

Université IBM i 2018

16 et 17 mai

IBM Client Center Paris



Session S01 – Nouveautés DB2 2017 - 2018

Christian Massé

Volubis

cmasse@volubis.fr

Université IBM i 2018

16 et 17 mai



Volubis.fr

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

Dans nos locaux, vos locaux ou par Internet

Université IBM i 2018

16 et 17 mai

thirty
years

Rappel : accès à tous nos cours (500+), toutes nos vidéos (20+) pour 1600€

RECHERCHE RAPIDE : (utilisez un ou deux mots-clés, reliés par et/ou, et par défaut)

- Par thèmes
- Par cursus
 - Dev. sur IBM i
 - Liste des cursus
 - Liste des outils
- Initiation
- Perfectionnement
- Versions
 - IBMi 7.3
 - IBMi 7.2
 - IBMi 7.1
 - IBMi V6.10
 - I5/OS V5R40
 - I5/OS & OS/400 V5R30
 - OS/400 V5R20
 - OS/400 V5R10
 - OS/400 V4R50

PAR CURSUS

PAR MOT-CLÉ:

PAR MODULE/SUJET

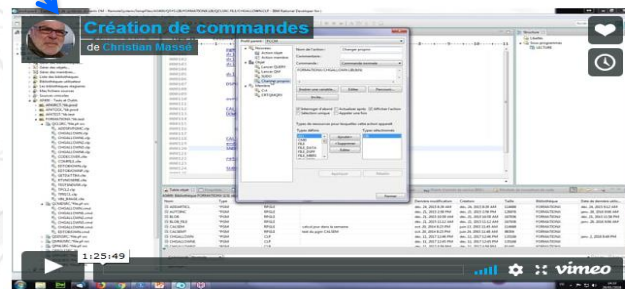
Voyez aussi

PRÉSENTATIONS, CLIENTS ACCESS, RDI

LES COURS SUR L'IBM I, SERVEUR WEB

29 Janvier 2018

*CMD : création de commandes (voyez le PDF)



Télécharger les sources

29 Novembre 2017

Java & RPG : JDBCR4 (voyez le PDF)



Nouveautés DB2 2017 - 2018



- JSON
 - JSON_TABLE
 - TR3
- Syntaxe SQL
- Limit et OFFSET
- SQL as a service
- ACS

A background network diagram consisting of numerous grey circular nodes connected by thin grey lines, forming a complex web-like structure. The nodes are distributed across the entire page, with a higher density in the center and corners.

JSON

Nouveautés DB2 2017 - 2018

■ TR1 : JSON_TABLE

(Exemple à partir de https://data.nantes.fr/api/publication/24440040400129_NM_VDN_00083/JARDINS_FAMILIAUX_VDN_STBL/content/?format=json)

```
72 --Jardins familiaux de la ville de Nantes
73 select * from JSON_TABLE(
74   SYSTOOLS.HTTPGETCLOB(
75     'http://data.nantes.fr/api/publication/24440040400129_NM_VDN_00083/JARDINS_FAMILIAUX_VDN_STBL/content/?format=json'
76     ,null), '$.data[*]')
77   COLUMNS( ASSO CHAR(35) PATH '$.ASSOCIATION',
78             ADRESSE VARCHAR(30) PATH '$.ADRESSE',
79             PARCELLES VARCHAR(25) PATH '$.NBRE_PARCELLES'
80           ) AS X;
81
82
```

ASSO	ADRESSE	PARCELLES
Asso " Les jardins de la Roche"	CHEMIN DE LA ROCHE	17
Asso " Les jardins de la Cressonnière	RUE DES RENARDS	10
Asso. "Des petits jardins du Lait d	RUE EMILE PEHANT	1
Confédération Syndicale des Famille	RUE SUZANNE LENGLEN	1
Asso.Jardins Familiaux Nantais	BD PIERRE DE COUBERTIN	14
Asso "les Jardins des Collines"	CHEMIN DES COLLINES	31
Asso. "Jardins de la Marrière "	RUE DU CROISSANT	90
Asso."la crapaudine"	AVENUE DES GOBELETS	92
ECOS	RUE D'HENDAYE	1

Nouveautés DB2 2017 - 2018



■ JSON_TABLE

JSON_TABLE(
 JSON_SOURCE
 JSON_PATH
COLUMNS nom type PATH 'json_path', ...) as X

JSON_SOURCE

- Flux JSON, peut-être une variable
- une colonne base de données (JSON ou BSON dans du VARCHAR)
- un flux (HTTPGETCLOB, par exemple)

Nouveautés DB2 2017 - 2018



■ JSON_TABLE

JSON_TABLE(
 JSON_SOURCE
 JSON_PATH
COLUMNS nom type PATH 'json_path', ...) as X

JSON_PATH

- \$ objet en cours
- . élément dans l'objet en cours
- *Nom* nom de l'élément
- [] élément dans un tableau

```
1 CREATE VARIABLE JSON_VAR VARCHAR(2000);
2 SET JSON_VAR='{
3     "id" : 901,
4     "name" : { "first":"John", "last":"Doe" },
5     "phones" : [{"type":"home", "number":"555-3762"},
6                 | {"type":"work", "number":"555-7252"}]
7     }';
8
9 SELECT * FROM JSON_TABLE(JSON_VAR,
10 '$'
11 COLUMNS(
12     id VARCHAR(10) PATH 'lax $.id',
13     first VARCHAR(10) PATH 'lax $.name.first',
14     last VARCHAR(10) PATH 'lax $.name.last' )
15 ) as t;
```

ID	FIRST	LAST
901	John	Doe

Nouveautés DB2 2017 - 2018



■ JSON_TABLE

... COLUMNS nom type-SQL PATH 'json_path', ...) as X

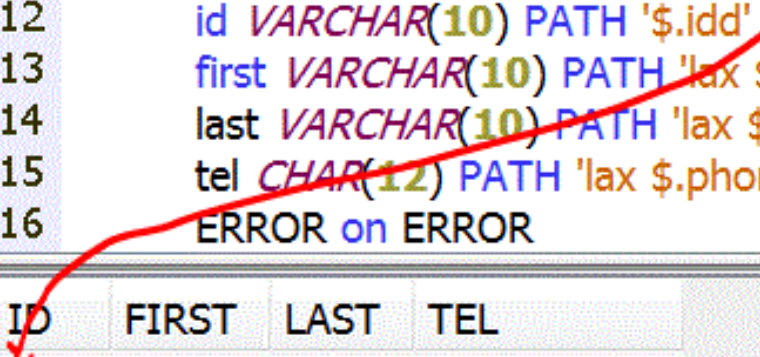
- un nom de colonne
- un type : CHAR(x) , DEC(x , y), etc...
- PATH
 - une règle d'utilisation
 - **lax**, utilisation souple
 - on peut faire référence à un élément, même quand c'est un tableau, l'itération est alors automatique
 - on peut faire référence à un tableau, même quand c'est un élément
 - les cas impossibles génèrent valeur nulle
 - **strict**, utilisation stricte, les cas précédents génèrent une erreur
 - L'élément JSON dont il faut extraire la valeur
 - (options)
 - NULL ON EMPTY
retourne valeur nulle sur un élément manquant
 - ERROR ON EMPTY
retourne une **erreur** sur un élément manquant
 - DEFAULT <une-valeur> ON EMPTY
retourne une valeur par défaut sur un élément manquant
 - ERROR ON ERROR
retourne une erreur en cas d'erreur (SQL16410)
 - DEFAULT <une-valeur> ON ERROR
retourne une valeur par défaut suite à une erreur

Nouveautés DB2 2017 - 2018

- JSON_TABLE

... COLUMNS nom type PATH 'json_path', ...) as X

```
9 SELECT * FROM JSON_TABLE(JSON_VAR,  
10 '$'  
11 COLUMNS(  
12   id VARCHAR(10) PATH '$.idd' NULL on EMPTY,  
13   first VARCHAR(10) PATH 'lax $.name.first',  
14   last VARCHAR(10) PATH 'lax $.name.last',  
15   tel CHAR(12) PATH 'lax $.phones[0].number')  
16 ERROR on ERROR
```



ID	FIRST	LAST	TEL
-	John	Doe	555-3762

Notez l'utilisation d'un tableau

Nouveautés DB2 2017 - 2018



- RPG aussi (quoique...)
 - Nouveau code-opération Data-Into
 - Utilise un parser... à écrire
 - IBM fournit un exemple pour le format JSON
 - Ce nouveau code opération est reconnu par RDI 9.6.0.2

```
jsonString =
  '{
+   "petInfo":'
+   '{
+     "a?b" : "a dash b",'
+     "pets":'
+     '['
+       '{"name":"Spot", "type":"dog", "age":3, "faveToy":"ball"},'
+       '{"name":"Puff", "type":"cat", "age":7, "faveToy":"string"}'
+     ],'
+     "veterinarian":"Dr Smith"
+   }'
+ '};'

data-into petInfo
  %data(jsonString
        : 'case=any countprefix=num allowextra=yes')
  %parser('AF4TEST/JSONPARSE');
for i = 1 to petInfo.numPets;
  msg = petInfo.pets(i).name + ' ' + petInfo.pets(i).type + ' '
        + 'age=' + %char(petInfo.pets(i).age);
  dsply msg;
endfor;
msg = 'Vet is ' + petInfo.veterinarian;
dsply msg;
return;
```

