs

```
select * from qsys2.JOURNAL_INFO;
```

JOURNAL_NAME	JOURNAL_LIBRARY	ASP_...	JOURNAL_ASPGRP	ATTACHED_JOURNAL_RECEIVER_NAME	ATTACHED_JOURNAL_RECEIVER_LIBRARY
QSQRN	AF4SRCT	1	*SYSBAS	JRNRCV4199	AF4SRCT
JRN	AF4TEST	1	*SYSBAS	JRN7291	AF4TEST
QSQRN	AF4TEST	1	*SYSBAS	RCV50002	AF4TEST
JCOMMIT	AF4TOOL	1	*SYSBAS	JCOMRC6001	AF4TOOL
QSQRN	AF4W	1	*SYSBAS	QSQRN4200	AF4W
AF4JRN	AF400	1	*SYSBAS	AF4RCV7298	AF400
ARCJRN	ARCAD_PRD	1	*SYSBAS	RCVARC1023	ARCAD_PRD
QSQRN	BDVIN_00	1	*SYSBAS	QSQRN6267	BDVIN_00
QSQRN	BDVINA	1	*SYSBAS	QSQRN3137	BDVINA
QSQRN	BDVINB	1	*SYSBAS	QSQRN1047	BDVINB
QSQRN	BDVINU	1	*SYSBAS	QSQRN0002	BDVINU
QSQRN	BDVIN0	1	*SYSBAS	JRNRCV4194	BDVIN0
QSQRN	BDVIN1	1	*SYSBAS	QSQRN6239	BDVIN1
QSQRN	BDVIN2	1	*SYSBAS	QSQRN4198	BDVIN2
QSQRN	BDVIN3	1	*SYSBAS	QSQRN4194	BDVIN3
QSQRN	BDVIN4	1	*SYSBAS	QSQRN4194	BDVIN4
QSQRN	BDVIN5	1	*SYSBAS	QSQRN4194	BDVIN5
QSQRN	BDVIN6	1	*SYSBAS	QSQRN4195	BDVIN6
QSQRN	BDVIN7	1	*SYSBAS	QSQRN4194	BDVIN7
QSQRN	BDVIN8	1	*SYSBAS	QSQRN4194	BDVIN8
QSQRN	BDVIN9	1	*SYSBAS	QSQRN4195	BDVIN9
JRN	CLUBCIB	1	*SYSBAS	RCV6266	CLUBCIB
JRN	CLUBPL	1	*SYSBAS	RCV7296	CLUBPL

DB2 for i service

·Autre notion ancienne (maintenant) mais qui évolue en permanence : SQL as a service ou IBM i service : accéder aux informations IBM i en SQL

Toujours en TR1, de nouvelles procédures stockées afin de gérer le cache des plans d'accès

IMPORT_PC_SNAPSHOT , importe une image dans la liste des images affichées par System i Navigator

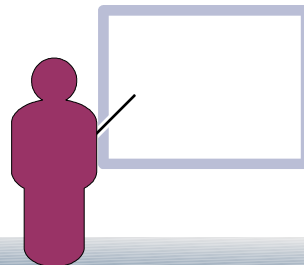
REMOVE_PC_SNAPSHOT , supprime une image du cache des plans d'accès

IMPORT_PC_EVENT_MONITOR , créé un moniteur d'événement sur le cache des plans d'accès

REMOVE_PC_EVENT_MONITOR , supprime un moniteur d'événement sur le cache des plans d'accès

Avant ces quatre opérations ne pouvaient être que manuelles
(via System i Navigator)

TR2 : CLEAR_PLAN-CACHE() **évite un IPL lors de phases de test.**



DB2 for i service

·Autre notion ancienne (maintenant) mais qui évolue en permanence : SQL as a service ou IBM i service : accéder aux informations IBM i en SQL

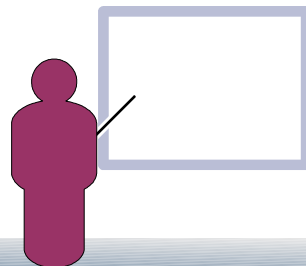
TR2 (prévue pour fin Mai)

Vue **DRDA_AUTHENTICATION_ENTRY_INFO** fournit la liste des authentications DDM/DRDA enregistrées par ADDSVRAUTE

la fonction table **ACTIVE_JOB_INFO()** retourne l'équivalent de WRKACTJOB (possibilité de filtrer sur le job ou le sous système)

la vue **SCHEDULED_JOB_INFO** retourne l'équivalent de WRKJOBSCDE

la fonction **OBJECT_STATISTICS()** qui retourne l'équivalent de DSPOBJD admet maintenant un nom ou un nom générique d'objet
(en plus de la bibliothèque et du type d'objet)



DB2 for i service

·Autre notion ancienne (maintenant) mais qui évolue en permanence : SQL as a service ou IBM i service : accéder aux informations IBM i en SQL

TR2 (prévue pour fin Mai)

La procédure SET_SERVER_SBS_ROUTING(user, serveur, sbs)

créé un sous système pour un serveur :

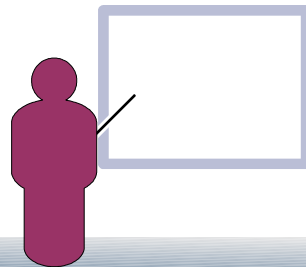
QZDASOINIT[serveur database]

QRWTSRVR[ddm-drda]

ou *ALL

et pour un utilisateur ou un groupe d'utilisateurs

la vue SERVER_SBS_ROUTING retourne la liste des sous systèmes créés par SET_SERVER_SBS_ROUTING



