

**Power
Week**

Université IBM i 2019



22 et 23 mai

IBM Client Center Paris

S17 – On peut vraiment faire cela avec SQL ?

Mercredi 22 mai 16h15-17h15

Philippe Bourgeois
IBM France
pbourgeois@fr.ibm.com



On peut vraiment faire cela avec SQL ?

- 1. Création de documents XML et JSON à partir de données DB2
- 2. Décomposition de documents XML et JSON en données DB2
- 3. Accès à des ressources Web et consommation de Services Web
- 4. Accès aux fonctions système IBM i
- 5. Recherches avancées
 - Expressions régulières
 - Recherche phonétique
 - Recherche linguistique
- 6. Divers (CONNECT BY, LISTAGG)

1. Création de documents XML et JSON à partir de données DB2

Les fonctions SQL de publication XML

Fonction	Usage
XMLATTRIBUTES	Renvoie une valeur XML représentant un attribut
XMLCOMMENT	Renvoie une valeur XML représentant un commentaire
XMLCONCAT	Renvoie une valeur XML représentant la concaténation de deux séquences XML
XMLDOCUMENT	Renvoie un document XML depuis une valeur XML
XMLELEMENT	Renvoie une valeur XML représentant un élément
XMLFOREST	Renvoie une valeur XML représentant une séquence d'éléments
XMLNAMESPACES	Renvoie une valeur XML représentant la déclaration d'espaces de nommage
XMLPARSE	Parse des données caractère pour produire un flux XML
XMLPI	Renvoie une valeur XML représentant une séquences de traitement
XMLROW	Renvoie une valeur XML représentant une ligne XML bien formée (avec une racine)
XMLSERIALIZE	Convertit un type XML dans le type demandé
XMLTEXT	Renvoie une chaîne de caractères compatible XML
XMLCAST	Convertit une chaîne en type XML
XMLAGG	Fonction de groupe : agrégation de valeurs XML
XMLGROUP	Fonction de groupe : groupement de valeurs XML

Exemple – Création d'un document XML

```
SELECT titre, annee, duree FROM video.films;
```

TITRE	ANNEE	DUREE
MEILLEUR (LE)	1984	129
POURSUITE IMPITOYABLE (LA)	1965	135
RIDEAU DECHIRE (LE)	1966	125
HOMMES DU PRESIDENT (LES)	1976	126
PSYCHOSE	1960	110
GOOD MORNING, VIETNAM !	1988	117

```
SELECT XMLGROUP(TRIM(titre) AS "titre",  
               annee AS "annee",  
               duree AS "duree"  
               OPTION ROW "film" ROOT "films")  
FROM video.films;
```

```
00001  
<films><film><titre>MEILLEUR (LE)  
</titre><annee>1984</annee><duree>129</duree>  
</film><film><titre>POURSUITE IMPITOYABLE (LA)  
</titre><annee>1965</annee><duree>135</duree></film>
```

Exemple – Création d'un fichier XML dans l'IFS

GEN_XML.SQLRPGLÉ

```
Ligne 1      Colonne 2      Insérer
  .+.....1.....2.....3.....4.....5.....6.....+...
000100 **free
000200 // Création d'un fichier XML dans l'IFS à partir de données DB2
000300
000301 // Déclaration du fichier IFS
000500 dcl-s fichierIFS SQLTYPE(CLOB_FILE);
000600
000601 // Informations concernant le fichier IFS
001200 fichierIFS_name = '/pb/xml/films2.xml';
001300 fichierIFS_nl   = %len(%trim(fichierIFS_name));
001400 fichierIFS_fo   = SQFCRT;
001500
001501 // Génération du fichier XML dans l'IFS à partir des données de DB2
001700 exec sql SELECT XMLDOCUMENT(XMLGROUP(TRIM(titre) AS "titre",
001701 annee AS "annee", duree AS "duree"
001702 OPTION ROW "film" ROOT "films"))
001703 into :fichierIFS FROM video.films;
001704
001900 *inlr = *on;
```

Fichiers de /pb/xml

emp.xml

films.xml

films2.xml

films2.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<films>
  <film>
    <titre>ANGEL HEART</titre>
    <annee>1987</annee>
    <duree>115</duree>
  </film>
  <film>
    <titre>APPRENTIS (LES)</titre>
    <annee>1995</annee>
    <duree>95</duree>
  </film>
  <film>
    <titre>BANDOLERO !</titre>
    <annee>1968</annee>
    <duree>115</duree>
  </film>
</films>
```

Les opérations possibles sur le fichier IFS

- Valeurs possibles de la variable **xxx_fo** :

```
// Informations concernant le fichier IFS
fichierIFS_name = '/pb/xml/films2.xml';
fichierIFS_nl   = %len(%trim(fichierIFS_name));
fichierIFS_fo   = SQFCRT;
```

- **SQFRD** : lecture du fichier dans l'IFS
- **SQFCRT** : création du fichier dans l'IFS
- **SQFOVR** : remplacement (écrasement) ou ajout du fichier dans l'IFS
- **SQFAPP** : mise à jour du fichier dans l'IFS

Les fonctions SQL de publication JSON

Fonction	Usage
JSON_ARRAY	Génère un tableau JSON
JSON_ARRAY_AGG	Génère un tableau JSON avec agrégation des données
JSON_OBJECT	Génère un objet JSON
JSON_OBJECTAGG	Génère un objet JSON avec agrégation des données

Exemple – Création d'un document JSON

```
SELECT titre, annee, duree FROM video.films;
```

TITRE	ANNEE	DUREE
MEILLEUR (LE)	1984	129
POURSUITE IMPITOYABLE (LA)	1965	135
RIDEAU DECHIRE (LE)	1966	125
HOMMES DU PRESIDENT (LES)	1976	126
PSYCHOSE	1960	110
GOOD MORNING, VIETNAM !	1988	117

```
SELECT JSON_OBJECT('films' VALUE JSON_ARRAYAGG(
  JSON_OBJECT('titre' VALUE RTRIM(titre),
    'annee' VALUE annee, 'duree' VALUE duree)))
FROM video.films;
```

```
{"films":[{"titre":"MEILLEUR (LE)","annee":1984,"duree":129},
{"titre":"POURSUITE IMPITOYABLE (LA)","annee":1965,
"duree":135},{"titre":"RIDEAU DECHIRE (LE)","a...
```

Exemple – Création d'un fichier JSON dans l'IFS

```
GEN_JSON.SQLRPGLE ☒
Ligne 1      Colonne 1      Insérer
...+...1...+...2...+...3...+...4...+...5...+...6...+...
000100 **free
000200 // Création d'un fichier JSON dans l'IFS à partir de données DB2
000300
000301 // Déclaration du fichier IFS
000500 dcl-s fichierIFS SQLTYPE(CLOB_FILE);
000600
000601 // Informations concernant le fichier IFS
001200 fichierIFS_name = '/pb/json/films2.json';
001300 fichierIFS_nl   = %len(%trim(fichierIFS_name));
001400 fichierIFS_fo   = SQFCRT;
001500
001501 // Génération du fichier JSON dans l'IFS à partir des données de
001700 exec sql SELECT JSON_OBJECT('films' VALUE JSON_ARRAYAGG(
001701     JSON_OBJECT('titre' VALUE RTRIM(titre),
001702     'annee' VALUE annee, 'duree' VALUE duree)))
001703     into :fichierIFS
001704     FROM video.films;
001705
001900 *inlr = *on;
```

Fichiers de /pb/json

films.json

films2.json

films2.json ☒

```
{
  "films": [{
    "titre": "MEILLEUR (LE)",
    "annee": 1984,
    "duree": 129
  },
  {
    "titre": "POURSUITE IMPITOYABLE (LA)",
    "annee": 1965,
    "duree": 135
  },
  {
    "titre": "RIDEAU DECHIRE (LE)",
    "annee": 1966,
    "duree": 125
  }
  ],
}
```

2. Décomposition de documents XML et JSON en données DB2

La fonction SQL XMLTABLE

- XMLTABLE : décomposition d'un document XML en données DB2
 - Renvoie une "pseudo table" : ensemble de lignes et de colonnes

- Syntaxe :

Générera autant de lignes que d'éléments ou d'attributs XML trouvés

```
SELECT colonne1,colonne2... FROM XMLTABLE('/chemin/élément ou attribut XML'  
PASSING XMLPARSE(DOCUMENT document_XML)  
COLUMNS nom_colonne1 type_colonne1 PATH '/chemin/élément ou attribut XML',  
nom_colonne2 type_colonne2 PATH '/chemin/élément ou attribut XML',  
etc.) AS X
```

Mapping Colonne \leftrightarrow Élément ou attribut XML

XMLTABLE – Exemple

Décomposition d'un fichier XML de l'IFS en données DB2

```
films.xml
<?xml version="1.0" encoding="UTF-8" ?>
<films>
  <film titre="MATRIX">
    <annee>1999</annee>
    <acteur>REEVES</acteur>
    <acteur>MOSS</acteur>
  </film>
  <film titre="SPIDERMAN">
    <annee>2002</annee>
    <acteur>MAGUIRE</acteur>
    <acteur>DUNST</acteur>
    <acteur>DAFOE</acteur>
  </film>
  <film titre="INCEPTION">
    <annee>2010</annee>
    <acteur>DI CAPRIO</acteur>
    <acteur>COTILLARD</acteur>
  </film>
  <film titre="GLADIATOR">
    <annee>2000</annee>
    <acteur>CROWE</acteur>
    <acteur>PHOENIX</acteur>
  </film>
</films>
```

```
SET TRANSACTION ISOLATION LEVEL READ COMMITTED;
```

```
SELECT * FROM XMLTABLE('/films/film'
  PASSING XMLPARSE(DOCUMENT GET_XML_FILE('/pb/xml/films.xml'))
  COLUMNS titre  VARCHAR(40) PATH '@titre',
           annee  CHAR(4)      PATH 'annee') AS x;
```

```
SET TRANSACTION ISOLATION LEVEL NONE;
```

TITRE	ANNEE
MATRIX	1999
SPIDERMAN	2002
INCEPTION	2010
GLADIATOR	2000

XMLTABLE – Exemple

Décomposition d'un fichier XML de l'IFS en données DB2