Nouveautés DB2 2017 - 2018



- Revenons à SQL, TR2 : Prédicat IS (not) JSON

The screenshot shows a DB2 SQL script editor window titled 'Untitled* - Run SQL Scripts - AS400(As400)'. The menu bar includes 'Fichier', 'Edition', 'Affichage', 'Run', 'VisualExplain', 'Options', and 'Connection'. The toolbar contains various icons for file operations and execution. The main area displays a SQL query:

```
1 select * from posample.customerj
2 where info is JSON
```

Below the query, the results are displayed in a table with two columns: 'CID' and 'INFO'. The results are as follows:

CID	INFO
4	{"customerinfo":{"addr":{"pcode-zip":44000,"street":"Place Royale"...
1000	{"customerinfo":{"addr":{"pcode-zip":"M6W 1E6","street":"5 Rosew...
1002	{"customerinfo":{"addr":{"pcode-zip":"Z9Z 2P2","street":"1150 Map...
1003	{"customerinfo":{"addr":{"pcode-zip":"N8X 7F8","street":"1596 Bas...
1004	{"customerinfo":{"addr":{"pcode-zip":44000,"street":"Place Royale"...
1005	{"customerinfo":{"addr":{"pcode-zip":123456,"street":"1596 Sunset...
1006	{"customerinfo":{"addr":{"pcode-zip":44470,"street":"5 rue du TER...
1111	{"customerinfo":{"addr":{"pcode-zip":"M6W 1E6","street":"5 Rosew...

Nouveautés DB2 2017 - 2018



- Prédicat IS (not) JSON, exemple d'utilisation

```
1 values CASE
2 when '{"producteur":{"appellation":"muscadet","nom":"Bruno Cormerais"}}' IS JSON then 1
3 else 0
4 END|
```

00001

1

```
1 values CASE
2 when '<{"producteur":{"appellation":"muscadet","nom":"Bruno Cormerais"}}>' IS JSON then 1
3 else 0
4 END|
```

00001

0

Nouveautés DB2 2017 - 2018



- Prédicat JSON_EXISTS

```
1 values CASE
2 when JSON_EXISTS('{ "producteur":{ "appellation": "muscadet", "nom": "Bruno Cormerais" } }' ,
3 'lax $.producteur.nom') then 1
4 else 0 END
```

00001
1

```
1 values CASE
2 when JSON_EXISTS('{ "producteur":{ "appellation": "muscadet", "nom": "Bruno Cormerais" } }' ,
3 'lax $.producteur.tel') then 1
4 else 0 END
```

00001
0

Nouveautés DB2 2017 - 2018



- TR3 : Accès aux données JSON
 - JSON_VALUE Retourne une valeur scalaire (unitaire)

```
1 CREATE VARIABLE JSON_VAR VARCHAR(2000);
2 SET JSON_VAR='{
3     "id" : 901,
4     "name" : { "first":"John", "last":"Doe" },
5     "phones" : [{"type":"home", "number":"555-3762"},
6                 {"type":"work", "number":"555-7252"}]
7 }';
8
9
10 values (json_value(JSON_VAR, '$.id' RETURNING integer));
11
```

00001

901

Nouveautés DB2 2017 - 2018



- TR3 : Accès aux données JSON
 - JSON_QUERY Retourne une Chaîne JSON (un extrait)

```
1 CREATE VARIABLE JSON_VAR VARCHAR(2000);
2 SET JSON_VAR='{
3     "id" : 901,
4     "name" : { "first":"John", "last":"Doe" },
5     "phones" : [{"type":"home", "number":"555-3762"},
6                 {"type":"work", "number":"555-7252"}]
7 }';
8
9
10 values (json_query(JSON_VAR, '$.name'));
```

00001
{"first":"John","last":"Doe"}

Nouveautés DB2 2017 - 2018



- TR3 : Publier du JSON
 - JSON_ARRAY

```
VALUES(JSON_ARRAY((SELECT DEPTNO FROM DEPT
                    WHERE DEPTNAME LIKE 'BRANCH OFFICE%'))));
```

--Produit un tableau de valeur (rappel entre [et])

==> ["F22","G22","H22","I22","J22"]

```
1 VALUES(JSON_ARRAY((SELECT DEPTNO FROM db2sample.DEPT
2                     WHERE DEPTNAME LIKE 'BRANCH OFFICE%'))));
```

00001

["F22","G22","H22","I22","J22"]

Nouveautés DB2 2017 - 2018



- TR3 : Publier du JSON
 - JSON_OBJECT

```
SELECT JSON_OBJECT('Nom' : LASTNAME,  
                   'date naissance' : HIREDATE,  
                   'Salaire' SALARY) FROM EMPLOYEE WHERE EMPNO = '000020';
```

--produit un objet JSON, chaque élément a un nom et une valeur

```
==> {"Nom":"THOMPSON","date naissance":"1973-10-10","Salaire":41250.00}
```

```
29  
30 SELECT JSON_OBJECT('Nom' : LASTNAME,  
31                   'date naissance' : HIREDATE,  
32                   'Salaire' : SALARY) FROM DB2SAMPLE.EMPLOYEE WHERE EMPNO = '000020';  
33 |
```

00001

```
{"Nom":"THOMPSON","date naissance":"10-10-73","Salaire":41250.00}
```

Nouveautés DB2 2017 - 2018

- TR3 : Publier du JSON
 - JSON_ARRAYAGG agrégation de valeurs (tableau)

```
116 -- avec notre base vinicole
117 SELECT JSON_OBJECT('Producteur' VALUE trim(pr_nom),
118                  'Ville' VALUE trim(pr_commune),
119                  'Appellation' VALUE trim(appellation),
120                  'Les-vins' VALUE
121                    JSON_ARRAYAGG(
122                      JSON_OBJECT('Vin' VALUE trim(vin_nom),
123                                'Cepage' VALUE trim(vin_cepage1))
124                      ORDER BY vin_nom)
125 FROM bdvin1.producteurs P JOIN bdvin1.vins V ON P.pr_code = V.pr_code
126      JOIN bdvin1.appellations A ON p.Appel_code = A.appel_code
127 GROUP BY pr_nom, pr_commune, appellation;
```

Ici, tous les vins d'un producteur

00001

```
{
  "Producteur": "A l'Ancienne Forge",
  "Ville": "Mittelbergheim",
  "Appellation": "Alsace",
  "Les-vins": [
    {
      "Vin": "Alsace sylvaner VV",
      "Cepage": "Sylvaner"
    },
    {
      "Vin": "Pinot noir",
      "Cepage": "Pinot noir"
    }
  ]
}
```

```
{
  "Producteur": "A Maccia",
  "Ville": "Ranzo",
  "Appellation": "Riviera ligure di ponente",
  "Les-vins": [
    {
      "Vin": "Riviera Ligure di Ponente Pigato",
      "Cepage": null
    },
    {
      "Vin": "Riviera Ligure di Ponente Rossese",
      "Cepage": "Rossese"
    }
  ]
}
```

```
{
  "Producteur": "A. Bichot & Cie",
  "Ville": "Beaune",
  "Appellation": "Beaune",
  "Les-vins": [
    {
      "Vin": "Beaune",
      "Cepage": "Pinot noir"
    },
    {
      "Vin": "Clos de Vougeot",
      "Cepage": "Pinot noir"
    },
    {
      "Vin": "Corton Charlemagne",
      "Cepage": "Pinot noir"
    }
  ]
}
```

Nouveautés DB2 2017 - 2018



- TR3 : Publier du JSON
 - JSON_OBJECTAGG agrégation d'objets

```
129 -- avec notre base vinicole
130 SELECT JSON_OBJECT('Producteur' VALUE trim(pr_nom),
131                   'Ville' VALUE trim(pr_commune),
132                   'Appellation' VALUE trim(appellation),
133                   'Les-vins' value JSON_OBJECTAGG(char(vin_code) VALUE trim(vin_nom)))
134 FROM bdvin1.producteurs P JOIN bdvin1.vins V ON P.pr_code = V.pr_code
135                   JOIN bdvin1.appellations A ON p.Appel_code = A.appel_code
136 GROUP BY pr_nom, pr_commune, appellation;
```

00001

```
{"Producteur":"A l'Ancienne Forge","Ville":"Mittelbergheim","Appellation":"Alsace","Les-vins":{"21007 ":"Alsace sylvaner VV","21008 ":"Pinot noir"}}
{"Producteur":"A Maccia","Ville":"Ranzo","Appellation":"Riviera ligure di ponente","Les-vins":{"12433 ":"Riviera Ligure di Ponente Pigato","12434 ":"Rivier
```