DB2 for i service

·Autre notion ancienne (maintenant) mais qui évolue en permanence : SQL as a service ou IBM i service : accéder aux informations IBM i en SQL

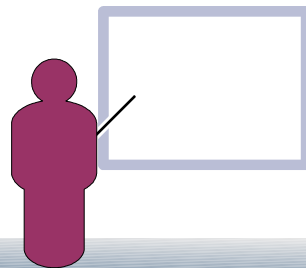
TR2 (prévue pour fin Mai)

La procédure SET_JVM()

permet de fixer des options à la JVM d'un job :

GC_ENABLED_VERBOSE | GC_DISABLED_VERBOSE
GENERATE_HEAP_DUMP
GENERATE_SYSTEM_DUMP
GENERATE_JAVA_DUMP

la vue JVM_INFO, retourne des informations sur les JVM actives



DB2 for i service

·Autre notion ancienne (maintenant) mais qui évolue en permanence : SQL as a service ou IBM i service : accéder aux informations IBM i en SQL

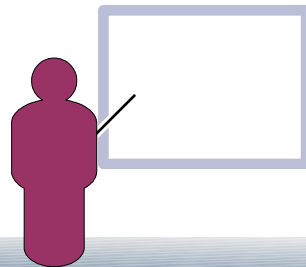
TR2 (prévue pour fin Mai)

La vue DATABASE_MONITOR_INFO

Retourne la liste des moniteurs base de données (STRDBMON)
et moniteurs d'événement de mémoire cache connus

Pour une liste complète des services, voyez :

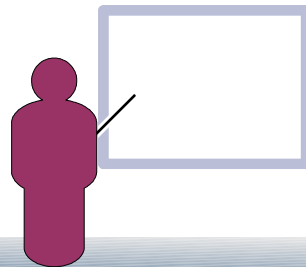
<https://www.ibm.com/developerworks/community/wikis/home?lang=en#!/wiki/IBM%20i%20Technology%20Updates/page/DB2%20for%20i%20-%20Services>



7.2 (2014)

- Nouveau registre CURRENT_USER
- Nouvelles instructions
 - TRUNCATE
 - TRANSFERT OWNERSHIP
 - Nouvelles fonctions
 - LPAD / RPAD
 - VERIFY_GROUP_FOR_USER(profil, groupe)
- Les TIMESTAMP en SQL ET en RPG supportent que vous indiquiez une précision (nombre de décimales après la seconde) entre **0 et 12**
- CREATE MASK indique si une colonne est retournée tel-que ou masquée ('xxx-xxx-xxx-1234' pour un n° de CB)
- CREATE PERMISSION indique si une ligne doit être retournée
pour ce dernier point voir la session 14 de demain matin (9h)

C'est ces deux points qui ont poussé à l'utilisation complète de SQE, y compris par les applications natives, Query et OPNQRYF



7.2 (TR1)

- Expression régulières
- ICU doit être installé sur votre système

```
5770SS1 39 Composants internationaux pour Unicode
```

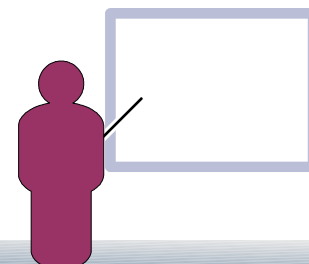
Il s'agit de rechercher si une chaîne de caractère ressemble à un modèle (un peu comme LIKE, mais en plus puissant)

```
>>-REGEXP_xxxx--(--origine--,--'modèle recherché'----->  
>--+-----+--+-----)----->  
'-,--position de début-' '-,--option-'
```

la variable origine doit être un champ de type Chaîne, numérique ou date-time convertible en UTF-16 (Unicode).

■ options proposées

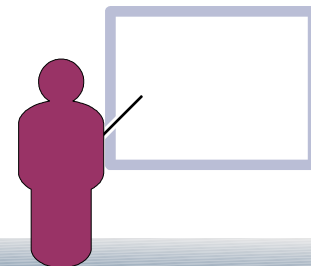
c	case sensible
i	case insensible
x	espaces ignorés
n	fin de ligne matérialisé par .
m	multi-ligne



7.2 (TR1)

- Modèle
- Tout caractère est recherché tel que
- Quelques méta-caractères utilisables dans une expression régulière

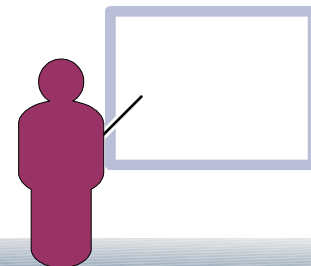
\w	tout caractère alpha ou numérique (pouvant constituer un mot)
\W	tout sauf caractère alpha ou numérique
\d	un chiffre
\D	tout sauf un chiffre
\s	caractère espace
\S	tout sauf un espace
[modèle]	tout caractère respectant le modèle
(suite)	regroupement (<i>voir exemple après les opérateurs logiques</i>)
.	tout caractère
^	début de ligne
\$	fin de ligne
\	déspecialise le caractère suivant (\. pour chercher un point)
\N	caractère Unicode du nom indiqué (voir unicode.org)
\u	caractère Unicode de la valeur hexa indiquée (voir unicode.org)
\U	tout sauf le caractère unicode indiqué



7.2 (TR1)

- Modèle
- Tout caractère est recherché tel que
- Quelques méta-caractères utilisables dans une expression régulière (suite)