```
films.xml
<?xml version="1.0" encoding="UTF-8" ?>
<films>
  <film titre="MATRIX">
    <annee>1999</annee>
    <acteur>REEVES</acteur>
    <acteur>MOSS</acteur>
  </film>
  <film titre="SPIDERMAN">
    <annee>2002</annee>
    <acteur>MAGUIRE</acteur>
    <acteur>DUNST</acteur>
    <acteur>DAFOE</acteur>
  </film>
  <film titre="INCEPTION">
    <annee>2010</annee>
    <acteur>DI CAPRIO</acteur>
    <acteur>COTILLARD</acteur>
  </film>
  <film titre="GLADIATOR">
    <annee>2000</annee>
    <acteur>CROWE</acteur>
    <acteur>PHOENIX</acteur>
  </film>
</films>
```

```
SELECT * FROM XMLTABLE('/films/film/acteur'
  PASSING XMLPARSE(DOCUMENT GET_XML_FILE('/pb/xml/films.xml'))
  COLUMNS titre  VARCHAR(40) PATH './@titre',
            annee  CHAR(4)     PATH './annee',
            acteur VARCHAR(20) PATH './') AS x;
```

TITRE	ANNEE	ACTEUR
MATRIX	1999	REEVES
MATRIX	1999	MOSS
SPIDERMAN	2002	MAGUIRE
SPIDERMAN	2002	DUNST
SPIDERMAN	2002	DAFOE
INCEPTION	2010	DI CAPRIO
INCEPTION	2010	COTILLARD
GLADIATOR	2000	CROWE
GLADIATOR	2000	PHOENIX

Fonction XMLTABLE – Compléments

- L'origine du document XML peut être:
 - Une variable
 - Un fichier dans l'IFS (récupération par la fonction GET_XML_FILE)
 - Une zone de type XML / VARCHAR dans une table DB2
 - Le résultat de l'appel d'un Service Web
- Il est possible de gérer les espaces de nommage
 - Fonction XMLNAMESPACES
- La recherche des éléments et attributs XML se fait par des expressions XPath (`XMLTABLE('//film[annee > 2000]')`)
- La gestion des éléments multiples peut se faire de différentes façons :
 - Récupération d'un seul élément (acteur[1], acteur[2]...)
 - Récupération de l'ensemble des éléments dans une zone XML, etc.

La fonction SQL JSON_TABLE

- JSON_TABLE : décomposition d'un document JSON en données DB2
 - Renvoie une "pseudo table" : ensemble de lignes et de colonnes
- Syntaxe :

```
SELECT colonne1,colonne2... FROM JSON_TABLE(document_JSON,  
'$<.chemin.étiquette>'  
COLUMNS nom_colonne1 type_colonne1 PATH '$.chemin.étiquette',  
nom_colonne2 type_colonne2 PATH '$.chemin.étiquette',  
etc.) AS X
```

Générera autant de
lignes que d'étiquettes
trouvées

Mapping Colonne ↔ Etiquette JSON

JSON_TABLE – Exemple

Décomposition d'un fichier **JSON** de l'IFS en données DB2

```
films.json
{
  "films": [
    {
      "titre": "MATRIX",
      "annee": 1999,
      "acteurs": ["REEVES", "MOSS"]
    },
    {
      "titre": "SPIDERMAN",
      "annee": 2002,
      "acteurs": ["MAGUIRE", "DUNST", "DAFOE"]
    },
    {
      "titre": "INCEPTION",
      "annee": 2010,
      "acteurs": ["DI CAPRIO", "COTILLARD"]
    },
    {
      "titre": "GLADIATOR",
      "annee": 2000,
      "acteurs": ["CROWE", "PHOENIX"]
    }
  ]
}
```

```
SET TRANSACTION ISOLATION LEVEL READ COMMITTED;

SELECT * FROM JSON_TABLE(
  GET_CLOB_FROM_FILE('/pb/json/films.json'), '$.films[*]'
  COLUMNS(titre VARCHAR(40) PATH '$.titre',
           annee CHAR(4) PATH '$.annee'))
  AS x;

SET TRANSACTION ISOLATION LEVEL NONE;
```

TITRE	ANNEE
MATRIX	1999
SPIDERMAN	2002
INCEPTION	2010
GLADIATOR	2000

JSON_TABLE – Exemple

Décomposition d'un fichier **JSON** de l'IFS en données DB2

```
films.json
{
  "films": [
    {
      "titre": "MATRIX",
      "annee": 1999,
      "acteurs": ["REEVES", "MOSS"]
    },
    {
      "titre": "SPIDERMAN",
      "annee": 2002,
      "acteurs": ["MAGUIRE", "DUNST", "DAFOE"]
    },
    {
      "titre": "INCEPTION",
      "annee": 2010,
      "acteurs": ["DI CAPRIO", "COTILLARD"]
    },
    {
      "titre": "GLADIATOR",
      "annee": 2000,
      "acteurs": ["CROWE", "PHOENIX"]
    }
  ]
}
```

```
SELECT * FROM JSON_TABLE(
  GET_CLOB_FROM_FILE('/pb/json/films.json'), '$.films[*]'
  COLUMNS(titre VARCHAR(40) PATH '$.titre',
           annee CHAR(4) PATH '$.annee',
           NESTED PATH '$.acteurs[*]'
           COLUMNS(acteur VARCHAR(20) PATH '$'))
  AS X;
```

TITRE	ANNEE	ACTEUR
MATRIX	1999	REEVES
MATRIX	1999	MOSS
SPIDERMAN	2002	MAGUIRE
SPIDERMAN	2002	DUNST
SPIDERMAN	2002	DAFOE
INCEPTION	2010	DI CAPRIO
INCEPTION	2010	COTILLARD
GLADIATOR	2000	CROWE
GLADIATOR	2000	PHOENIX

Fonction JSON_TABLE – Compléments

- L'origine du document JSON peut être:
 - Une variable
 - Un fichier dans l'IFS (récupération par la fonction GET_CLOB_FROM_FILE)
 - Une zone de type VARCHAR dans une table DB2
 - Le résultat de l'appel d'un Service Web
- Il est possible de définir :
 - Des règles de mapping (lax ou strict) : `PATH 'lax $.acteurs'`
 - Des options :
 - NULL ON EMPTY : retourne valeur nulle sur un élément manquant
 - ERROR ON EMPTY : retourne une erreur sur un élément manquant
 - DEFAULT xxx ON EMPTY : retourne une valeur par défaut sur un élément manquant
 - ERROR ON ERROR : retourne une erreur en cas d'erreur (SQL16410)
 - DEFAULT xxx ON ERROR : retourne une valeur par défaut en cas d'erreur

3. Accès à des ressources Web et consommation de Services Web

Les fonctions HTTP de SQL

- Ensemble de fonctions SQL permettant d'appeler les méthodes HTTP (HEAD, GET, POST, PUT, DELETE)
- Deux ensembles de fonctions disponibles dans la bibliothèque SYSTOOLS : HTTP* et HTTP*VERBOSE
 - HTTP*
 - UDF – Fonctions scalaires (renvoient la valeur résultat de la réponse HTTP)
 - HTTP*VERBOSE
 - UDTF – Fonctions table (renvoient une table contenant une ligne, avec une colonne pour la valeur de l'entête HTTP et une colonne pour la valeur résultat de la réponse HTTP)

Les fonctions HTTP de SQL

- Fonctions de base
 - **HTTP**[Méthode][TypeDuRésultat]
 - Où
 - Méthode = {GET, POST, PUT, DELETE}
 - TypeDuRésultat = {BLOB, CLOB}
 - Par exemple HTTPGETCLOB
 - Permet de demander une ressource par la méthode HTTP GET, le résultat étant retourné sous la forme d'un CLOB (Character Large Object)
- Autres fonctions
 - HTTPHEAD
 - URLENCODE et URLDECODE
 - BASE64ENCODE et BASE64DECODE

Syntaxe des fonctions HTTP

- Syntaxe :
 - **HTTP**[Méthode][TypeDuRésultat]('URL de la ressource Web', 'entêtes HTTP', <'corps de la requête'>)
- Documentation :

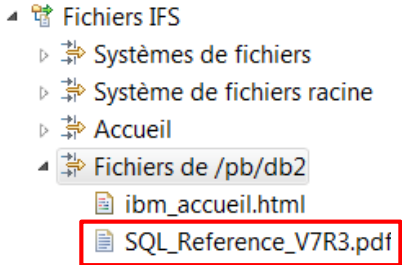
Accessing web services
using IBM DB2 for i HTTP UDFs and
UDTFs

HTTP method	Name	Return type	Parameter	Parameter type
GET	httpGetBlob	BLOB(2G)		
			URL	VARCHAR(2048)
			HTTPHEADER	CLOB(10K)
POST	httpPostBlob	BLOB(2G)		
			URL	VARCHAR(2048)
			HTTPHEADER	CLOB(10K)
			REQUESTMSG	BLOB(2G)

...

Exemple – Récupération d'un PDF sur le Web et ajout dans l'IFS