Nouveautés DB2 2017 - 2018



■ TR3 : Publier du JSON

• Options

- FORMAT JSON | BSON

Indique que la data est déjà formatée :

évite les caractères d'échappement \ devant { et [

- Order BY

tri la donnée lors de l'agrégation

- ABSENT on NULL (*dft*)
- NULL on NULL

- RETURNING CLOB(2G) CCSID 1208 FORMAT JSON (*dft*)
- RETURNING CHAR/VARCHAR...ENCODING UTF-8 | UTF16
- RETURNING BLOB

Nouveautés DB2 2017 - 2018



- TR3 : Publier du JSON, Exemple

```
35 with TEMP as (  
36   select Json_object('Producteur' value trim(pr_nom),  
37                    'Ville' value trim(pr_commune),  
38                    'Appellation' value trim(appellation),  
39                    'Les-Vins' value  
40                      JSON_OBJECTAGG(  
41                        char(vin_code) value trim(vin_nom)  
42                      )  
43                    ) as JSON_DATA  
44   from bdvin1.producteurs P join bdvin1.vins V on P.pr_code = v.pr_code  
45                        join BDVIN1.appellations A on P.appel_code = A.appel_code  
46   group by pr_nom, pr_commune, appellation)  
47 SELECT json_object('Les-producteur' value json_ARRAYAGG(JSON_DATA format JSON)) from TEMP;
```

00001

```
{\"Les-producteur\": [{\"Producteur\": \"A l'Ancienne Forge\", \"Ville\": \"Mittelbergheim\", \"Appellation\": \"Alsace\", \"...\"}]}
```

Nouveautés DB2 2017 - 2018

- TR3 : Publier du JSON
 - Rappel le résultat d'un Select peut aller dans un Fichier
 - Revoyez les CLOB et CLOB_FILE

Sur notre page « freeware »



Nouveautés DB2 2017 - 2018

■ Exemple

```
**free
dcl-s fichierJSON SQLTYPE(CLOB_FILE);
fichierjson_name = '/temp/producteurs.json';
fichierjson_nl = %len(%trim(fichierjson_name));
fichierjson_fo = SQFOVR;

exec sql
with temp
as (select json_object('Producteur' value trim(pr_nom),
                    'Ville' value trim(pr_commune),
                    'Appellation' value trim(appellation),
                    'Les-Vins' value
                    JSON_OBJECTAGG(char(vin_code) value
                    trim(vin_nom)))

as JSON_DATA
from bdvin1.producteurs P
join bdvin1.vins V on P.pr_code = v.pr_code
join BDVIN1.appellations A on P.appel_code = A.appel_code
group by pr_nom,
         pr_commune,
         appellation)
select json_object('Les-producteur' value json_ARRAYAGG(JSON_DATA
format JSON))
into :fichierjson
from temp;
```

```
*inlr = *ON;
```

Browse : /temp/producteurs.json
Record : 1 of 10233 by 18
Column : 1 132 by 131
Control : █

```
+...1...+...2...+...3...+...4...+...5...+...6...+...7...+...8...+...9...+...0...+...1...
*****Beginning of data*****
{"Les-producteur":[{"Producteur":"A l'Ancienne Forge","Ville":"Mittelbergheim","Appellation":"Alsace","Les-Vins":
ace sylvaner vv","21008":"Pinot noir"}],{"Producteur":"A Maccia","Ville":"Ranzo","Appellation":"Riviera lig
s-Vins":{"12433":"Riviera Ligure di Ponente Pigato","12434":"Riviera Ligure di Ponente Rossese"}},{"P
ot & Cie","Ville":"Beaune","Appellation":"Beaune","Les-Vins":{"18487":"Beaune","18837":"Clos de Vouge
Corton Charlemagne","18840":"Corton-Bressandes","18977":"Mazis-Chambertin","18836":"Meursault",
rchet","18905":"Musigny","18903":"Puligny-Montrachet","18978":"Romanée-Saint-Vivant","18838
"}}, {"Producteur":"A. Cochet et Fils","Ville":"Mareuil","Appellation":"Beaune","Les-Vins":{"14757":"
"}}
```




Syntaxe SQL

Nouveautés DB2 2017 - 2018



- TR1 : Fonctions OLAP
 - **PERCENT_RANK** affiche le pourcentage du rang (de 0 à 100 %)

```
37 with temp as (  
38   select year(logtime) as annee, count(*) as nombre  
39   from httplog.httplog where year(logtime) < 2017 -- 2017 non significatif  
40   group by year(logtime)  
41 )  
42 select annee, nombre, percent_rank() over (order by nombre desc) * 100 as PC_rang  
43 from temp;
```

ANNEE	NOMBRE	PC_RANG
2016	12080356	0
2015	7895074	25.00
2014	6815648	50.0
2013	6245919	75.00
2012	3903384	100

Nouveautés DB2 2017 - 2018



■ TR1 : SQL PSM

- en SQL PSM vous pouvez désormais inclure du code SQL **ou C**

INCLUDE [SQL] bibliothèque/fichier(membre)

ou

INCLUDE [SQL] membre

Si c'est du C, utilisez l'Api Qp0zLprintf pour écrire dans la JOBLOG

- en SQL PSM, vous pouvez utiliser SET OPTION pour indiquer des options destinées au compilateur

BINDOPT permet désormais d'indiquer des options de liage (BNDDIR par exemple)

INCFILE, le nom du fichier source à inclure

Nouveautés DB2 2017 - 2018



■ TR1 : Divers

- SYSPARTITIONSTAT est un élément du catalogue SQL et contient la liste des membres des fichiers DB2 (ou sources)

Il intègre désormais **PARTITION_TEXT** VARGRAPHIC(50) CCSID 1200, contenant le texte du membre.

- La fonction SQL **REPLACE** évolue :

elle admettait trois paramètres

REPLACE(source, chaine-recherchée, chaine-remplacement)

elle peut désormais être utilisée avec deux paramètres uniquement

REPLACE(source, chaine-recherchée)

La chaîne recherchée est tout simplement **supprimée**

Nouveautés DB2 2017 - 2018



■ TR1 : Divers

- Jusqu'à maintenant une fonction pouvait être indiquée :

Non Déterministe (NOT DETERMINISTIC)

ne retournant pas toujours la même valeur (NOW, par exemple)

- Déterministe (DETERMINISTIC)

Retournant toujours la même valeur (PI=3,1416, par exemple)

- Désormais vous avez le choix entre :

- GLOBAL DETERMINISTIC (comme avant)

- STATEMENT DETERMINISTIC

Retournant la même valeur pendant la durée de la requête (taux de change, par exemple)

dans ce dernier cas, DB2 ne rappelle pas la fonction lors d'une même requête.

Nouveautés DB2 2017 - 2018



■ TR2 : Embedded SQL & RPG4

le source généré par le pré-compilateur SQL (type SQLRPGLE) est enfin, en format libre :

```
.....
104 **FREE
105 /** Exec sql
106 /** select pr_nom, pr_commune, ifnull(pr_tel, '-'),
107 /** appellation, count(*) into :unproducteur
108 /** from producteurs P join appellations A
109 /**     on P.appel_code = A.appel_code
110 /**     join vins V
111 /**     on P.pr_code = V.pr_code
112 /** where P.pr_code = :code
113 /** group by pr_nom, pr_commune, ifnull(pr_tel, '-'), appellation;

114         SQL_00005 = CODE;                               //SQL
115         SQLER6 = -4;                                     //SQL 2
116         SQLROUTE_CALL(                                  //SQL
117             SQLCA                                       //SQL
118             : SQL_00000                                  //SQL
119         );                                             //SQL
120         IF SQL_00003 = '1';                             //SQL
121         EVAL UNPRODUCTEUR.NOM = SQL_00006;             //SQL
770WDS V7R3M0 160422 RN      IBM ILE RPG      QTEMP/WSPRODVINS      VOLUBIS      10

<----- Sp,cifications source -----
gne .....1.....2.....3.....4.....5.....6.....7.....8.....
122         EVAL UNPRODUCTEUR.VILLE = SQL_00007;         //SQL
123         EVAL UNPRODUCTEUR.TEL = SQL_00008;          //SQL
124         EVAL UNPRODUCTEUR.APPELLATION = SQL_00009;   //SQL
125         EVAL UNPRODUCTEUR.NBVINS = SQL_00010;       //SQL
126         ENDIF;                                       //SQL
127 //
```

Nouveautés DB2 2017 - 2018



■ TR2 : Fonction LISTAGG

- place "à l'horizontale" toutes les occurrences d'une expression pour un groupe (agrégation)

```
1 select pr_nom, listagg(vin_code, ' , ')
2 from bdvin1.producteurs join bdvin1.vins using(pr_code)
3 group by pr_nom
```

PR_NOM	00002
von Mumm'sches Weingut	15710 , 15711 , 15712 , 15713 , 15714
weingut august eser	15599 , 15600 , 15601 , 15602 , 15603 , 15604
weingut okonomierat rebholz	15493 , 15494 , 15495 , 15496 , 15497 , 15498 , 15499
weingut schloss reinhartshausen	15744 , 15745 , 15746 , 15747 , 15748 , 15749
weingut winfried frey und sohne	15360 , 15361 , 15362 , 15363 , 15364 , 15365 , 15366 , 15367

Nouveautés DB2 2017 - 2018



- TR3 : Gestion des droits (DCL)
- Nouvelles options de gestion des droits sur une bibliothèque
- GRANT **CREATEIN** on schema BIB1 to USER1 : droit de création d'objets
 - => *OBJOPR + *READ + *ADD + *EXECUTE
- GRANT **USAGE** on schema BIB1 to USER2 :
 - droit d'utilisation => *USE
- GRANT **ALL** on schema BIB1 to USER3
 - => les deux
- même chose sur REVOKE...