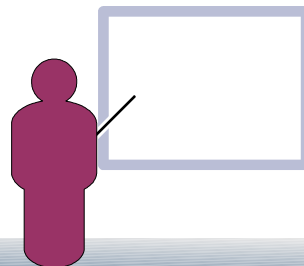
<code>\p{propriété Unicode}</code> (voir Unicode Categories)	<code>\p{L}</code> -> une lettre <code>\p{N}</code> -> un chiffre <code>\p{P}</code> -> une ponctuation <code>\p{Z}</code> -> un espace <code>\p{M}</code> -> Mark (accents latins[<code>^..</code>], umlauts allemand, etc) <code>\p{S}</code> -> un symbole mathématique
<code>\P{propriété Unicode}</code>	<code>\P{L}</code> -> tout sauf une lettre <code>\P{N}</code> -> tout sauf un chiffre <code>\P{P}</code> -> tout sauf une ponctuation <code>\P{Z}</code> -> tout sauf un espace <code>\P{M}</code> -> tout sauf Mark <code>\P{S}</code> -> tout sauf un symbole mathématique



7.2 (TR1)

- Modèle
- Quelques exemples de modèles

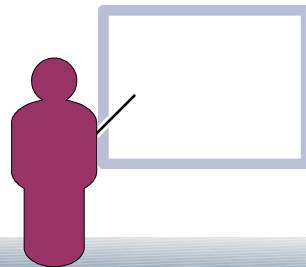
[abc]	un a ou un b ou un c
[^abc]	tout sauf a, b et c
[C-M]	toute lettre comprise entre C et M
[a-z]	une minuscule
[A-Z]	une majuscule
[0-9]	un chiffre
[a-zA-Z0-9]	une minuscule ou une majuscule ou un chiffre
[a-z][A-Z][0-9]	une minuscule ou une majuscule ou un chiffre



7.2 (TR1)

- Modèle
- Quelques opérateurs logiques

	ou ($a b \rightarrow a$ ou b)
*	0 à n fois l'expression précédente
+	1 à n fois l'expression précédente
?	0 ou 1 fois l'expression précédente
{n}	n fois exactement l'expression précédente
{n,}	n fois ou plus l'expression précédente
{n,m}	entre n et m fois l'expression précédente
*?	0 fois ou plus, le moins de fois possible
+?	1 fois ou plus, le moins de fois possible
??	0 ou 1 fois, 0 de préférence



7.2 (TR1)

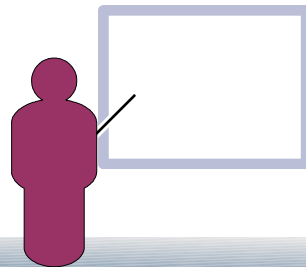
- Modèle
 - Les parenthèses servent à regrouper des caractères ou des méta-caractères, particulièrement pour les opérateurs logiques
 - `abc|def` signifie a,b,(c ou d), e, f
 - `(abc)|(def)` signifie abc ou def
 - `abc+` est vrai pour abc, abcc, abccc, abccccccc
 - `(abc)+` est vrai pour abc, abcabc, abcabcabc, etc...

par exemple, cette expression

- `(\w+\.)+((org)|(com)|(gouv)|(fr))`

permet de rechercher un suite de caractères alphanumériques `\w+` , suivie d'un point `\.` , cette série étant présente éventuellement plusieurs fois `(\w+\.)+`

puis soit org, soit com, soit gouv, soit fr : `(org)|(com)|(gouv)|(fr)`



7.2 (TR1)

- REGEXP_LIKE

- s'utilise comme un LIKE mais avec la puissance des expressions régulières

Chateau ou Château ?

```
select pr_nom
  from bdvin1/producteurs
 where REGEXP_LIKE(pr_nom, 'Ch[aâ]teau')
```

```
PR_NOM
Château de Fuissé
Château des Jacques
Château Thivin, Claude Geoffray
Château de Puligny-Montrachet
Château de Davenay
Château de Nervers
```

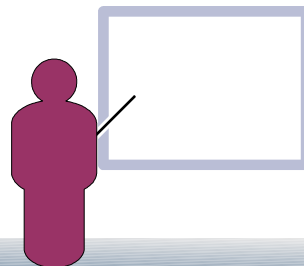
Bouscassé ou Bouscasé ou Bouscacé, bref : un ou plus du couple (S ou C)

```
select pr_nom
  from bdvin1/producteurs
 where REGEXP_LIKE(pr_nom, 'Bousca[sc]+é')
```

```
PR_NOM
Château Montus-Château Bouscassé
```

vous pouvez ajouter l'option i, pour être insensible à la casse

```
select pr_nom
  from bdvin1/producteurs
 where REGEXP_LIKE(pr_nom, 'bousca[sc]+é', 'i')
```



7.2 (TR1)

- REGEXP_LIKE

- Un n° de tel doit avoir
2 chiffres-1 espace-2 chiffres-1 espace-2 chiffres-1 espace-2 chiffres

liste des téléphones qui ne respectent pas cette règle

```
select pr_tel
  from bdvin1/producteurs
 where not REGEXP_LIKE(pr_tel, '\d\d\s\d\d\s\d\d\s\d\d')
```