```
HTTP1B.SQLRPGLE x
Ligne 1      Colonne 8      Insérer
.....1.....2.....3.....4.....5.....6.....7.....8.....9.....0.....
000100 **free
000200 // Récupération d'un PDF sur le Web et stockage dans l'IFS
000300
000400 // Déclaration du fichier BLOB
000500 dcl-s fichierIFS SQLTYPE(BLOB_FILE);
000501
000502 // Déclaration de l'URL du PDF et du timeout de connexion
000503 dcl-s url varchar(100) inz('https://www.ibm.com/support/knowledgecenter/ssw_ibm_i_73/db2/rbafzpdf.pdf');
000504 dcl-s headers varchar(100) inz('<httpHeader connectTimeOut="5000"></httpHeader>');
000600
000700 // Informations concernant le fichier IFS
000800 fichierIFS_name = '/pb/db2/SQL_Reference_V7R3.pdf';
000900 fichierIFS_nl   = %len(%trim(fichierIFS_name));
001000 fichierIFS_fo   = SQFCRT;
001100
001200 // Récupération de la ressource Web et alimentation du fichier dans l'IFS
001300 exec sql values SYSTOOLS.HTTPGETBLOB(:url, :headers) into :fichierIFS;
001400
001500 *inlr = *on;
```



- Fichiers IFS
 - Systèmes de fichiers
 - Système de fichiers racine
 - Accueil
 - Fichiers de /pb/db2
 - ibm_accueil.html
 - SQL_Reference_V7R3.pdf

Consommation de Services Web en SQL

- Appel du Service Web par les fonctions HTTP
- Si le résultat est du XML ou du JSON, décomposition par les fonctions XMLTABLE ou JSON_TABLE
- Compléments :
 - Si besoin il est possible d'indiquer les caractéristiques d'un proxy HTTP
 - Il est possible d'appeler un Service Web sécurisé par un certificat SSL (fichier de propriétés + utilitaire keytool)

Accessing web services
using IBM DB2 for i HTTP UDFs and UDTFs → **Using a truststore and keystore for SSL**

Exemple 1 – Appel d'un Service Web – 1/3

PARIS | Data → Connexion

🏠 - Accueil **Les données** L'API La licence La démarche Cartographie

258 jeux de données

Tri des jeux Récemment modifiés

Modifié Populaires De A à Z ⋮

Filtres

Trouver un jeu de données. 🔍

Vue

- 📊 Analyse 249
- 📍 Carte 147
- 🖼️ Image 4

Vélib' : Disponibilité temps réel

Déplacements

Ce jeu de données donne les disponibilités en temps réel des stations Vélib'.

Modifié	18 avril 2019 15:42
Producteur	Autolib Velib Métropole
Licence	Open Database License (ODbL)
Mots clés	velib, vélos, mobilité, Métropole
Données	1 360 éléments

- 📊 Tableau
- 📍 Carte
- 📊 Analyse
- 📄 Export
- ⚙️ API

Paris Wi-Fi - Sites disposant du service

Liste des sites municipaux disposant d'un hotspot Paris Wi-Fi permettant une connexion à Internet limitée.

Modifié	18 avril 2019 09:00
Producteur	Mairie de Paris / Direction des Systèmes d'Information et du Numérique
Licence	Open Database License (ODbL)
Mots clés	wifi, internet, Paris Wi-Fi
Données	248 éléments

- 📊 Tableau
- 📍 Carte
- 📊 Analyse
- 📄 Export
- ⚙️ API

Exemple 1 – Appel d'un Service Web – 2/3

<http://opendata.paris.fr/api/records/1.0/search/?dataset=sites-disposant-du-service-paris-wi-fi>

```
nhits:                248
parameters:          {...}
records:
  0:
    datasetid:        "sites-disposant-du-service-paris-wi-fi"
    recordid:          "fd012d2daa7bf6c8c59985568ecfcf229d1840d7"
    fields:
      etat2:           "Opérationnel"
      nombre_de_borne_wifi: 2
      arc_adresse:      "32 RUE BERGER"
      geo_point_2d:    [...]
      nom_site:          "JARDIN NELSON MANDELA"
      idpw:             "PW0110"
      geo_shape:        {...}
      cp:               "75001"
      ...
```

Retourne du JSON

Exemple 1 – Appel d'un Service Web – 3/3

```
SELECT *  
FROM JSON_TABLE(  
  SYSTOOLS.HTTPGETCLOB('http://opendata.paris.fr/api/records/1.0/search/?dataset=sites-disposant-du-service-paris-wi-fi',"),  
  '$.records[*]'  
COLUMNS(  
  nom CHAR(50) PATH '$.fields.nom_site',  
  activite CHAR(100) PATH '$.fields.arc_adresse')) AS resultat;
```

NOM	ACTIVITE
JARDIN NELSON MANDELA	32 RUE BERGER
MAIRIE DU 10 EME	72 RUE DU FAUBOURG SAINT MARTIN
MAISON DES INITIATIVES ETUDIANTES	50 RUE DES TOURNELLES
BERGES RIVE DROITE - PONT NEUF	BERGE DE SEINE RIVE DROITE - SORTIE TUNNEL TUILERIE
SQUARE CLAUDE BERNARD	176 BOULEVARD MACDONALD
SQUARE D'ANVERS	PLACE D'ANVERS
SQUARE DES JARDINIERS	2 PASSAGE DUMAS
SQUARE GANDON	211 BOULEVARD MASSENA

Exemple 2 – Appel d'un Service Web (1/3)

- API de reconnaissance visuelle d'IBM Watson
 - <https://www.ibm.com/watson/services/visual-recognition/>
 - Modèle pré entraîné : détermination de l'âge et du genre des visages sur une photo

The screenshot displays the IBM Watson Visual Recognition API interface. On the left, there are two blue buttons: 'Custom Models' and 'Pre Trained Models'. Below them, a text block explains that Watson has pre-trained models for various categories like scenes, objects, faces, colors, food, and other content. In the center, a photograph of two men is shown with green dashed boxes around their faces, indicating face detection. Below the photo, there are small text labels for each man: 'TIM ROWE' (BUSINESS ARCHITECT, OPERATIONAL DEVELOPMENT AND SYSTEMS MANAGEMENT) and 'SCOTT FORSTIE' (BUSINESS ARCHITECT, DIGITAL). On the right, a dark panel titled 'Face Model' shows the results for two faces. The results are as follows:

Face	Age	Gender	Confidence
Face 1	47 - 50	MALE	0.88
Face 2	48 - 51	MALE	0.83

Exemple 2 – Appel d'un Service Web (2/3)

- https://apikey: [redacted]
[redacted]
[redacted]@gateway.watsonplatform.net/visual-recognition/api/v3/detect_faces?url=http://pages.nextbooks.com/nxtbooks/ibmsystemsmag/ibmsystems_power_201806/iphone/ibmsystemsmag_ibmsystems_power_201806_p0035_hires.jpg?1527789949&version=2018-03-19

```
▼ images:  
  ▼ 0:  
    ▼ faces:  
      ▼ 0:  
        ▼ age:  
          min: 47  
          max: 50  
          score: 0.8763784  
        ▼ face_location:  
          height: 160  
          width: 146  
          left: 377  
          top: 196  
        ▼ gender:  
          gender: "MALE"  
          gender_label: "homme"  
          score: 0.9999864  
      ▼ 1:  
        ▼ age:  
          min: 48  
          max: 51  
          score: 0.82596815  
        ▼ face_location:  
          height: 136  
          width: 125  
          left: 817  
          top: 173  
        ▼ gender:  
          gender: "MALE"  
          gender_label: "homme"  
          score: 0.99986994
```

Exemple 2 – Appel d'un Service Web (3/3)

```
SELECT *  
FROM JSON_TABLE(SYSTOOLS.HTTPGETCLOB('https://apikey: ... &version=2018-03-19', ''),  
 '$.images[*].faces[*]'  
COLUMNS(  
  min_age      INT          PATH '$.age.min',  
  max_age      INT          PATH '$.age.max',  
  age_score    DECIMAL(5, 2) PATH '$.age.score',  
  genre        VARCHAR(6)   PATH '$.gender.gender',  
  genre_score  DECIMAL(5, 2) PATH '$.gender.score'))  
AS X;
```

MIN_AGE	MAX_AGE	AGE_SCORE	GENRE	GENRE_SCORE
47	50	0.87	MALE	0.99
48	51	0.82	MALE	0.99

4. Les services IBM i : accès aux fonctions système IBM i

Les services IBM i

- Ce sont des vues, procédures et fonctions **SQL** fournies en standard, permettant d'accéder à des fonctions IBM i **système** :
 - PTFs
 - Sécurité
 - Messages
 - Bibliothèques
 - Gestion des travaux
 - Communication
 - Stockage
 - Produits
 - Spoules
 - Centre de santé
 - Journaux
 - Java
 - Applications
- Alternative aux commandes CL et APIs
- Pourquoi SQL ?
 - Standard
 - Compétences
 - Puissance du langage
 - Utilisation externe (Java, PHP...)

Les services IBM i

- Les services sont documentés sur le site **IBM i Technology Updates**
 - ibm.biz/DB2foriServices
- Des exemples sont fournis dans le produit **IBM i Access Client Solutions (ACS)**
 - Fonctionnalité "Exécution de scripts SQL"
- Le service **SERVICES_INFO** donne la liste de tous les services
- Des rapports et graphiques **DB2 Web Query** sont fournis en standard pour habiller en mode Web le résultat SQL

Liste des Services IBM i

- ibm.biz/DB2foriServices

You are in: [IBM i Technology Updates](#) > [IBM i Technology Updates](#) > [DB2 for i - Technology Updates](#) > [DB2 for i - Services](#)

DB2 for i - Services

 5 | Updated 4/24/19 by [ScottForstie](#) | Tags: *None*

Page Actions ▾

IBM i Service	Type of Service	IBM i 7.4	IBM i 7.3	IBM i 7.2
PTF Services				
QSYS2.GROUP_PTF_INFO	View		Base	Base
QSYS2.PTF_INFO	View		Enhanced in Base	Base
SYSTOOLS.FIRMWARE_CURRENCY	View	Base	SF99703 Level 15	-
SYSTOOLS.GROUP_PTF_CURRENCY	View		Enhanced: SF99703 Level 3	SF99702 Level 3 Enhanced: SF99702 Level 14
SYSTOOLS.GROUP_PTF_DETAILS	View		Enhanced: SF99703 Level 3	SF99702 Level 9 Enhanced: SF99702 Level 14
Security Services				
QSYS2.AUTHORITY_COLLECTION	View		Base	-

Les services IBM i – Exemples d'utilisation

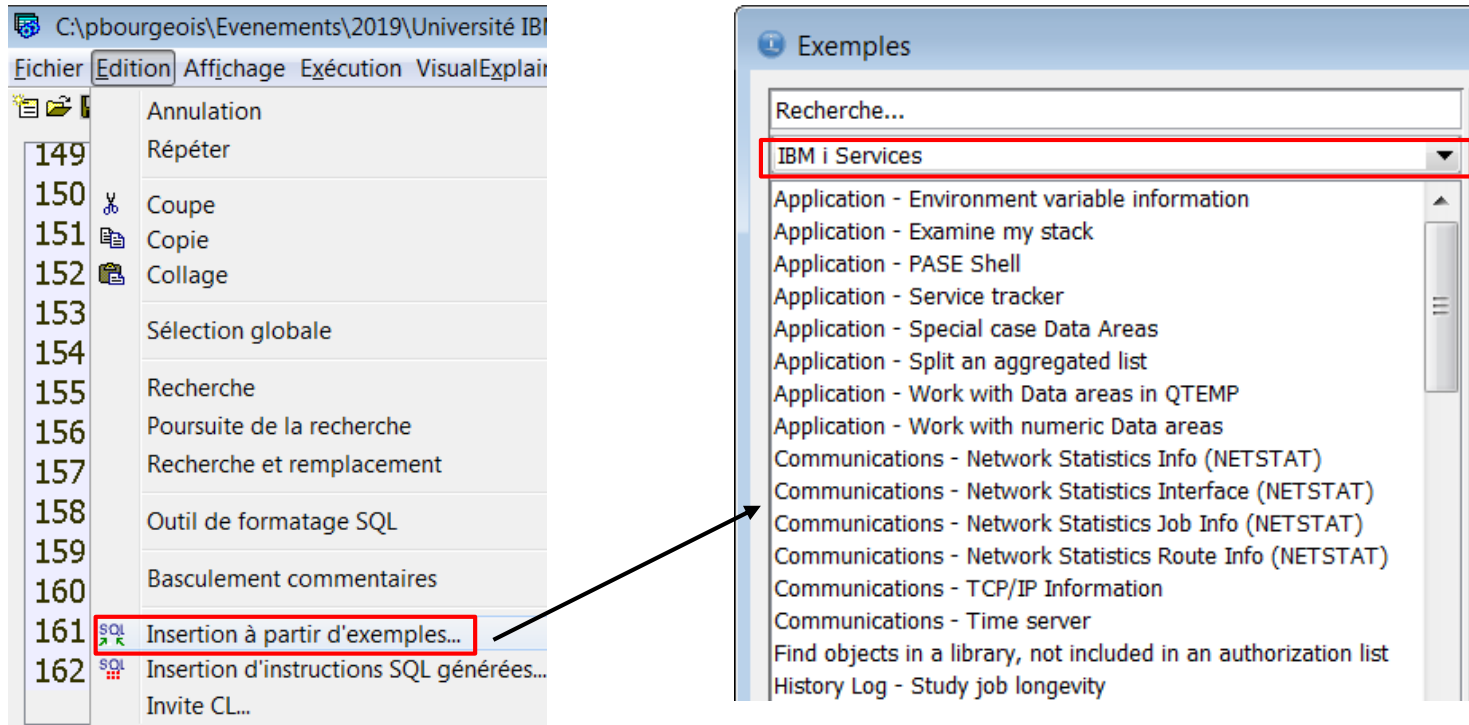
- Colonne **Example** du résultat de la requête `SELECT * FROM QSYS2.SERVICES_INFO;`

EXAMPLE