LIMIT & OFFSET

Nouveautés DB2 2017 - 2018



■ Limit et Offset

■ Support étendu

- Accepté dans les SELECT imbriqués
- Accepté dans les vues (select imbriqués uniquement)
- Accepté sur le 5250 (la requête est entièrement ré-exécutée)

```
> select * from httplog.httplog  
order by logtime limit 10 offset 1  
Instruction SELECT exécutée.
```

```
===> select * from httplog.httplog  
order by logtime limit 10 offset 11
```

F3=EXIT F4=Invite F6=Insérer ligne F9=Rappel F10=Copier
F12=Annuler F13=Services F24=Autres
Requête en cours. 2408447 enregistrements extraits, 2408447 traités.

14 * * * * * X SYSTEM

Nouveautés DB2 2017 - 2018



■ Limit et Offset

- Peuvent être le résultat d'une expression

```
> create variable qgpl.page_number int
La variable PAGE_NUMBER a été créée dans QGPL.
> values 1 into page_number
Instruction VALUES exécutée.
> select * from bdvin1.vins
limit 10 offset (page_number * 10) + 1
Instruction SELECT exécutée.
```

Nouveautés DB2 2017 - 2018



- Limit et Offset
- Peuvent être intégrés à UPDATE et DELETE avec **Order BY**

```
-- supprimer les 10 plus anciennes commandes  
DELETE FROM Commandes where nocli = 45 order by DATCMD LIMIT 10
```

Exemple

```
> delete from httplog.httplog  
  order by logtime limit 100  
  100 lignes supprimées de HTTPLOG dans HTTPLOG.  
--- ■
```

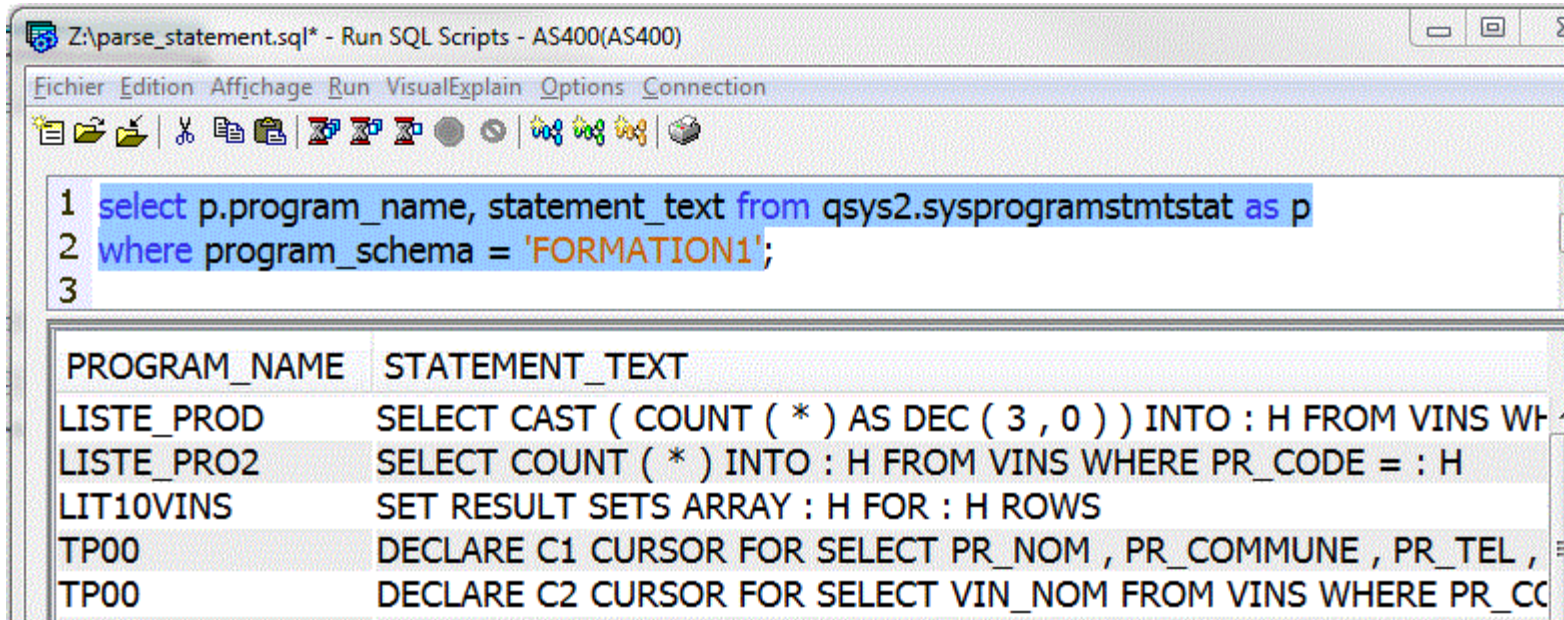


DB2 for i services

Nouveautés DB2 2017 - 2018

- **PARSE_STATEMENT**

Le fichier catalogue QSYS2.SYSPROGRAMSTMTSTAT donnait déjà la liste des instructions SQL



The screenshot shows a DB2 SQL script editor window titled "Z:\parse_statement.sql* - Run SQL Scripts - AS400(AS400)". The menu bar includes "Fichier", "Edition", "Affichage", "Run", "VisualExplain", "Options", and "Connection". The toolbar contains various icons for file operations and execution. The script content is as follows:

```
1 select p.program_name, statement_text from qsys2.sysprogramstmtstat as p
2 where program_schema = 'FORMATION1';
3
```

Below the script, the results are displayed in a table with two columns: PROGRAM_NAME and STATEMENT_TEXT.

PROGRAM_NAME	STATEMENT_TEXT
LISTE_PROD	SELECT CAST (COUNT (*) AS DEC (3 , 0)) INTO : H FROM VINS WH
LISTE_PRO2	SELECT COUNT (*) INTO : H FROM VINS WHERE PR_CODE = : H
LIT10VINS	SET RESULT SETS ARRAY : H FOR : H ROWS
TP00	DECLARE C1 CURSOR FOR SELECT PR_NOM , PR_COMMUNE , PR_TEL ,
TP00	DECLARE C2 CURSOR FOR SELECT VIN_NOM FROM VINS WHERE PR_CC

Nouveautés DB2 2017 - 2018



■ PARSE_STATEMENT

PARSE_STATEMENT permet de retrouver la liste des informations manipulées (Tables, colonnes, curseurs)

```
1 select p.program_name, s.* from qsys2.sysprogramstmtstat as p ,
2 TABLE(qsys2.parse_statement(statement_text)) as s
3 where program_schema = 'FORMATION1'
```

PROGRAM_NAME	NAME_TYPE	NAME	SCHEMA	RDB	COLUMN_NAME	USAGE_TYPE	NAME_START_POSITION	SQL_STATEMENT_
LISTE_PROD	TABLE	VINS	-	-	-	QUERY	60	QUERY
LISTE_PROD	COLUMN	-	-	-	PR_CODE	QUERY	71	QUERY
LISTE_PRO2	TABLE	VINS	-	-	-	QUERY	34	QUERY
LISTE_PRO2	COLUMN	-	-	-	PR_CODE	QUERY	45	QUERY
TP00	COLUMN	-	-	-	PR_NOM	QUERY	30	DECLARE CURSOR
TP00	COLUMN	-	-	-	PR_COMMUNE	QUERY	39	DECLARE CURSOR