PR_TEL

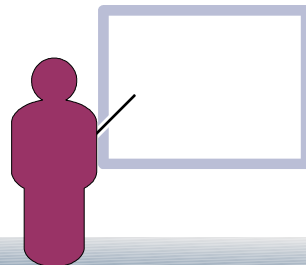
0173 78 1 14
0173 50 2 76
0173 78 5 86
02735 - 22 32
02735 - 24 74
02734 - 36 78

liste des téléphones contenant une lettre

```
select pr_tel
  from bdvin1/producteurs
 where
  REGEXP_LIKE(pr_tel, '\p{L}')
```

PR_TEL

03 80 61 08 17 et 03
04 74 04 12 35 et 04
03 80 21 30 43 et 03
02 47 52 62 55 et 02
04 68 55 01 01 et 04
04 94 66 80 20 et 04
04 94 28 20 17 et 0
097-55330à



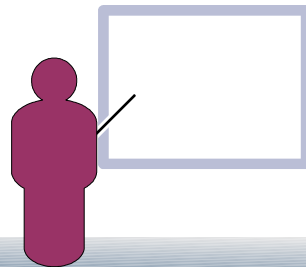
7.2 (TR1)

- REGEXP_COUNT
- compte le nbr de fois ou une expression régulière est vraie
(au moins 2 fois le mot château)

```
select * from bdvin1/producteurs  
where REGEXP_COUNT(pr_nom, 'Ch[aâ]teau')>1
```

PR_NOM
Château Montus-Château Bouscassé


Si M. Brumont (propriétaire dudit château, appellation Madiran), me lit, mon email est en début de présentation, nous acceptons les colis ;-)



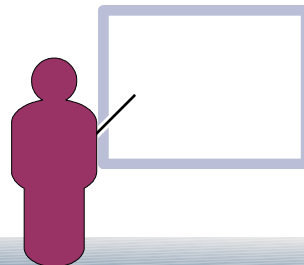
7.2 (TR1)

- REGEXP_INSTR
- retourne la position de la chaîne respectant l'expression régulière

```
select pr_nom, regexp_instr(pr_nom, 'Ch[aâ]teau')  
  from bdvin1/producteurs  
  where REGEXP_LIKE(pr_nom, 'Ch[aâ]teau')  
 order by 2 desc
```



PR_NOM	REGEXP_INSTR
Domaine Langlois-Château	18
Grand Enclos du Château de Cérons	17
Le Couvent de Château Peyros	15
Domaine du Château de la Valette	12
Domaine du Château	12
Domaine du Château de Praz	12
Champagne Château de Boursault	11
Langlois-Château	10
Caves du Château d'Auvernier	10
Cave du Château de Chénas	9
Vieux Château Champs de Mars	7
Vieux Château Ferron	7
Vieux Château l'Abbaye	7
Vieux Château Chambeau	7
Vieux Château Landon	7
Vieux Château Gaubert	7
Vieux Château Certan	7
Château de Fuissé	1
Château des Jacques	1



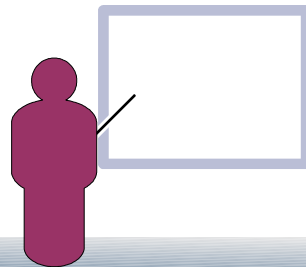
7.2 (TR1)

- REGEXP_SUBSTR
- retourne la chaîne respectant l'expression régulière

```
select pr_nom, regexp_substr(pr_nom, 'Ch[aâ]teau')  
  from bdvin1/producteurs  
  where REGEXP_LIKE(pr_nom, 'Ch[aâ]teau')
```




PR_NOM	REGEXP_SUBSTR
Château de Fuissé	Château
Château des Jacques	Château
Château Thivin, Claude Geoffray	Château
Château de Puligny-Montrachet	Château
Château de Davenay	Château
Château de Nervans	Château



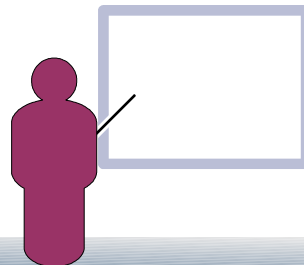
7.2 (TR1)

- REGEXP_REPLACE
- remplace (ou enlève) la chaîne correspondant à une expression régulière

```
select pr_nom, regexp_replace(pr_nom, 'Ch[aâ]teau', 'castel')  
from bdvin1/producteurs  
where REGEXP_LIKE(pr_nom, 'Ch[aâ]teau')
```



PR_NOM	REGEXP_REPLACE
Château de Fuissé	castel de Fuissé
Château des Jacques	castel des Jacques
Château Thivin, Claude Geoffray	castel Thivin, Claude Geoffray
Château de Puligny-Montrachet	castel de Puligny-Montrachet
Château de Davenay	castel de Davenay
Château de Nervers	castel de Nervers

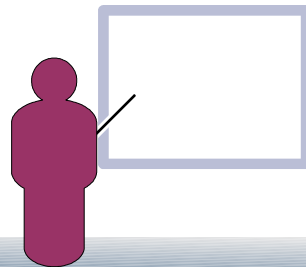


7.2 (TR1)

- Toutes ces fonctions peuvent, bien sûr, être utilisées directement dans les programmes SQLRPG, SQLRPGLE (ou Cobol)

```
dcl-s msg char(120)
inz('bienvenue sur le site www.volubis.fr. conseil et formation-
sur IBM i');
dcl-s exp varchar(45) inz('(\w+\.)+((org)|(com)|(gouv)|(fr))');
dcl-s str char(40);
dcl-s ind int(5);
EXEC SQL
VALUES Regexp_Substr(:Msg, :Exp) INTO :Str:Ind;
if ind = 0;
  dsply str;
else;
  dsply 'non trouvé';
endif;
*inLR = *on;
```

```
call regexp
DSPLY www.volubis.fr
```