```
-- Description: Will an IPL impact PTF state?  SELECT PTF_IDENTIFIER, PTF_IPL_ACTION, A.* FROM QSYS2.PTF_INFO A WHERE PTF_IPL_A
-- Description: What is the most recently installed CUM?  SELECT MAX(PTF_GROUP_LEVEL) AS CUM_LEVEL FROM QSYS2.GROUP_PTF_INFO WH
-- Description: Determine whether the PTF Groups are current  SELECT * FROM SYSTOOLS.GROUP_PTF_CURRENCY WHERE PTF_GROUP_RELEA
-- Description: Review all unapplied PTFs contained within PTF Groups installed -- on the partition against the live PTF detail available from IBM
-- Description: Which users are at risk of becoming disabled due to lack of use?  SELECT * FROM QSYS2.USER_INFO WHERE STATUS = '*ENABLE
-- Description: Which function usage IDs exist  SELECT * FROM QSYS2.FUNCTION_INFO ORDER BY FUNCTION_ID;
-- Description: Which Database function usage has been granted or revoked?  SELECT * FROM QSYS2.FUNCTION_USAGE WHERE FUNCTION_ID
-- Description: Review *ALLOBJ users  SELECT AUTHORIZATION_NAME, STATUS, NO_PASSWORD_INDICATOR, PREVIOUS_SIGNON, TEXT_DESCR
-- Description: Does this user have authority to query this file  VALUES ( CASE WHEN QSYS2.SQL_CHECK_AUTHORITY('QSYS2','SYSLIMITS') = 1
-- Description: Secure salary column values in the SQL Performance Center  CALL SYSPROC.SET_COLUMN_ATTRIBUTE('TOYSTORE', 'EMPLOYEE
-- Description: Retrieve the DRDA Authentication Entry info  SELECT * FROM QSYS2.DRDA_AUTHENTICATION_ENTRY_INFO ORDER BY AUTHOR
-- Description: Compare System Values across two partitions  DECLARE GLOBAL TEMPORARY TABLE SESSION.Remote_System_Values ( SYSTEM
-- Description: Find the jobs with SQL statements executing  WITH ACTIVE_USER_JOBS (Q_JOB_NAME, CPU_TIME, RUN_PRIORITY) AS ( SELECT
-- Description: Find the top 10 consumers of CPU in the QUSRWRK and QSYSWRK subsystems  select CPU_TIME, A.* FROM table(QSYS2.ACTIVE
-- Description: Review the job scheduled entries which are no longer in effect  SELECT * FROM QSYS2.SCHEDULED_JOB_INFO WHERE STATUS
-- Description: Return all available pool information.  SELECT * FROM TABLE(QSYS2.MEMORY_POOL('YES')) X;
-- Description: Review the details of all active memory pools.  SELECT * FROM QSYS2.MEMORY_POOL_INFO;
-- Description: Review elapsed system status over a 60 second window  DECLARE GLOBAL TEMPORARY TABLE SESSION.SYSTEM_STATUS_REFER
-- Description: Review current system status, including the job max percentage  SELECT TOTAL_JOBS_IN_SYSTEM, MAXIMUM_JOBS_IN_SYSTEM,
```

Les services IBM i – Exemples d'utilisation

- La fonction **Insertion à partir d'exemples** d'ACS (Exécution de scripts SQL)



Exemple – PTFs

-- Liste des groupes de PTFs et différence par rapport au dernier niveau IBM

```
SELECT *  
FROM SYSTOOLS.GROUP_PTF_CURRENCY ORDER BY ptf_group_currency;
```

PTF_GROUP_CURRENCY	PTF_GROUP_ID	PTF_GROUP_TITLE	PTF_GROUP_LEVEL_INSTALLED	PTF_GROUP_LEVEL_AVAILABLE	PTF_GROUP_LAST_UPDATED_BY_IBM
INSTALLED LEVEL IS CURRENT	SF99658	SF99658 720 DB2 Web Query for i V2.1.1	6	6 08/24/2017	
INSTALLED LEVEL IS CURRENT	SF99712	720 Technology Refresh plus Recommended Groups	1	1 05/19/2016	
INSTALLED LEVEL IS CURRENT	SF99717	SF99717 720 Technology Refresh	9	9 09/13/2018	
INSTALLED LEVEL IS CURRENT	SF99720	Current Cumulative PTF Media Documentation	18249	18249 10/01/2018	
INSTALLED LEVEL IS CURRENT	SF99747	SF99747 720 DB2 Web Query for i V2.1.0	17	17 08/24/2017	
INSTALLED LEVEL IS CURRENT	SF99766	720 Print PTFs	3	3 01/07/2015	
INSTALLED LEVEL IS CURRENT	SF99767	SF99767 720 720 TCP/IP PTF	4	4 06/22/2018	
INSTALLED LEVEL IS CURRENT	SF99769	720 IBM i integration with BladeCenter and System x	1	1 12/18/2014	
UPDATE AVAILABLE	SF99223	SF99223 720 IBM Open Source Solutions for i	5	6 11/06/2017	
UPDATE AVAILABLE	SF99481	SF99481 720 WebSphere App Server V8.5	9	14 03/01/2019	
UPDATE AVAILABLE	SF99702	SF99702 720 DB2 for IBM i	23	25 04/10/2019	
UPDATE AVAILABLE	SF99713	SF99713 720 IBM HTTP Server for i	28	31 04/08/2019	
UPDATE AVAILABLE	SF99714	SF99714 720 Performance Tools	4	6 10/25/2017	
UPDATE AVAILABLE	SF99715	SF99715 720 Backup Recovery Solutions	25	48 03/20/2019	
UPDATE AVAILABLE	SF99716	SF99716 720 Java	17	20 03/28/2019	
UPDATE AVAILABLE	SF99718	SF99718 720 Group Security	65	73 04/16/2019	
UPDATE AVAILABLE	SF99719	SF99719 720 Group Hiper	129	145 04/16/2019	
UPDATE AVAILABLE	SF99775	SF99775 720 Hardware and Related PTFs	32	33 02/08/2019	
UPDATE AVAILABLE	SF99776	SF99776 720 High Availability for IBM i	1	11 04/19/2019	

Exemple – Profils utilisateur

```
273 -- Profils utilisateur qui n'ont pas été utilisés depuis 90 jours
274 SELECT AUTHORIZATION_NAME, STATUS, LAST_USED_TIMESTAMP
275 FROM QSYS2.USER_INFO
276 WHERE LAST_USED_TIMESTAMP < CURRENT_TIMESTAMP - 90 DAYS
277 AND STATUS <> '*DISABLED'
278 AND AUTHORIZATION_NAME NOT LIKE 'Q%'
279 ORDER BY 3 DESC;
280
```

AUTHORIZATION_NAME	STATUS	LAST_USED_TIMESTAMP
CHEVALIER	*ENABLED	2018-09-25 00:00:00.000000
DELICOURT	*ENABLED	2018-09-25 00:00:00.000000
DERI	*ENABLED	2018-09-04 00:00:00.000000
DUDUYER	*ENABLED	2018-08-29 00:00:00.000000
BOUTEMY	*ENABLED	2018-06-29 00:00:00.000000
MOREAU	*ENABLED	2018-04-17 00:00:00.000000

Exemple – Profils utilisateur