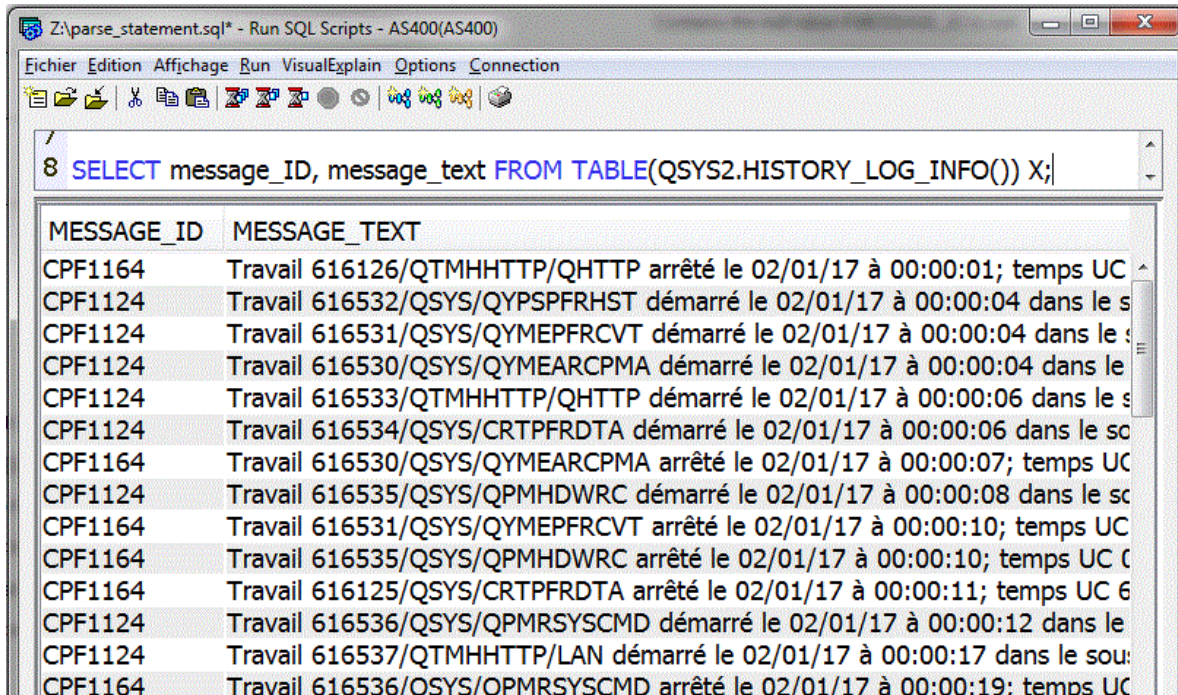
A background graphic consisting of a complex network of interconnected nodes and lines, resembling a web or a data network. The nodes are represented by small grey circles, and the lines are thin, light grey lines connecting these nodes. The overall pattern is dense and irregular, filling the entire page.

IBM i services

Nouveautés DB2 2017 - 2018

- HISTORY_LOG_INFO

Affiche l'équivalent de DSPLOG



Nouveautés DB2 2017 - 2018



■ JOB_INFO

Affiche les travaux suivant les critères indiqués

```
>>-JOB_INFO--(----->
>+-----*ALL--+----->
  '+-----+-----status-'
    '-JOB_STATUS_FILTER--=>-'
>+-----*ALL--+----->
  '-,+-----+-----type-'
    '-JOB_TYPE_FILTER--=>-'
>+-----*ALL-----+>
  '-,+-----+-----sous-système-'
    '-JOB_SUBSYSTEM_FILTER--=>-'
>+-----USER-----+>
  '-,+-----+-----utilisateur-'
    '-JOB_USER_FILTER-- => -'
>+-----*ALL-----+----).
  '-,+-----+-----travail-'
    '-JOB_SUBMITTER_FILTER-- => -'
```

Nouveautés DB2 2017 - 2018



■ JOB_INFO

Affiche les travaux suivant les critères indiqués

```
11 select * from table(QSYS2.JOB_INFO(  
12 JOB_SUBSYSTEM_FILTER => 'QHHTTPSVR',  
13 JOB_USER_FILTER => '*ALL'  
14 )) as x;
```

JOB_NAME	JOB_INFORMATION	JOB_STATUS
617091/QTMHHTTP/VOLUBIS	YES	ACTIVE
617093/QTMHHTTP/VOLUBIS	YES	ACTIVE
617094/QTMHHTTP/VOLUBIS	YES	ACTIVE
612516/QTMHHTTP/WFORMATION	YES	ACTIVE
612264/QTMHHTTP/ADMIN	YES	ACTIVE
617315/QTMHHTTP/WSERVICE1	YES	ACTIVE
612233/QTMHHTTP/LAN	YES	ACTIVE
617095/QTMHHTTP/VOLUBIS	YES	ACTIVE
617100/QTMHHTTP/VOLUBIS	YES	ACTIVE
617314/QTMHHTTP/WSERVICE1	YES	ACTIVE
617317/QWSERVICE/WSERVICE1	YES	ACTIVE
617530/QTMHHTTP/VOLUBIS	YES	ACTIVE
617318/QTMHHTTP/WSERVICE1	YES	ACTIVE

Nouveautés DB2 2017 - 2018



■ MESSAGE_QUEUE_INFO

Affiche les messages d'une message Queue

```
1 select MESSAGE_TEXT from QSYS2.MESSAGE_QUEUE_INFO
2  WHERE MESSAGE_QUEUE_NAME = 'QSYSOPR'
```

MESSAGE_TEXT

IPL exécuté à partir de la zone machine ££MACH£B.

Système de fichier Slash (/) monté.

Système de fichier /QOpenSys monté.

Système de fichier /QDLS monté.

Système de fichier /QSYS.LIB monté.

Système de fichier /QOPT monté.

Système de fichier /QFileSvr.400 monté.

Système de fichier /QNTC monté.

Système de fichier /dev/QASP01 monté.

Erreur pendant l'application ou le retrait de PTF au ...

Nouveautés DB2 2017 - 2018



■ OBJECT_PRIVILEGES

Affiche les droits par objet

```
1 SELECT AUTHORIZATION_NAME, OBJECT_AUTHORITY
2 FROM QSYS2.OBJECT_PRIVILEGES
3 WHERE SYSTEM_OBJECT_SCHEMA = 'BDVIN1' AND
4      OBJECT_TYPE = '*FILE'
```

AUTHORIZATION_NAME	OBJECT_AUTHORITY
*PUBLIC	*CHANGE
QSECOFR	*ALL
FORMATION1	*ALL
CM	*ALL
*PUBLIC	*CHANGE

Nouveautés DB2 2017 - 2018



- **AUTHORIZATION_LIST_INFO**

Affiche les Objets protégés par une liste d'autorisation

```
1 SELECT * FROM QSYS2.AUTHORIZATION_LIST_INFO
2 WHERE AUTHORIZATION_LIST = 'QPMCCDATA';
```

AUTHORIZATION_LIST	SYSTEM_OBJECT_SCHEMA	SYSTEM_OBJECT_NAME	SYSTEM_OBJECT_TYPE
QPMCCDATA	QPFRDATA	QAPMAPPN	*FILE
QPMCCDATA	QPFRDATA	QAPMBUS	*FILE
QPMCCDATA	QPFRDATA	QAPMCIOP	*FILE
QPMCCDATA	QPFRDATA	QAPMCONF	*FILE
QPMCCDATA	QPFRDATA	QAPMDIOP	*FILE
QPMCCDATA	QPFRDATA	QAPMDISK	*FILE
QPMCCDATA	QPFRDATA	QAPMDISKRB	*FILE

Nouveautés DB2 2017 - 2018



- AUTHORIZATION_LIST_USER_INFO

Affiche les utilisateurs d'une liste d'autorisation

```
1 SELECT *
2 FROM QSYS2.AUTHORIZATION_LIST_USER_INFO
3 WHERE AUTHORIZATION_LIST = 'QPMCCDATA';
```

AUTHORIZATION_LIST	AUTHORIZATION_NAME	OBJECT_AUTHORITY
QPMCCDATA	*PUBLIC	*EXCLUDE
QPMCCDATA	QSYS	*ALL
QPMCCDATA	CM	*ALL
QPMCCDATA	QLWISVR	*USE