7.2 (TR1)

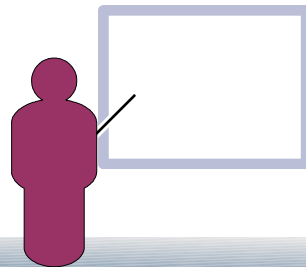
- Instruction PIPE

nouvelle instruction du SQL PSM offrant une alternative aux fonctions tables pour retourner une ligne à la fois

- Contrairement aux fonctions tables écrites en RPG qui retournent une ligne à chaque appel, les fonctions tables écrites en PL/SQL devaient retourner le résultat d'un Select SQL

```
CREATE FUNCTION BDVIN0.LESVINS() RETURNS TABLE(NOM CHARACTER (25), PRODUCTEUR CHARACTER (30), CEPAGE CHARACTER (20))  
LANGUAGE SQL READS SQL DATA  
RETURN  
SELECT left(MIN_NOM, 25), left(PR_NOM, 30), left(MIN_CEPAGE1, 20) FROM BDVIN0.VINS JOIN BDVIN0.PRODUCTEURS USING(pr_code)  
;
```

Désormais nous pouvons ajouter de la logique à cela :



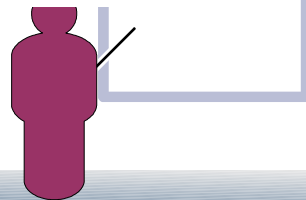
7.2 (TR1)

- Instruction PIPE

```
CREATE FUNCTION BDVIN0.LESMINSCHERS()
RETURNS TABLE( NOM CHARACTER (25) , PRODUCTEUR CHARACTER (30) , CEPAGE CHARACTER (20) )
LANGUAGE SQL READS SQL DATA
BEGIN
DECLARE code int;
DECLARE NOM CHAR(25);
DECLARE PR_NOM CHAR(30);
DECLARE CEPAGE CHAR(20);
DECLARE EOF INT;
DECLARE PRDX DEC(13 , 2 );
DECLARE C1 CURSOR FOR SELECT vin_code, left(VIN_NOM, 25), left(PR_NOM, 30) , left(VIN_CEPAGE1, 20) FROM BDVIN0.VINS JOIN BDVIN0.PRODUCTEURS USING(pr_code);
DECLARE CONTINUE HANDLER FOR SQLSTATE '02000'
    SET EOF = 1;

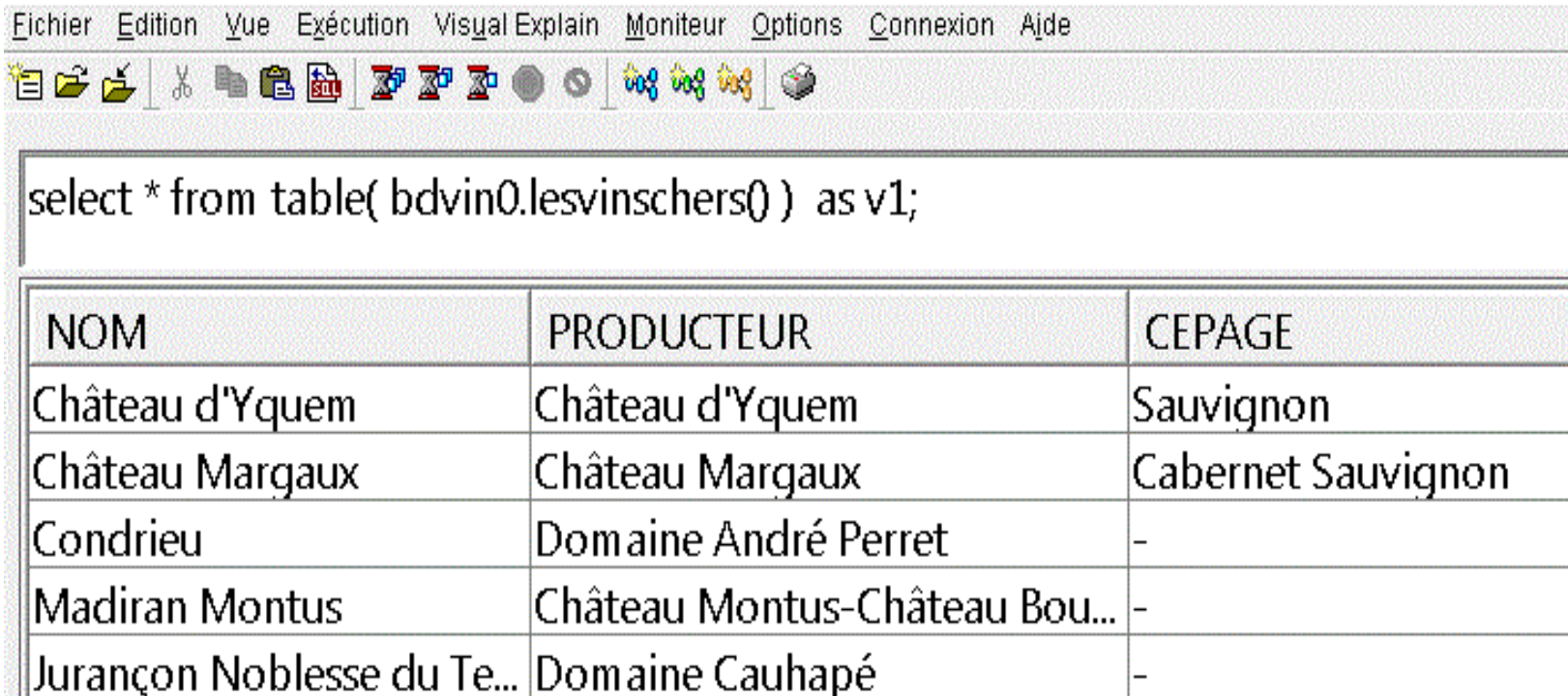
    OPEN C1;
BCL: LOOP
    FETCH C1 INTO code, NOM, PR_NOM, CEPAGE;
    IF EOF = 1 THEN
        LEAVE BCL;
    END IF;
    select max(cav_prxactuel) into PRDX from bdvin0.ma_cave where vin_code = code;
    if prix > 20 THEN
        PIPE (NOM, PR_NOM, CEPAGE);
    END IF;
END LOOP;

CLOSE C1;
RETURN;
END;
```