```
CREATE OR REPLACE PROCEDURE PB.DISABLE_UNUSED_PROFILES(IN p_number_of_days INTEGER)
BEGIN
  DECLARE v_user          VARCHAR(10);
  DECLARE v_eof          INTEGER DEFAULT 0;
  DECLARE v_query_text  VARCHAR(1000) DEFAULT
    'SELECT AUTHORIZATION_NAME FROM QSYS2.USER_INFO
     WHERE LAST_USED_TIMESTAMP < CURRENT_TIMESTAMP - ? DAYS
     AND STATUS <> '*DISABLED' AND AUTHORIZATION_NAME NOT LIKE "Q%";
  DECLARE DORMANT_USERS CURSOR FOR user_query;
  DECLARE CONTINUE HANDLER FOR SQLSTATE'02000' SET v_eof = 1;

  PREPARE user_query FROM user_query_text;
  OPEN DORMANT_USERS USING p_number_of_days;

  L1 : LOOP
    FETCH DORMANT_USERS INTO v_user;
    IF (v_eof = 1) THEN LEAVE L1;
    END IF;
    EXECUTE IMMEDIATE 'CALL QSYS2.QCMDEXC("CHGUSRPRF USRPRF(' CONCAT v_user CONCAT ') STATUS(*DISABLED))';
  END LOOP;
  CLOSE DORMANT_USERS;
END;
```

Procédure qui désactive les profils utilisateur qui n'ont pas été utilisés depuis xx jours

A lancer par exemple tous les mois

Exemple – Sécurité

```
247 -- Les utilisateurs qui ont les droits *ALLOBJ, soit directement, soit via un profil de groupe
248 SELECT AUTHORIZATION_NAME, STATUS, NO_PASSWORD_INDICATOR, PREVIOUS_SIGNON, TEXT_DESCRIPTION
249 FROM QSYS2.USER_INFO
250 WHERE SPECIAL_AUTHORITIES LIKE '%*ALLOBJ%' OR AUTHORIZATION_NAME IN (SELECT USER_PROFILE_NAME
251 FROM QSYS2.GROUP_PROFILE_ENTRIES
252 WHERE GROUP_PROFILE_NAME IN (SELECT
253 AUTHORIZATION_NAME
254 FROM QSYS2.USER_INFO
255 WHERE SPECIAL_AUTHORITIES LIKE '%*ALLOBJ%'))
256 ORDER BY AUTHORIZATION_NAME;
257
```

AUTHORIZATION_NAME	STATUS	NO_PASSWORD_INDICATOR	PREVIOUS_SIGNON	TEXT_DESCRIPTION
BOUCHOT	*ENABLED	NO	2014-01-16 15:43:06.000000	Jean-Marie Bouchot
BOURDIN	*ENABLED	NO	2018-12-12 15:26:22.000000	-
BOURGEOIS	*ENABLED	NO	2019-04-23 10:36:59.000000	Philippe Bourgeois
BOUTEMY	*ENABLED	NO	2018-06-29 11:04:36.000000	Patrick Boutemy

Exemple – Messages

Les messages de QSYSOPR de type
interrogation qui n'ont pas eu de réponse

```
150 WITH REPLIED_MSGS(KEY)
151   AS (SELECT a.message_key
152         FROM qsys2.message_queue_info a INNER
153         JOIN qsys2.message_queue_info b ON a.message_key = b.associated_message_key
154         WHERE A.MESSAGE_QUEUE_NAME = 'QSYSOPR' AND A.MESSAGE_QUEUE_LIBRARY = 'QSYS' AND
155              B.MESSAGE_QUEUE_NAME = 'QSYSOPR' AND B.MESSAGE_QUEUE_LIBRARY = 'QSYS'
156         ORDER BY b.message_timestamp DESC)
157 SELECT a.message_text
158 AS "INQUIRY", A.FROM_USER, A.MESSAGE_TIMESTAMP FROM qsys2.message_queue_info a
159 LEFT EXCEPTION JOIN REPLIED_MSGS b ON
160   a.message_key = b.key WHERE MESSAGE_QUEUE_NAME = 'QSYSOPR' AND MESSAGE_QUEUE_LIBRARY =
161   'QSYS' AND message_type = 'INQUIRY' ORDER BY message_timestamp DESC;
```

INQUIRY	FROM_USER	MESSAGE_TIMESTAMP
3 PTFs DB2 installées le 4 mai. IPL possible le 4 à 19h00 ?	BOURGEOIS	2017-05-04 18:33:03.407914

Exemple – Objets

Quelles sont les dates de dernière modification et de dernière utilisation des objets d'une bibliothèque ?

```
165 SELECT objname, objtype, change_timestamp, last_used_timestamp, days_used_count
166 FROM TABLE(QSYS2.OBJECT_STATISTICS('AS425F', '*ALL')) as t
167 ORDER BY change_timestamp DESC;
168
```

OBJNAME	OBJTYPE	CHANGE_TIMESTAMP	LAST_USED_TIMESTAMP	DAYS_USED_COUNT
EMP_DETAIL	*PGM	2017-02-28 11:56:02.000000	2017-03-01 00:00:00.000000	2
QRPGLESRC	*FILE	2017-02-28 11:43:09.000000	2017-02-28 00:00:00.000000	21
SOURCES	*FILE	2016-11-18 09:28:57.000000	2017-05-04 00:00:00.000000	20
TABEMPL4	*FILE	2016-08-25 15:12:22.000000	2016-12-16 00:00:00.000000	3

Exemple – Travaux

Les travaux actifs avec les requêtes SQL les plus longues

```
248 WITH ACTIVE_USER_JOBS(JOB_NAME, CPU_TIME, RUN_PRIORITY)
249   AS (SELECT JOB_NAME, CPU_TIME, RUN_PRIORITY
250        FROM TABLE(ACTIVE_JOB_INFO()) AS T1
251        WHERE JOB_TYPE <> 'SYS')
252   SELECT JOB_NAME, CPU_TIME, RUN_PRIORITY, V_SQL_STATEMENT_TEXT, ABS(CURRENT_TIMESTAMP -
253        V_SQL_STMT_START_TIMESTAMP) AS SQL_STMT_DURATION FROM ACTIVE_USER_JOBS,
254   TABLE(QSYS2.GET_JOB_INFO(JOB_NAME)) AS T2
255   WHERE V_SQL_STMT_STATUS = 'ACTIVE' ORDER BY SQL_STMT_DURATION DESC;
256
```

Q_JOB_NAME	CPU_TIME	RUN_PRIORITY	V_SQL_STATEMENT_TEXT	SQL_STMT_DURATION
562630/QUSER/QRWTSRVR	7802	20	DECLARE C2 CURSOR FOR S2	53.102170
568166/SCOTT/OPADEV000T	8847	20	INSERT into toystore.sales_2 ...	27.158785
566454/QUSER/QZDASOINIT	763	20	WITH ACTIVE_USER_JOBS (Q_JOB_NAME, CPU...	0.000078

Exemple – Spoules

Les 5 OUTQ qui contiennent le plus de spoules, les spoules étant classés par nombre de pages

```
286 WITH outqs_manyfiles(libname, queueName)
287 AS (SELECT OUTPUT_QUEUE_LIBRARY_NAME, OUTPUT_QUEUE_NAME
288 FROM QSYS2.OUTPUT_QUEUE_INFO
289 ORDER BY NUMBER_OF_FILES DESC
290 FETCH FIRST 5 ROWS ONLY)
291 SELECT libname, queueName, create_timestamp, spooled_file_name, user_name, total_pages, SIZE
292 FROM outqs_manyfiles INNER
293 JOIN QSYS2.OUTPUT_QUEUE_ENTRIES ON queueName = OUTPUT_QUEUE_NAME AND libname =
294 OUTPUT_QUEUE_LIBRARY_NAME
295 ORDER BY TOTAL_PAGES DESC;
296
```

LIBNAME	QUEUENAME	CREATE_TIMESTAMP	SPOOLED_FILE_NAME	USER_NAME	TOTAL_PAGES	SIZE
QUSRSYS	QEZJOBLOG	2017-05-07 12:35:52.522000	QPJOBLOG	QSYS	40	264
QUSRSYS	QEZJOBLOG	2017-05-08 04:05:15.963000	QPJOBLOG	QSYS	15	120
QUSRSYS	QEZJOBLOG	2017-05-07 12:36:57.264000	QPJOBLOG	QTCP	12	96
QUSRSYS	HP4000	2017-05-04 17:07:16.301000	AUTORENS	BOURGEOIS	10	56
QUSRSYS	HP4000	2017-02-28 10:56:01.978000	EMP_DETAIL	BOURGEOIS	8	48

Exemple – Journaux

Qui a supprimé des lignes dans la table
EMPLOYES ces 7 derniers jours ?