Nouveautés DB2 2017 - 2018



■ JOBQ_INFO

Affiche les JOBQ et le sous-système attaché

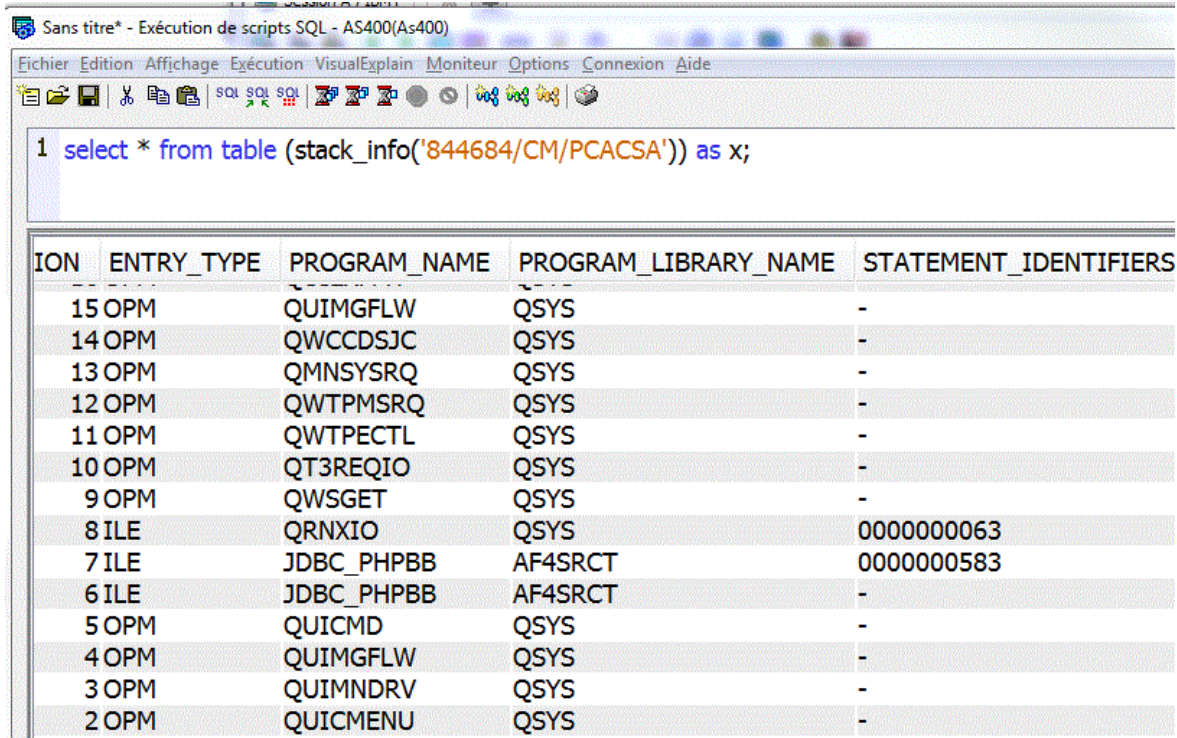
```
1 select * from qsys2.JOB_QUEUE_INFO
```

JOB_QUEUE_NAME	JOB_QUEUE_LIBRARY	JOB_QUEUE_STATUS	NUMBER_OF_JOBS	SUBSYSTEM_NAME	SUBSYSTEM_LIBRARY
VOLUBISJQ	LIBSYS	RELEASED	0	VOLUBIS.FR	LIBSYS
JOBQUSER	QGPL	RELEASED	0	-	-
QBASE	QGPL	HELD	0	-	-
QBATCH	QGPL	RELEASED	0	QBATCH	QSYS
QBATCHS	QGPL	RELEASED	0	QBATCH	QSYS

Nouveautés DB2 2017 - 2018

- STACK_INFO

Affiche la liste d'invocation d'un job (* = le job en cours)



The screenshot shows a DB2 SQL execution window titled "Sans titre* - Exécution de scripts SQL - AS400(As400)". The window contains a menu bar with options: Fichier, Edition, Affichage, Exécution, VisualExplain, Moniteur, Options, Connexion, Aide. Below the menu bar is a toolbar with various icons. The main area displays a SQL query: `1 select * from table (stack_info('844684/CM/PCACSA')) as x;`. Below the query, the results of the execution are shown in a table with the following columns: ION, ENTRY_TYPE, PROGRAM_NAME, PROGRAM_LIBRARY_NAME, and STATEMENT_IDENTIFIERS. The table contains 15 rows of data, with the last two rows (7 and 6) having non-empty STATEMENT_IDENTIFIERS values.

ION	ENTRY_TYPE	PROGRAM_NAME	PROGRAM_LIBRARY_NAME	STATEMENT_IDENTIFIERS
15	OPM	QUIMGFLW	QSYS	-
14	OPM	QWCCDSJC	QSYS	-
13	OPM	QMNSYSRQ	QSYS	-
12	OPM	QWTPMSRQ	QSYS	-
11	OPM	QWTPECTL	QSYS	-
10	OPM	QT3REQIO	QSYS	-
9	OPM	QWSGET	QSYS	-
8	ILE	QRNXIO	QSYS	0000000063
7	ILE	JDBC_PHPBB	AF4SRCT	0000000583
6	ILE	JDBC_PHPBB	AF4SRCT	-
5	OPM	QUICMD	QSYS	-
4	OPM	QUIMGFLW	QSYS	-
3	OPM	QUIMNDRV	QSYS	-
2	OPM	QUICMENU	QSYS	-

Nouveautés DB2 2017 - 2018



- ASP_INFO

donne des informations sur les ASP (fonctionne avec l'ASP système)

```
1 select asp_type, system_storage as pourcent , total_capacity_available as reste
2 from qsys2.asp_info;
3 |
```

ASP_TYPE	POURCENT	RESTE
SYSTEM	41	284002

- ASP_VARY_INFO

donne l'état des ASP indépendant

Nouveautés DB2 2017 - 2018



■ HISTORY_LOG_INFO et DISPLAY_JOURNAL (QAUDJRN uniquement)

peuvent fournir des informations au format [SYSLOG](#) (Unix)

```
1 SELECT syslog_facility as facility , syslog_severity as severity , syslog_event as event
2 FROM TABLE (QSYS2.HISTORY_LOG_INFO(START_TIME => now() - 2 hours,
3 GENERATE_SYSLOG =>'RFC3164')) as x;
```

FACILITY	SEVERITY	EVENT
1	6 <14>	Nov 8 08:45:29 AS400 CEF:0!IBM!IBM i!7.3!QSYS-QHST!CPIAD0B!Low!reason=CPIAD0B msg=*SIGNON server job 84491...
1	6 <14>	Nov 8 08:45:29 AS400 CEF:0!IBM!IBM i!7.3!QSYS-QHST!CPF1124!Low!reason=CPF1124 msg=Travail 845065/QUSER/QZD...
1	6 <14>	Nov 8 08:45:29 AS400 CEF:0!IBM!IBM i!7.3!QSYS-QHST!CPIAD09!Low!reason=CPIAD09 msg=User CM from client 10.3.1....
1	6 <14>	Nov 8 08:45:29 AS400 CEF:0!IBM!IBM i!7.3!QSYS-QHST!CPF1124!Low!reason=CPF1124 msg=Travail 845066/QUSER/QZD...

```
5 SELECT syslog_facility as facility , syslog_severity as severity , syslog_event as event
6 FROM TABLE (QSYS2.DISPLAY_JOURNAL('QSYS', 'QAUDJRN',
7 GENERATE_SYSLOG =>'RFC5424'
8 ) ) AS X
9 WHERE syslog_event IS NOT NULL;
```

FACILITY	SEVERITY	EVENT
4	4 <36>	1 2017-11-06T05:26:35.270944+01:00 AS400.VOLUBIS.FR - - - - CEF:0 IBM IBM i 7.3 QSYS-QAUDJRN T
4	4 <36>	1 2017-11-06T05:26:39.091328+01:00 AS400.VOLUBIS.FR - - - - CEF:0 IBM IBM i 7.3 QSYS-QAUDJRN T
4	4 <36>	1 2017-11-06T05:26:39.091600+01:00 AS400.VOLUBIS.FR - - - - CEF:0 IBM IBM i 7.3 QSYS-QAUDJRN T
4	4 <36>	1 2017-11-06T05:26:39.985280+01:00 AS400.VOLUBIS.FR - - - - CEF:0 IBM IBM i 7.3 QSYS-QAUDJRN T
4	4 <36>	1 2017-11-06T05:26:47.832768+01:00 AS400.VOLUBIS.FR - - - - CEF:0 IBM IBM i 7.3 QSYS-QAUDJRN T

Nouveautés DB2 2017 - 2018



- le projet [DashforIBMi](#) montre bien la puissance de Node.js.

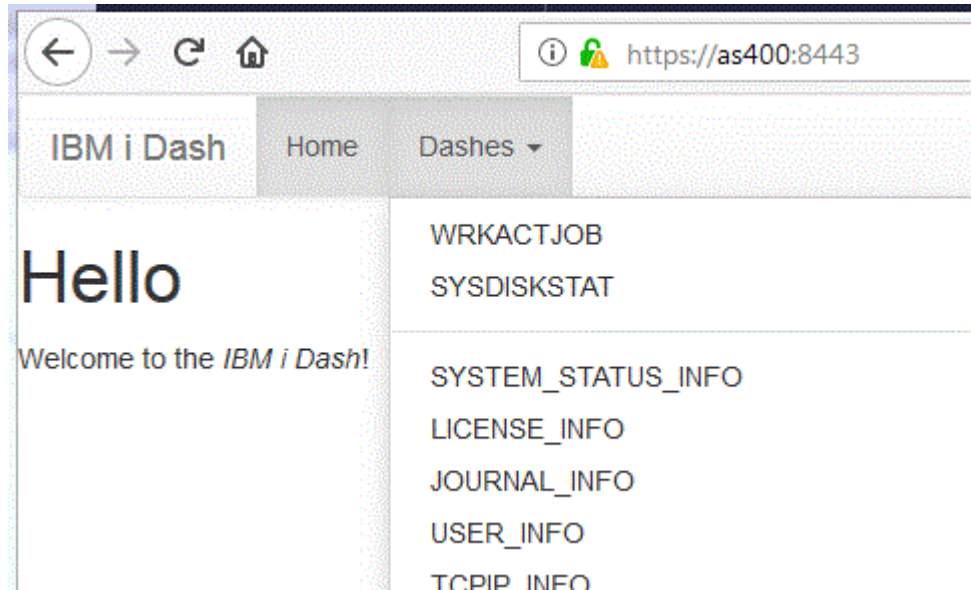
Il est basé sur une architecture modèle/vue/contrôleur

Les vues étant réalisées à l'aide [jade](#) , template de représentation

- Au niveau serveur IBMi, tout cela est fortement basé sur « SQL as a service » que nous venons de voir !