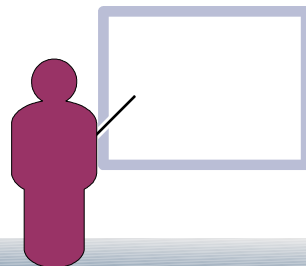
7.2 (TR1)

- Instruction PIPE



The screenshot shows a software interface with a menu bar (Fichier, Edition, Vue, Exécution, Visual Explain, Moniteur, Options, Connexion, Aide) and a toolbar. The main window contains a SQL query: `select * from table(bdvin0.lesvinschers()) as v1;`. Below the query is a table with three columns: NOM, PRODUCTEUR, and CEPAGE. The table contains five rows of data.

NOM	PRODUCTEUR	CEPAGE
Château d'Yquem	Château d'Yquem	Sauvignon
Château Margaux	Château Margaux	Cabernet Sauvignon
Condrieu	Domaine André Perret	-
Madiran Montus	Château Montus-Château Bou...	-
Jurançon Noblesse du Te...	Domaine Cauhapé	-



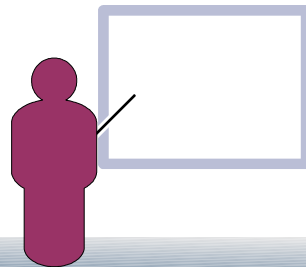
7.2 (TR1)

- Dans le même temps le debug du SQL PSM est amélioré

Un programme ou une fonction écrite en PL/SQL, génère du C avant d'être compilé. Le debug, avant, était réalisé au niveau du langage C

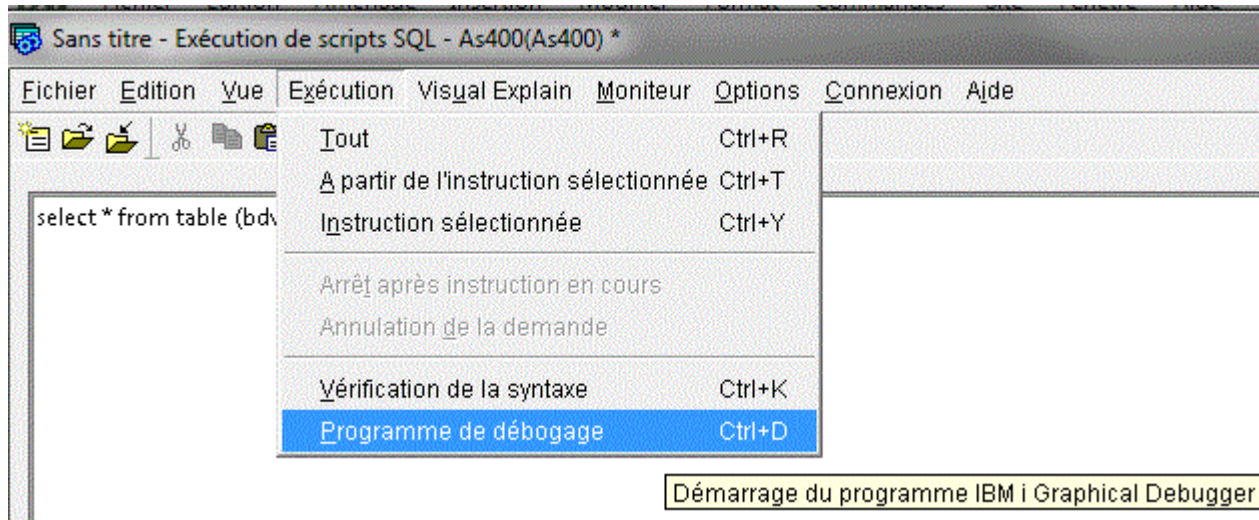
- le mode PAS à PAS devenait compliqué (une instruction SQL peut générer plus de 10 lignes)
- les variables n'étaient visibles que si vous aviez indiqué un LABEL dans le code.
- ce n'est plus le cas , exemple avec la fonction suivante (extrait)

```
CREATE FUNCTION BDVINO.LESVINSCHERS()
RETURNS TABLE(NOM CHARACTER (25), PRODUCTEUR CHARACTER (30), CEPAGE CHARACTER (20) )
LANGUAGE SQL READS SQL DATA
SET OPTION DBGVIEW = *source
BEGIN
DECLARE code int;
```