```
309 SELECT ENTRY_DATA, ENTRY_TIMESTAMP, JOURNAL_ENTRY_TYPE, COUNT_OR_RRN AS ROW_DELETED,  
310 "CURRENT_USER", RTRIM(JOB_NUMBER) CONCAT '/' CONCAT RTRIM(JOB_USER) CONCAT '/' CONCAT  
311 RTRIM(JOB_NAME) AS Qualified_Job_Name  
312 FROM TABLE(QSYS2.DISPLAY_JOURNAL('PBSQL', 'QSQJRN',  
313 STARTING_TIMESTAMP => CURRENT_TIMESTAMP - 7 DAYS,  
314 JOURNAL_ENTRY_TYPES => 'DL',  
315 OBJECT_LIBRARY => 'PBSQL',  
316 OBJECT_NAME => 'EMPLOYES',  
317 OBJECT_OBJTYPE => '*FILE',  
318 OBJECT_MEMBER => 'EMPLOYES')) AS T  
319 ORDER BY ENTRY_TIMESTAMP DESC;  
320
```

ENTRY_DATA	ENTRY_TIMESTAMP	JOURNAL_ENTRY_TYPE	ROW_DELETED	CURRENT_USER	QUALIFIED_JOB_NAME
005AC8C5D3C...	2017-05-14 17:43:...	DL	25	BOURGEOIS	226922/QUSER/QZDASOINIT
0078D1C5C1D...	2017-05-14 17:43:...	DL	3	BOURGEOIS	226922/QUSER/QZDASOINIT

Les exemples de services IBM i fournis avec ACS

Exemples

Recherche...

IBM i Services

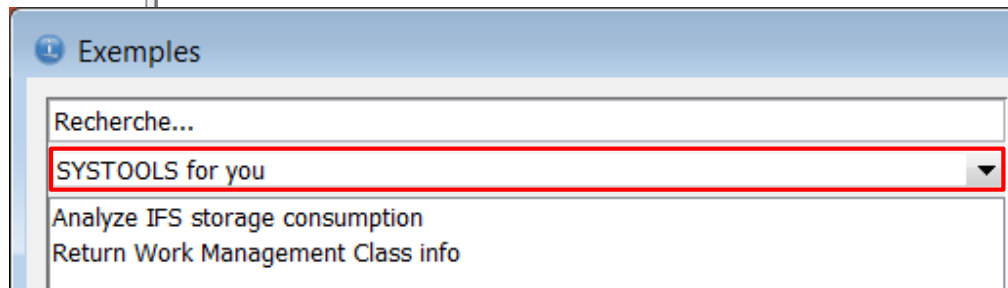
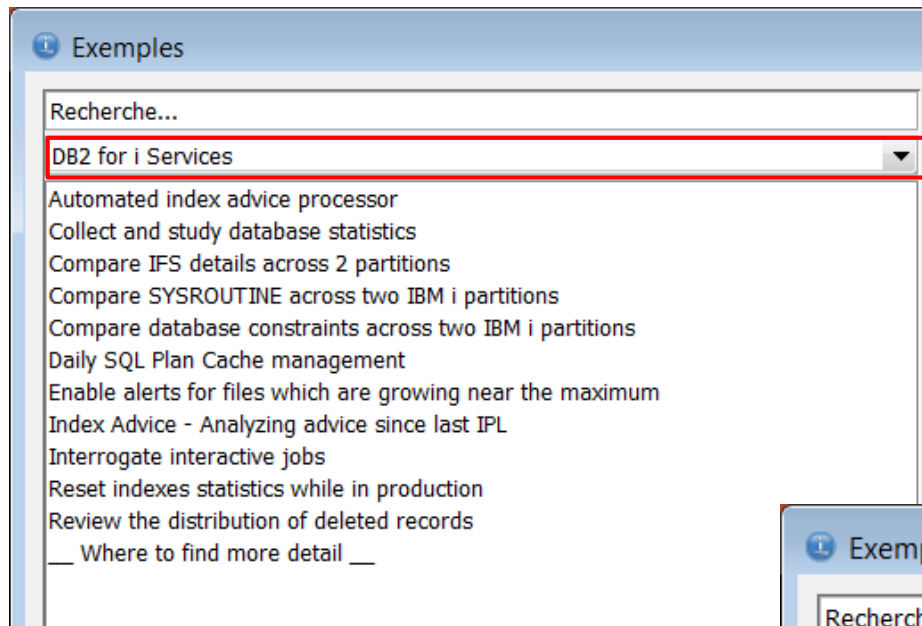
- Application - Environment variable information
- Application - Examine my stack
- Application - PASE Shell
- Application - Service tracker
- Application - Special case Data Areas
- Application - Split an aggregated list
- Application - Work with Data areas in QTEMP
- Application - Work with numeric Data areas
- Communications - Network Statistics Info (NETSTAT)
- Communications - Network Statistics Interface (NETSTAT)
- Communications - Network Statistics Job Info (NETSTAT)
- Communications - Network Statistics Route Info (NETSTAT)
- Communications - TCP/IP Information
- Communications - Time server
- Find objects in a library, not included in an authorization list
- History Log - Study job longevity
- Java - Find instances of old Java versions being used
- Java - JVM Health

- Librarian - Examine least and most popular routines
- Librarian - Find objects
- Librarian - Library list
- License Management - Expiration processing
- Message Handling - Query a message file
- Message Handling - Abnormal IPL Predictor
- Message Handling - Reply List
- Message Handling - Review system operator inquiry messages
- Message Handling - Review system operator unanswered inquiry messages
- PTF - Firmware Currency
- PTF - Group PTF Currency
- PTF - Group PTF Details
- PTF - PTF information
- Password failures over the last 24 hours
- Product - Expiring license info
- Review public authority to files in library TOYSTORE
- Review the object ownership summary for objects in a library
- Security - Authority Collection
- Security - Authority Collection
- Security - Authority Collection (analyze)
- Security - Authority Collection (capture)
- Security - Authority Collection (review)
- Security - Authorization List detail
- Security - Check authority to query
- Security - DISPLAY_JOURNAL() of the audit journal

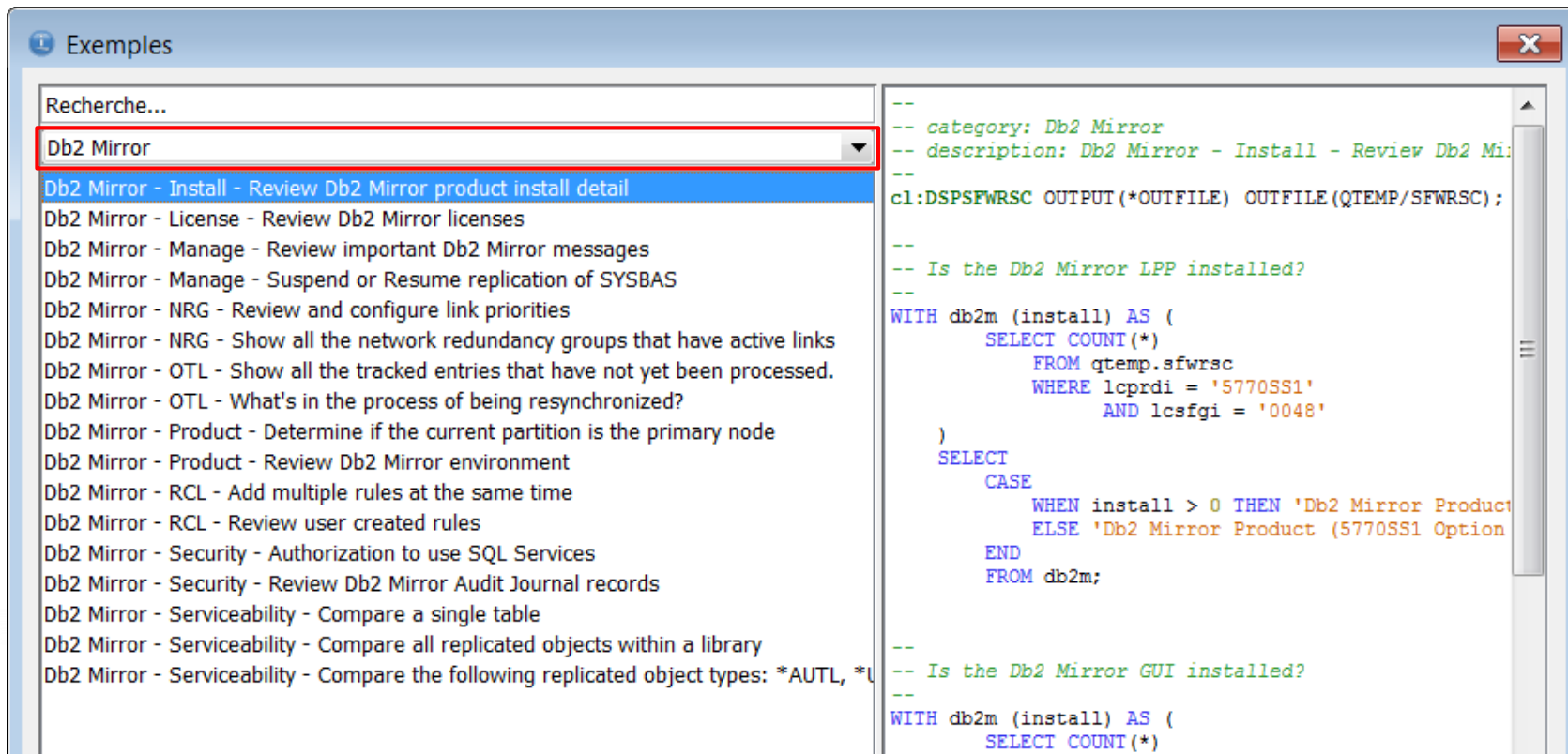
- Security - DRDA Authentication Entry info
- Security - Function Usage
- Security - Group profile detail
- Security - Review *ALLOBJ users
- Security - Review *JOBCTL users
- Security - Secure column values within SQL tooling
- Security - User Info Sign On Failures
- Security - User Info close to disabled
- Security - Which users are changing data via a remote connection?
- Spool - Consume my most recent spooled file
- Spool - Output queue basic detail
- Spool - Output queue exploration
- Spool - Search all QZDASOINIT spooled files
- Spool - Top 10 consumers of spool storage
- Storage - ASP management
- Storage - IASP Vary ON and OFF steps
- Storage - Media Library
- Storage - Review status of all storage H/W
- Storage - Storage details for a specific user
- Storage - Temporary storage consumption, by DB workload
- Storage - Temporary storage consumption, by active jobs
- Storage - Top 10 Spool consumers, by user
- Storage - Top 10 consumers, by user
- Storage - iASP storage consumption
- System Health - System Limits tracking
- System Health - Tracking the largest database tables

- What is our journaling setup, by object type
- What unofficial IBM i code exists in QSYS?
- Work Management - Active Job Info - Largest Query
- Work Management - Active Job Info - Lock contention
- Work Management - Active Job Info - Long running SQL statements
- Work Management - Active Job Info - QTEMP consumption
- Work Management - Active Job Info - Temp storage top consumers
- Work Management - Active Job info - Longest active DRDA connections
- Work Management - Active Job info - Longest running SQL statements
- Work Management - Active Job info - SQL Server Mode study
- Work Management - Active Job info - Temp storage consumers, by memory pool
- Work Management - Active Job info - Top ZDA CPU consumers
- Work Management - Interactive jobs
- Work Management - Job Description Initial Library List
- Work Management - Job Descriptions
- Work Management - Job Queues
- Work Management - Jobs that are waiting to run
- Work Management - Object lock info
- Work Management - Record lock info
- Work Management - SET_SERVER_SBS_ROUTING and ad hoc users
- Work Management - Scheduled Job Info
- Work Management - System Status
- Work Management - System Values

Les exemples de services IBM i fournis avec ACS



Les exemples de services IBM i fournis avec ACS



The screenshot shows a window titled 'Exemples' with a search bar containing 'Recherche...'. Below the search bar, a dropdown menu is open, showing a list of search results. The first result, 'Db2 Mirror', is highlighted with a red box. The rest of the list includes various tasks related to Db2 Mirror, such as license review, message management, replication control, and environment review.

On the right side of the window, a text editor displays SQL code. The code includes comments in green and SQL statements in blue and black. The visible code is as follows:

```
--  
-- category: Db2 Mirror  
-- description: Db2 Mirror - Install - Review Db2 Mi  
--  
c1:DSPSPWRSC OUTPUT(*OUTFILE) OUTFILE(QTEMP/SFWRSC);  
  
--  
-- Is the Db2 Mirror LPP installed?  
--  
WITH db2m (install) AS (  
    SELECT COUNT(*)  
        FROM qtemp.sfwrsc  
        WHERE lcpmdi = '5770SS1'  
            AND lcsfgi = '0048'  
    )  
SELECT  
    CASE  
        WHEN install > 0 THEN 'Db2 Mirror Product'  
        ELSE 'Db2 Mirror Product (5770SS1 Option'  
    END  
FROM db2m;  
  
--  
-- Is the Db2 Mirror GUI installed?  
--  
WITH db2m (install) AS (  
    SELECT COUNT(*)
```

Exemple – Services DB2

Les "objets" SQL manipulés par des programmes

```
463 SELECT PROGRAM_NAME, T.*
464 FROM QSYS2.SYSPROGRAMSTMTSTAT, TABLE(QSYS2.PARSE_STATEMENT(STATEMENT_TEXT)) AS T
465 WHERE PROGRAM_SCHEMA = 'AS425F';
466
467
```

PROGRAM_NAME	NAME_TYPE	NAME	SCHEMA	RDB	COLUMN_NAME	USAGE_TYPE	NAME_START_POSITION	SQL_STATEMENT_TYPE
AUTORENS	TABLE	TABEMPL3	AS425F	-	-	TARGET TA...	13	INSERT
AUTORENS	COLUMN	TABEMPL3	AS425F	-	NOM	TARGET TA...	33	INSERT
AUTORENS	COLUMN	TABEMPL3	AS425F	-	SEXE	TARGET TA...	39	INSERT
AUTORENS	COLUMN	TABEMPL3	AS425F	-	NUMSRV	TARGET TA...	46	INSERT
AUTORENS	COLUMN	TABEMPL3	AS425F	-	SALAIRE	TARGET TA...	55	INSERT
AUTORENS	COLUMN	TABEMPL3	AS425F	-	DATENAI	TARGET TA...	65	INSERT
BLOBRPG1	TABLE	TABEMP2	-	-	-	TARGET TA...	8	UPDATE
BLOBRPG1	COLUMN	TABEMP2	-	-	CV	TARGET TA...	20	UPDATE
BLOBRPG1	COLUMN	-	-	-	MAT	QUERY	35	UPDATE

5. Fonctions de recherche avancées

Fonctions de recherche avancées

- LIKE
 - Recherche avec jokers (% et _)
- REGEXP_LIKE
 - Recherche sur expression régulière
- SOUNDIX et DIFFERENCE
 - Recherche phonétique
- CONTAINS et SCORE
 - Recherche linguistique

Expressions régulières

- Recherche si une chaîne correspond à un **modèle**
 - `SELECT * FROM employes WHERE REGEXP_LIKE(nom, 'phil{1,2}ip{1,2}e', 'i')`
- Prérequis :
 - La chaîne doit être alphanumérique, numérique, date ou heure
 - L'option 39 de SS1 (Composants internationaux pour Unicode) doit être installée
- Fonctions **REGEXP_xxx** :
 - `REGEXP_LIKE` Cherche si une chaîne vérifie une expression régulière
 - `REGEXP_COUNT` Compte le nombre de fois qu'une expression régulière est vérifiée par une chaîne de caractères
 - `REGEXP_INSTR` Indique la position dans la chaîne de caractères qui vérifie l'expression régulière
 - `REGEXP_REPLACE` Permet de remplacer la sous-chaîne de caractères qui vérifie une expression régulière
 - `REGEXP_SUBSTR` Retourne la sous-chaîne de caractères qui vérifie l'expression régulière

Expressions régulières

- Constitution des modèles – Quelques métacaractères :

<code>\w</code>	Tout caractère alpha ou numérique
<code>\W</code>	Tout sauf caractère alpha ou numérique
<code>\d</code>	Un chiffre
<code>\D</code>	Tout sauf un chiffre
<code>\s</code>	Le caractère espace
<code>\S</code>	Tout sauf un espace
<code>[modèle]</code>	Tout caractère respectant le modèle
<code>(suite)</code>	Regroupement de caractères
<code>.</code>	Tout caractère
<code>^</code>	Début de ligne
<code>\$</code>	Fin de ligne
<code>\</code>	Désécialise le caractère suivant (<code>\.</code> pour chercher un point)

Expressions régulières

- Constitution des modèles – 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

Expressions régulières

- Constitution des modèles – Quelques exemples :

[abc]	Un a ou un b ou un c
[^abc]	Tout sauf a, b et c
[L-R]	Toute lettre comprise entre L et R
[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
abc def	a,b,(c ou d), e, f
(abc) (def)	abc ou def
abc+	abc, abcc, abccc...
(abc)+	abc, abcabc, abcabcabc...

Expressions régulières

- Exemples
 - Combien de clients n'ont pas un code postal composé de 5 chiffres ?

```
SELECT COUNT(*) FROM clients WHERE REGEXP_COUNT(codepos, '\d{5}') = 0;
```

SOUNDEX et DIFFERENCE

- **SOUNDEX** permet de coder, sur 4 caractères, la représentation phonétique (en anglais) d'une chaîne de caractères
- **DIFFERENCE** permet de connaître la proximité des SOUNDEX respectifs de 2 arguments, de 0 (minimum), à 4 (maximum)

```
SELECT nom, DIFFERENCE(nom, 'PHILIPPE')  
FROM phonetique;
```

NOM	00002
PHILIPPE	4
PHILIFE	4
PHILLIPE	4
PHILLIPPE	4
PHILIP	4
PHILIPS	3
FELIPE	3
PHILOU	3
ANNIE	1
ANDRE	0

CONTAINS et SCORE

- **CONTAINS** : recherche si une colonne contient (au niveau linguistique) une expression
- **SCORE** : renvoie la "pertinence" du contenu de la colonne par rapport à l'expression recherchée (nombre compris entre 0 et 1 avec 3 positions décimales)
- CONTAINS et SCORE sont apportées par le produit DB2 **Omnifind**
 - Produit 5733-OMF, gratuit, à installer
- Prérequis
 - L'option 39 de SS1 (Composants internationaux pour Unicode) doit être installée
 - Les tables DB2 doivent avoir une clé primaire

Le produit OMNIFIND

- Permet de faire des recherches sur le **contenu** :
 - De colonnes de tables DB2 (CHAR, VARCHAR, CLOB, BLOB, XML...)
 - De fichiers dans l'IFS (PDF, XLS, HTML, PPT, DOC...)
 - De spoules dans des OUTQ
 - De membres dans des fichiers physiques source
- Pour les recherches dans des colonnes de tables DB2 :
 - Il faut créer et alimenter un **index de texte**
 - Par l'appel de procédures stockées, scripts shell ou par IBM Navigator for i
 - Cet index n'est pas maintenu par le système
 - Maintenance différée automatique programmée
 - Ou maintenance manuelle
 - Ou maintenance automatique non programmée mais déclenchée par trigger

Recherche avancée dans DB2 – Exemples

Table **CLIENTS**

CLIENT_ID	NOM
1	Defoi
2	Defois
3	Defoie
4	Defoy
5	Defoye
6	De Foi
7	De Foy
8	De Foye
9	Defour
10	Beaumont
11	Beaux monts
12	Bomon
13	Rouget de Lille
14	Rouge et de Lille

Création et alimentation de l'index texte **Omnifind**

```
CALL SYSPROC.SYSTS_CREATE('AS425F', 'IDXTXT_CLI',  
'AS425F.CLIENTS(NOM)', 'FORMAT TEXT LANGUAGE fr_FR');
```

```
CALL SYSPROC.SYSTS_UPDATE ('AS425F', 'IDXTXT_CLI', "");
```

Exemples

NOM
Defoi
Defois
Defoie
Defoy
Defoye
De Foi
De Foy
De Foye
Defour
Beaumont
Beaux monts
Bomon
Rouget de Lille
Rouge et de Lille

- 1 `SELECT nom FROM clients WHERE UPPER(nom) LIKE ('DE%F%');`
- 2 `SELECT nom FROM clients WHERE DIFFERENCE(nom, 'defoi') >= 3;`
- 3 `SELECT nom FROM clients WHERE DIFFERENCE(nom, 'defoi') = 4;`
- 4 `SELECT nom FROM clients WHERE REGEXP_LIKE(nom, 'de.*fo[iy].*', 'i');`
- 5 `SELECT nom FROM clients WHERE SCORE(nom, 'defoi~0.2')*1000 > 2;`

1

NOM
Defoi
Defois
Defoie
Defoy
Defoye
De Foi
De Foy
De Foye
Defour

2


NOM
Defoi
Defois
Defoie
Defoy
Defoye
De Foi
De Foy
De Foye
Defour

3

NOM
Defoi
Defoie
Defoy
Defoye
De Foi
De Foy
De Foye

4

NOM
Defoi
Defois
Defoie
Defoy
Defoye
De Foi
De Foy
De Foye



5

NOM
Defoi
Defois
Defoie
Defoy
Defoye
De Foi
Defour

Exemples

NOM
Defoi
Defois
Defoie
Defoy
Defoye
De Foi
De Foy
De Foye
Defour
Beaumont
Beaux monts
Bomon
Rouget de Lille
Rouge et de Lille

```
SELECT nom FROM clients WHERE DIFFERENCE(nom, 'beaumont') >= 3;
```

NOM
Beaumont
Bomon