Nouveautés DB2 2017 - 2018

- le projet [DashforIBMi](#) montre bien la puissance de Node.js.



Nouveautés DB2 2017 - 2018



- le projet [DashforIBMi](#) montre bien la puissance de Node.js.

A screenshot of a web browser displaying the 'IBM i Dash' interface. The browser's address bar shows the URL 'https://as400:8443/SYSTEM_STATUS_INFO'. The page title is 'SYSTEM_STATUS_INFO'. Below the title, there is a SQL query: 'SELECT * FROM QSYS2.SYSTEM_STATUS_INFO' and a link 'For more info Click Here'. The main content is a table with system status metrics.

Metric	Value
TOTAL_JOBS_IN_SYSTEM	384
MAXIMUM_JOBS_IN_SYSTEM	163520
ACTIVE_JOBS_IN_SYSTEM	364
INTERACTIVE_JOBS_IN_SYSTEM	.00
ELAPSED_TIME	1
ELAPSED_CPU_USED	.90
ELAPSED_CPU_SHARED	4.20
ELAPSED_CPU_UNCAPPED_CAPACITY	.70
CONFIGURED_CPUS	1
CPU_SHARING_ATTRIBUTE	UNCAPPED

Nouveautés DB2 2017 - 2018



■ DashforIBMi

```
app.get('/SYSTEM_STATUS_INFO', function(req, res) {
  var title = "SYSTEM_STATUS_INFO"
  var sql = "SELECT * FROM QSYS2.SYSTEM_STATUS_INFO"
  var url = "https://www.ibm.com/developerworks/community/wikis/home?lang=en#!/wiki/IBM%20i%20Technology%20Updates/pa
  try {
    db.exec(sql, function(results) {
      res.render('table_view', { title: title, sql: sql, url: url, results: results});
      //console.log(results);
    });
  }
  catch (err) {
    console.log(err);
  }
});

app.get('/LICENSE_INFO', function(req, res) {
  var title = "LICENSE_INFO"
  var sql = "SELECT * FROM QSYS2.LICENSE_INFO"
  var url = "https://www.ibm.com/developerworks/community/wikis/home?lang=en#!/wiki/IBM%20i%20Technology%20Updates/pa
  try {
    db.exec(sql, function(results) {
      res.render('table_view', { title: title, sql: sql, url: url, results: results});
      //console.log(results);
    });
  }
  catch (err) {
    console.log(err);
  }
});

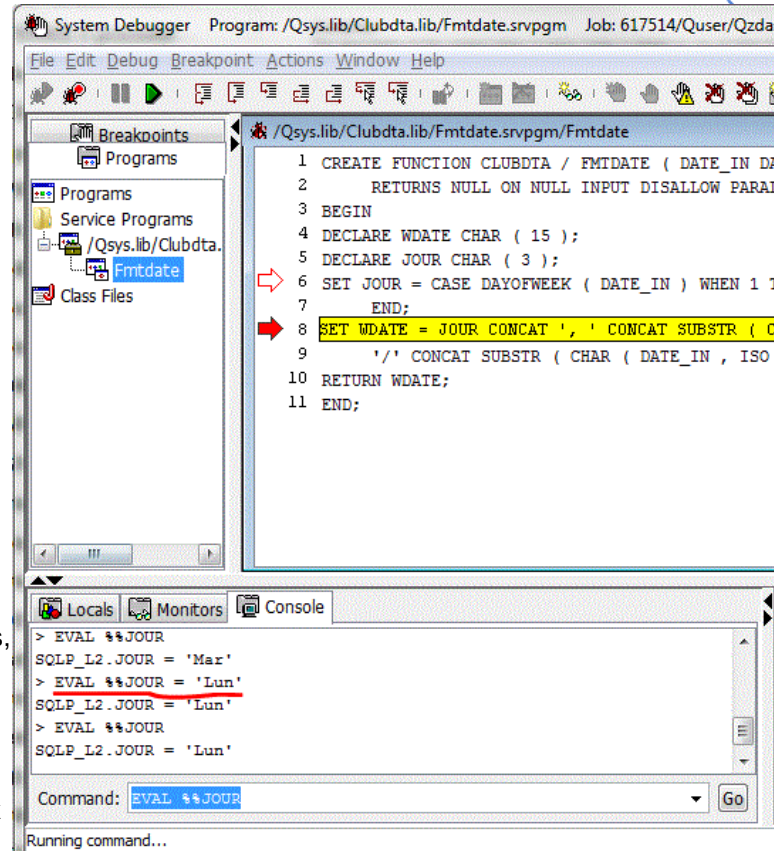
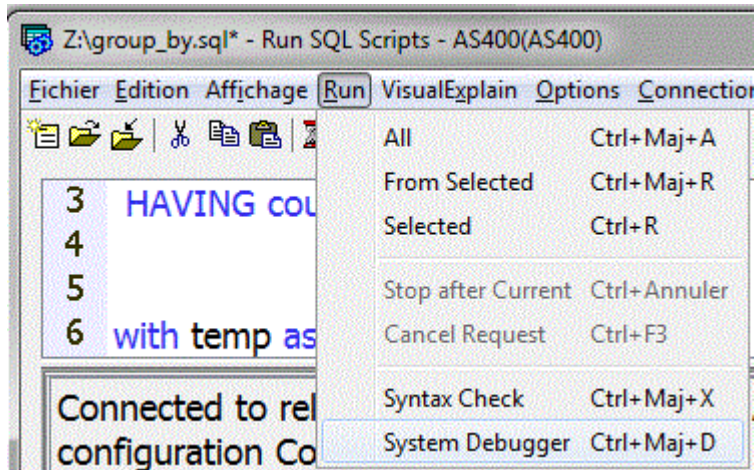
app.get('/JOURNAL_INFO', function(req, res) {
  var title = "JOURNAL_INFO"
  var sql = "SELECT * FROM QSYS2.JOURNAL_INFO"
```

The background of the slide features a complex network diagram. It consists of numerous small, light-gray circular nodes connected by thin, light-gray lines. The nodes are distributed across the entire page, with a higher density in the center and towards the right side. The connections between nodes form a web-like structure, with some nodes having multiple connections and others having only one or two. The overall appearance is that of a large, interconnected network or graph.

ACS

Nouveautés DB2 2017 - 2018

- Le Debug SQL (accessible depuis ACS) évolue



cette évolution est accessible pour les procédures, les fonctions et les triggers, compilés avec SET OPTION DBGVIEW=*SOURCE et recompilés depuis la TR1

- les types de variables couvert par le debug (visualisation) sont plus nombreux
- il est possible de modifier le contenu de variables