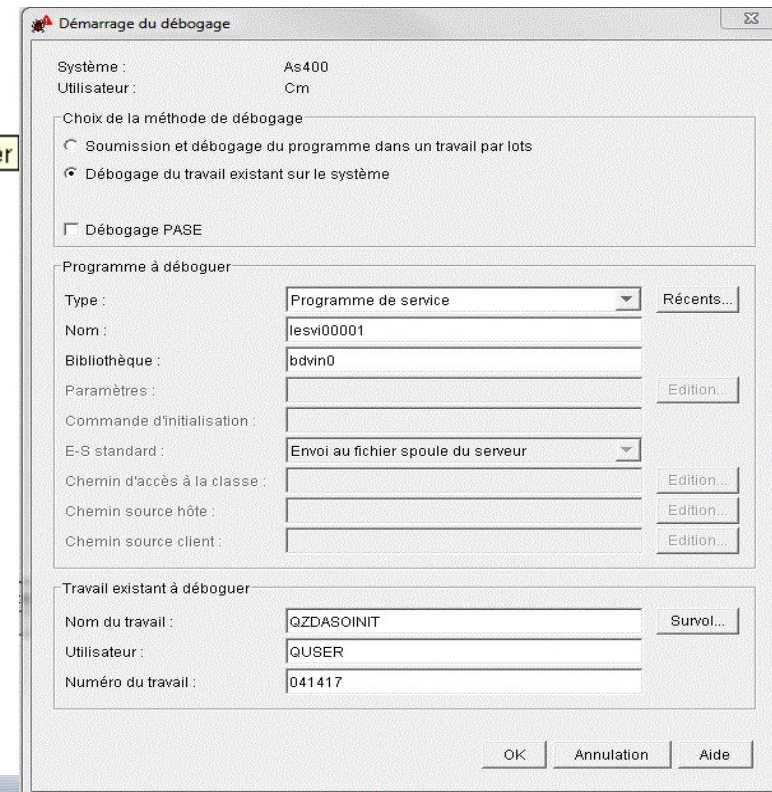


7.2 (TR1)

- Lançons le Debug



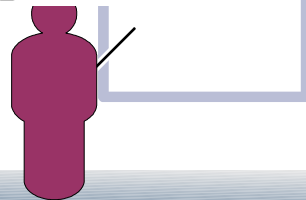
Démarrage du programme IBM i Graphical Debugger



7.2 (TR1)

- Lançons le Debug

```
1 CREATE FUNCTION BDVINO . LESVINSCHERS ( ) RETURNS TABLE ( NOM CHARACTER ( 25 ) , PRO
2     SQL READS SQL DATA SET OPTION DBGVIEW = * SOURCE
3 BEGIN
4 DECLARE CODE INT;
5 DECLARE NOM CHAR ( 25 );
6 DECLARE PR_NOM CHAR ( 30 );
7 DECLARE CEPAGE CHAR ( 20 );
8 DECLARE EOF INT;
9 DECLARE PRIX DEC ( 13 , 2 );
10 DECLARE C1 CURSOR FOR SELECT VIN_CODE , LEFT ( VIN_NOM , 25 ) , LEFT ( PR_NOM , 30 )
11     PRODUCTEURS USING ( PR_CODE );
12 DECLARE CONTINUE HANDLER FOR SQLSTATE '02000'
13 SET EOF = 1;
14 OPEN C1;
15 BCL : LOOP
16 FETCH C1 INTO CODE , NOM , PR_NOM , CEPAGE;
17 IF EOF = 1
18 THEN
19 LEAVE BCL;
20 END IF;
21 SELECT MAX ( CAV_PRXACTUEL ) INTO PRIX FROM BDVINO . MA_CAVE WHERE VIN_CODE = CODE;
```



7.2 (TR1)

- Les variables sont visibles même sans LABEL (en fait un label SQLP_L2 a été ajouté)

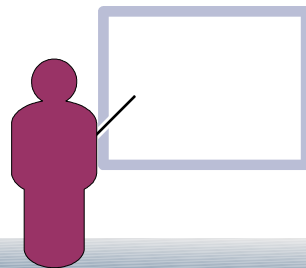
```
20 END IF;  
21 SELECT MAX ( CAV_PRXACTUEL ) INTO PRIX FROM BDVINO . MA_CAVE WHERE VIN_CODE = CODE;  
22 IF PRIX > 20  
23 THEN
```

Variable : SQLP_L2.CODE = 24548

un simple F11 passe à l'instruction SQL suivante

```
20 END IF;  
21 SELECT MAX ( CAV_PRXACTUEL ) INTO PRIX FROM BDVINO . MA_CAVE WHERE VIN_CODE = CODE;  
22 IF PRIX > 20  
23 THEN
```

- Pour profiter de ces nouveautés en mode graphique, il faut installer la SI53809, coté client



7.2 (TR1)

- les messages SQL contenant un nom de variable afficheront désormais, les noms longs de ces dernières plutôt que les noms système
- SQL0190 Attributs de la colonne &3 de &1 dans &2 incompatibles.
- SQL0196 La colonne &3 de &1 dans &2 ne peut pas être supprimée.
- SQL0404 La valeur destinée à la colonne ou la variable &1 est trop longue.
- SQL0406 Erreur de conversion pendant l'affectation d'une valeur à &2.
- SQL0407 Valeurs indéfinies non admises dans la colonne ou la variable &5.
- SQL0415 Opérandes d'opération de définition, de constructeur de tableau ou de VALUES non compatibles.

Complément d'informations sur message

ID message : SQL0404

Message : La valeur destinée à la colonne ou la variable VIN_CEPAGE1 est trop longue.