```
SELECT nom FROM clients WHERE SCORE(nom, 'beaumont~0.1')*1000 >= 1;
```

NOM
Beaumont
Beaux monts
Bomon




Exemples

NOM
Defoi
Defois
Defoie
Defoy
Defoye
De Foi
De Foy
De Foye
Defour
Beaumont
Beaux monts
Bomon
Rouget de Lille
Rouge et de Lille


```
SELECT nom FROM clients WHERE DIFFERENCE(nom, 'rouget de lille') >= 3;
```

NOM
Rouget de Lille
Rouge et de Lille



```
SELECT nom FROM clients WHERE SCORE(nom, 'rouge* de lille') > 0;
```

NOM
Rouget de Lille
Rouge et de Lille



Exemples

Table **PROJETS2**

COMMENTAIRE
projet à décaler
Projet décalé
Il faut que l'on décale le projet
Projet non retenu

Création et alimentation de l'index texte **Omnifind**

```
CALL SYSPROC.SYSTS_CREATE('AS425F', 'IDTXT_PRJ',  
    'AS425F.PROJETS2(COMMENTAIRE)', 'FORMAT TEXT LANGUAGE fr_FR');
```

```
CALL SYSPROC.SYSTS_UPDATE ('AS425F', 'IDTXT_PRJ', "");
```

```
SELECT commentaire FROM projets2 WHERE CONTAINS(commentaire, 'décaler') = 1;
```

COMMENTAIRE
projet à décaler
Projet décalé
Il faut que l'on décale le projet

Recherche dans des fichiers IFS avec Omnifind

-- 1. Création d'une collection texte

```
CALL SYSPROC.SYSTS_CRTCOL('PB_OMNI1', 'FORMAT TEXT');
```

```
-- CALL SYSPROC.SYSTS_CRTCOL('PB_OMNI1',
```

```
-- 'FORMAT TEXT UPDATE FREQUENCY D(*) H(*) M(0, 15, 30, 45)');
```

```
SET CURRENT SCHEMA PB_OMNI1;
```

```
SET CURRENT PATH PB_OMNI1;
```

-- 2. Indexation des fichiers des répertoires de l'IFS

```
CALL ADD_IFS_STMF_OBJECT_SET_WITH_SUBDIR('/pb/omnifind/livres');
```

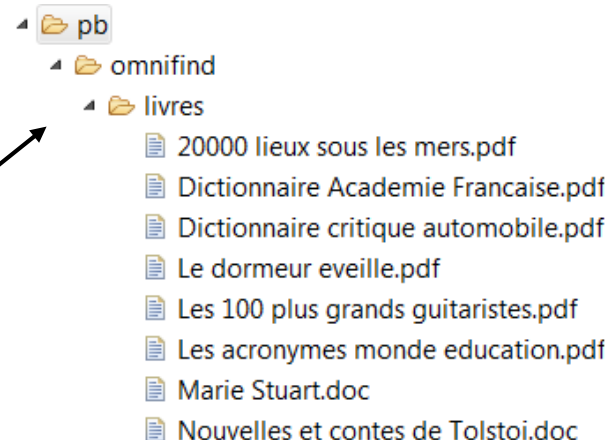
```
CALL UPDATE;
```

-- 3. Vérification

```
CALL QUERY_OBJECT_SET;
```

```
CALL GET_OBJECT_STATUS;
```

```
CALL GET_OBJECTS_NOT_INDEXED;
```



Recherche dans des fichiers IFS avec Omnifind

-- 4. Recherche

```
CALL SEARCH('capitaine');
```

-- OU

```
CALL qdbtlib.SYSTS_SEARCH_RETURN_TO_CALLER('PB_OMNI1', 'capitaine', NULL, NULL);
```

-- (nouvelle fonction 7.3 TR5 / 7.2 TR9 qui permet d'utiliser l'instruction SQL ASSOCIATE LOCATOR)

OBJECTINFOR

```
<Stream_File xmlns="http://www.ibm.com/xmlns/prod/db2textsearch/obj1"><file_path>/pb/omnifind/livres/Marie Stuart.doc</file_path></Stream_File>
```

```
<Stream_File xmlns="http://www.ibm.com/xmlns/prod/db2textsearch/obj1"><file_path>/pb/omnifind/livres/Nouvelles et contes de Tolstoi.doc</file_path></Stream_File>
```

```
<Stream_File xmlns="http://www.ibm.com/xmlns/prod/db2textsearch/obj1"><file_path>/pb/omnifind/livres/20000 lieux sous les mers.pdf</file_path></Stream_File>
```

Recherche dans des fichiers IFS avec Omnifind

-- Création d'une fonction de recherche dans l'IFS

```
CREATE or REPLACE FUNCTION search_IFS (  
  p_search_string varchar(1000),
```

...



[ps://www.ibm.com/developerworks/community/wikis/home?lang=en#!/wiki/IBM%20Technology%20Updates/page/OmniFind%20-%20SYSTS_SEARCH_RETURN_TO_CALLER%28%29](https://www.ibm.com/developerworks/community/wikis/home?lang=en#!/wiki/IBM%20Technology%20Updates/page/OmniFind%20-%20SYSTS_SEARCH_RETURN_TO_CALLER%28%29)

```
314 -- Recherche
```

```
315
```

```
316 SELECT ifs_object, modify_time FROM TABLE(search_IFS('capitaine')) AS x;
```

IFS_OBJECT	MODIFY_TIME
/pb/omnifind/livres/Marie Stuart.doc	2012-06-30 13:28:00.000000
/pb/omnifind/livres/Nouvelles et contes de Tolstoi.doc	2012-08-03 16:38:00.000000
/pb/omnifind/livres/20000 lieux sous les mers.pdf	2012-08-14 15:53:02.000000

Université IBM i

22 et 23 mai 2019

6. Divers

Récurtivité avec CONNECT BY

- Récurtivité : pour traiter des données hiérarchiques et/ou cycliques
- Deux façons de traiter la récurtivité :
 - CTE récurtives
 - Ordre **CONNECT BY**

- Table FLIGHTS pour l'exemple :

DEPARTURE	ARRIVAL
New York	Paris
Chicago	Miami
New York	London
London	Athens
Athens	Nicosia
Paris	Madrid
Paris	Cairo
Chicago	Frankfurt
Frankfurt	Moscow
Frankfurt	Beijing
Moscow	Tokyo
Frankfurt	Vienna
Paris	Rome
Miami	Lima
New York	Los Angeles
Los Angeles	Tokyo
Tokyo	Hawaii
Washington	Toronto
Cairo	Paris

Récurtivité avec CONNECT BY

```
458 SELECT CONNECT_BY_ROOT departure AS "Départ", arrival AS "Arrivée",
459 CONNECT_BY_ROOT TRIM(departure) CONCAT SYS_CONNECT_BY_PATH(TRIM(arrival), '> ') AS "Route",
460 LEVEL-1 AS "Nb d'escales"
461 FROM flights
462 WHERE CONNECT_BY_ISCYCLE = 0
463 START WITH departure = 'New York' CONNECT BY NOCYCLE departure = PRIOR arrival
464 ORDER BY LEVEL;
```

Départ	Arrivée	Route	Nb d'escales
New York	Paris	New York > Paris	0
New York	London	New York > London	0
New York	Los Angeles	New York > Los Angeles	0
New York	Rome	New York > Paris > Rome	1
New York	Cairo	New York > Paris > Cairo	1
New York	Madrid	New York > Paris > Madrid	1
New York	Athens	New York > London > Athens	1
New York	Tokyo	New York > Los Angeles > Tokyo	1
New York	Nicosia	New York > London > Athens > Nicosia	2
New York	Hawaii	New York > Los Angeles > Tokyo > Hawaii	2

OLAP – Agrégation avec la fonction LISTAGG

- Combinaison d'informations d'agrégation avec des données "détail"

MAT	NOM	SX	SRV	SAL	DAT_NAI
20	MICHEL	M	911	2140.00	1955-03-17
10	ANNIE	F	911	3080.00	1959-05-12
50	JACQUES	M	911	2360.00	1956-11-11
40	DANIELE	F	977	2810.00	1962-07-13
30	MARC	M	977	3570.00	1972-02-28
60	CLAUDE	M	990	2210.00	1988-05-02

```
471 SELECT srv AS "Service", SUM(sal) AS "Somme des salaires",  
472 LISTAGG(TRIM(nom), ', ') WITHIN GROUP(ORDER BY nom) AS "Employés"  
473 FROM tabempl GROUP BY srv;  
474
```

Service	Somme des salaires	Employés
911	7580.00	ANNIE, JACQUES, MICHEL
977	6380.00	DANIELE, MARC
990	2210.00	CLAUDE

Pour en savoir plus

- Université IBM i **2019**
 - **S25** – Les fonctions OLAP de SQL
- Université IBM i **2018**
 - **S09** – DB2 et support de JSON
 - **S10** – APIs et Services Web – Introduction
- Université IBM i **2017**
 - **S23** – La puissance des Services IBM i
 - **S34** – Manipulez du JSON et du XML en SQL

Pour en savoir plus

- <https://developer.ibm.com/articles/sql-json-publishing-functions>



The screenshot shows the IBM Developer website. At the top, there is a dark navigation bar with the text "IBM Developer" and three dropdown menus: "Topics", "Community", and "More open source at IBM". Below the navigation bar is a "Systems" section with a small icon of a server rack. On the left side, there is a sidebar menu with the following categories: "CODE" (Models, Code Patterns, Open Projects), "CONTENT" (Announcements, Articles, Courses, Series, Tutorials, Videos), and "COMMUNITY" (Events, Meetups, Webinars, Podcasts, Blogs, Newsletters, Social). The main content area is titled "ARTICLE" and features the article "SQL JSON publishing functions" by Birgitta Hauser, published on October 9, 2018. The article is categorized under "Databases" and "Systems".

Pour en savoir plus

- **Workshops IBM Lab Services**
 - IBM i - Mise en œuvre de XML et JSON – 1 jour
 - IBM i - Mise en œuvre des Services Web – 2 jours
 - IBM i - SQL avancé – 3 jours
- **Sur site client**
 - Prix à la journée quel que soit le nombre de personnes à former
 - Réduction pouvant aller jusqu'à 100% si vous disposez de vouchers IBM i
- **Dans les locaux d'IBM Bois-Colombes**
 - Prix au workshop par personne
 - Gratuit si vous disposez de vouchers IBM i