Nouveautés DB2 2017 - 2018

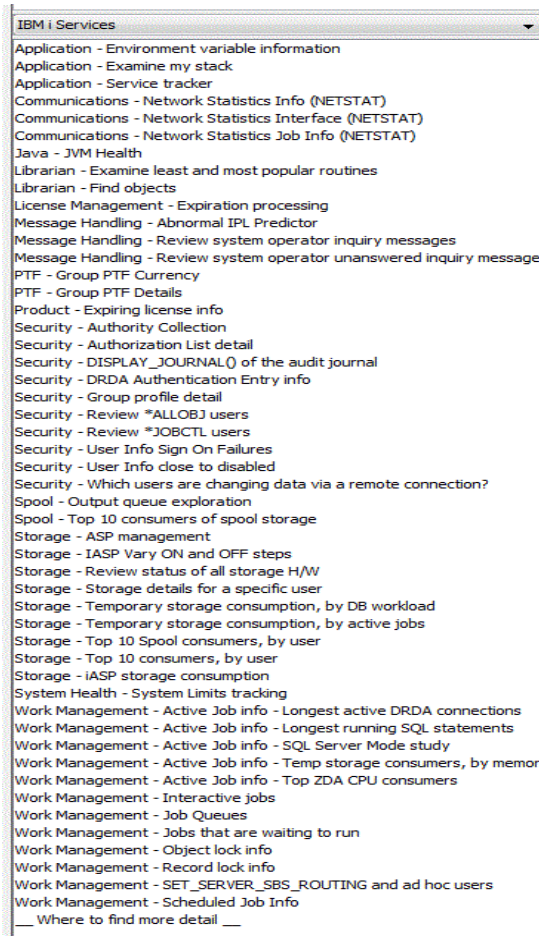
■ ACS 1.1.7.3

Gestionnaire de scripts

De nombreux exemples proposés

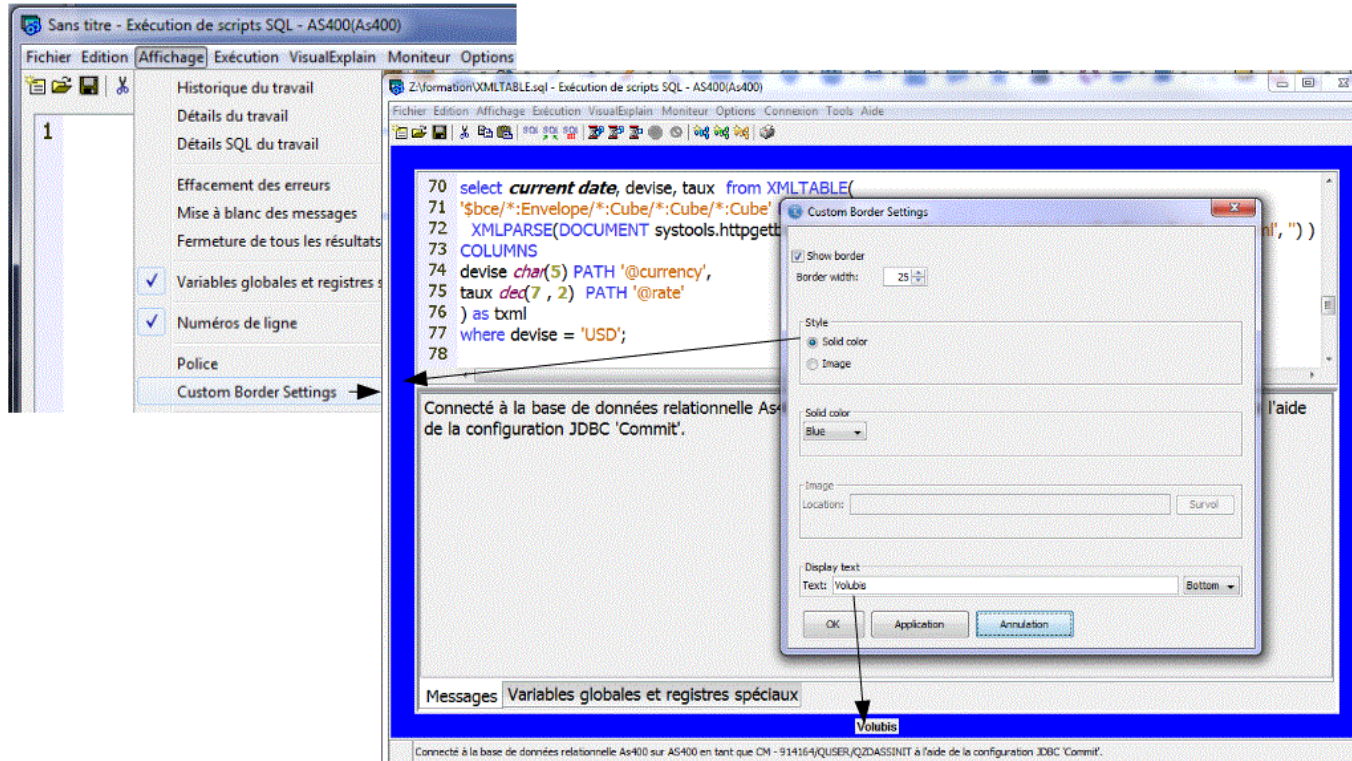
Dont beaucoup de services vus plus haut (IBMi services)

(Edition / insertion à partir d'exemples)



Nouveautés DB2 2017 - 2018

- Une astuce



The screenshot shows the DB2 SQL Editor interface. The 'Affichage' (View) menu is open, and the 'Custom Border Settings' option is selected. A dialog box titled 'Custom Border Settings' is displayed over the editor. The dialog has the following settings:

- Show border
- Border width: 25
- Style: Solid color, Image
- Solid color: Blue
- Image: Location: [] Survival
- Display text: Text: Volubis, Bottom
- Buttons: OK, Application, Annulation

The SQL editor window shows the following code:

```
70 select current date, devise, taux from XMLTABLE(  
71 '$bce/*:Envelope/*:Cube/*:Cube/*:Cube'  
72 XMLPARSE(DOCUMENT systools.httpgett  
73 COLUMNS  
74 devise char(5) PATH '@currency',  
75 taux dec(7, 2) PATH '@rate'  
76 ) as bxml  
77 where devise = 'USD';  
78
```

The status bar at the bottom of the editor displays 'Volubis' and 'Connecté à la base de données relationnelle As400 sur AS400 en tant que CM - 914164/QUSER/QZDASSINIT à l'aide de la configuration JDBC 'Commit'.'

Nouveautés DB2 2017 - 2018

- Nouvelles options « Schémas » et

« SQL performance Center »

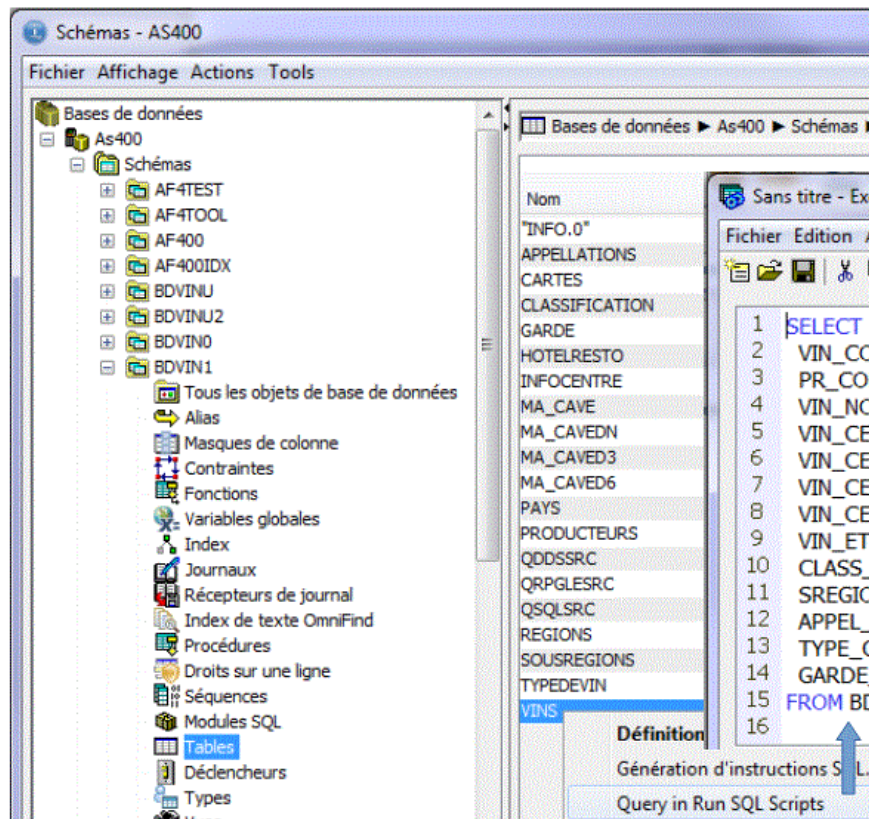


Pour cela, retrouvez nous lors de la session 25

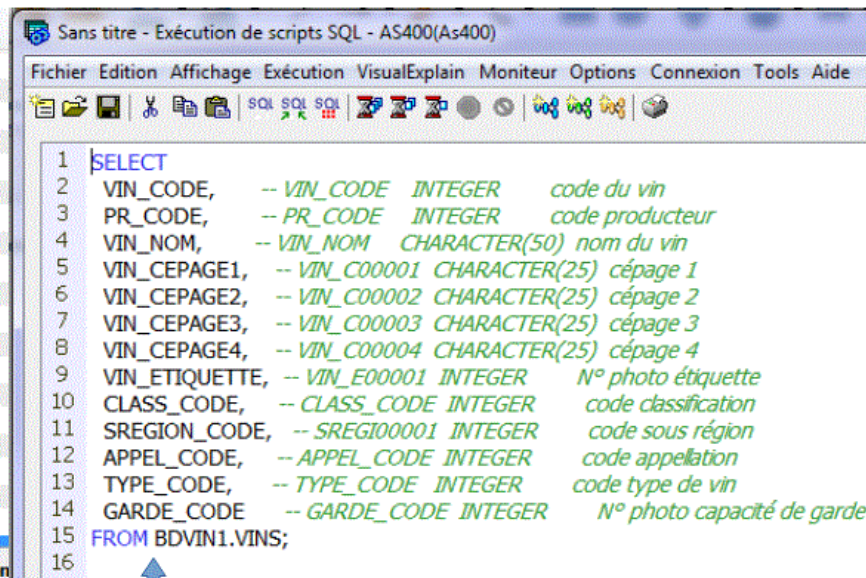
-> Demain Jeudi à 11h

Nouveautés DB2 2017 - 2018

- Par exemple



Depuis l'option « Schémas »



Merci de votre attention

**N'oubliez pas de remplir
le questionnaire
de satisfaction !**