Cause : Une instruction INSERT, UPDATE, MERGE, SET, VALUES INTO ou GET DIAGNOSTICS indique une valeur de longueur supérieure au maximum autorisé pour une chaîne stockée dans VIN_CEPAGE1. La longueur de VIN_CEPAGE1 est 25 et la longueur de la chaîne 405.

Que faire : Réduisez la longueur de la chaîne de 405 à 25 maximum, puis renouvelez votre demande.

7.2 (TR2)

- Annoncée **très** récemment, probablement disponible 29 Mai
- Hormis les informations vues plus hauts (system limits et DB2 for i Services), voici les nouveautés
- **CREATE OR REPLACE TABLE**

.../...

ON REPLACE PRESERVE ALL ROWS (dff)

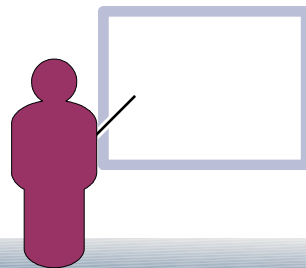
Toutes les lignes sont conservées, des colonnes peuvent être ajoutées ou retirées

ON REPLACE PRESERVE ROWS

Toutes les lignes sont conservées (sauf suppression d'une plage de valeurs sur une table partitionnée) , des colonnes peuvent être ajoutées ou retirées

ON REPLACE DELETE ROWS

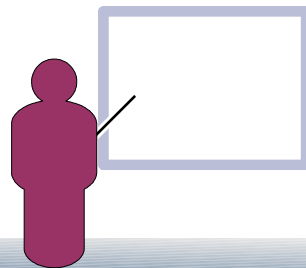
Toutes les lignes sont détruites, les triggers « delete » ne sont pas appelés



7.2 (TR2)

- La procédure `GENERATE_SQL()` génère `CREATE OR REPLACE` sur les tables (l'API et System i Navigator génère des `ALTER TABLE`)
- Les retours chariot sont maintenant préservés par `RUNSQLSTM`, ce qui permet de retrouver le source (par les méthodes ci-dessus) proprement formaté.
- `ALTER TABLE ADD PARTITION` conserve les logiques DDS dépendant de cette table
- Les limites des fonctions SQL sont repoussées
 - De 90 à 1024 paramètres
 - De 125 à 2014 valeurs retournées

De ce dernier chiffre, on doit retirer le nombre de paramètres en entrée.



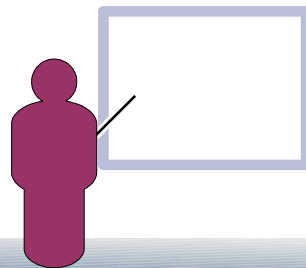
7.2 (TR2)

- Nouvelles fonctions (*pour les UUID*)
 - VARBINARY_FORMAT : convertit chaîne en hexadécimal (binary)
 - VARCHAR_FORMAT_BINARY : convertit héxa en chaîne (formatée)

```
VALUES VARBINARY_FORMAT('D83D6360-1818-11DB-9804-B622A1EF5492',  
                        'XXXXXXXX-XXXX-XXXX-XXXX-XXXXXXXXXXXX')
```

Retourne X'D83D6360181811DB9804B622A1EF5492'

- Index IMO
 - Aujourd'hui un index est utilisé
 - Soit pour réaliser la sélection (Where) ← le plus souvent
 - Soit pour faire le tri à l'affichage (order by)
- Désormais un index contenant les deux critères (zone de sélection PUIS zones de tri) pourra être utilisé (et suggéré) dans les cas suivants :
 - Requête avec comme objectif (OPTIMIZE FOR) *FIRSTIO
 - Table source et Résultat volumineux



7.2 (TR2)

- JSON
- Fonctionnalité fournie en « technology preview »

Stockage du JSON en tant que BLOB

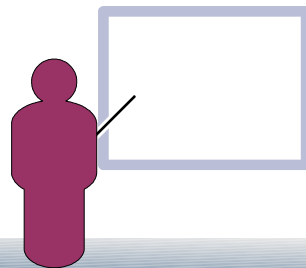
JSON2BSON pour produire un BLOB
BSON2JSON pour extraire

Création de collections JSON sous DB2 (tables à un seul champ BLOB)

Deux interfaces possibles

- DB2 JSON command line
- DB2 JSON API java

Déjà implémenté sous DB2 for Z/OS et LUW.



7.2 (TR2)

- JSON
- En attendant, voyez cet utilitaire sur :

<http://www.mcpressonline.com/programming/techtip-json-and-xml-conversion-in-db2-for-i.html>

Proposant deux fonctions que nous avons testé :

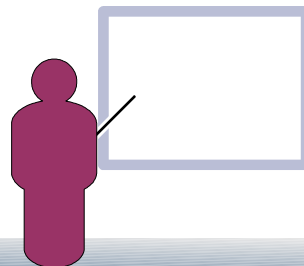
- JSON2XML transforme du JSON en XML
- XML2JSON transforme du XML en JSON

```
values
af4test.xml2json(' <livres><livre><numero>1</numero><titre>essai</titre></livre></livres>')
```

```
00001
{"livres":{"livre":{"titre":"essai","numero":1}}
```

Voyez aussi le projet YAJL (scott Klement) pour traiter du JSON en RPG

<http://www.scottklement.com/yajl/>



7.2 (TR2)

- Plus d'infos sur notre site Volubis.fr, quand nous aurons pu tester (29 Mai)

- Merci

Prochaine Pause-Café : 23 Juin par Internet

Evolutions de l'IBM i (dont TR2) et perspectives

