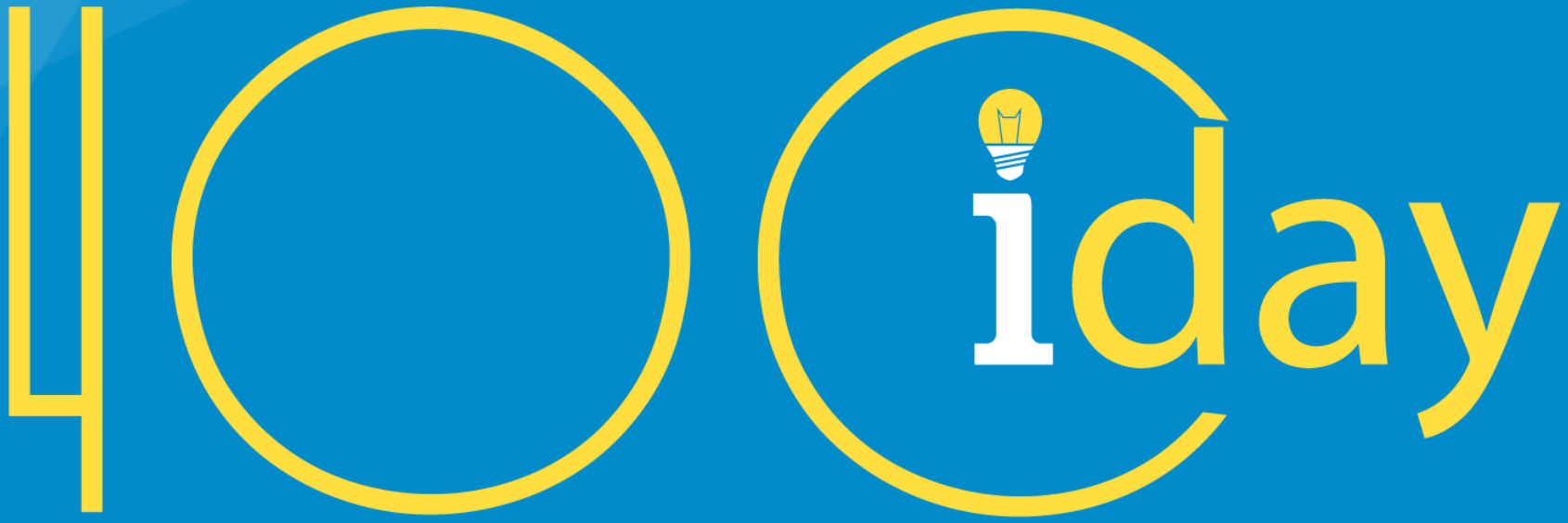


Audit/Sécurité
Tables temporelles

2 et 9 Février 2017

@GaiaFrance

www.gaia.fr
www.know400.fr

contact@gaia.fr

News !

- ACS 1.1.6.2
  - Corrections de bugs
  - Amélioration de la coloration syntaxique dans l'exécution de script SQL
  - Embarque un exécutable pour support de EHLLAPI
    - EHLLAPI bridge
    - Non documenté à l'heure actuelle …
- Serveur de web services
  - Génération de fichier swagger pour les services REST
    - Utilisables avec Swagger UI, Postman

News !

- Téléchargement automatique des Group PTF
  - Commande `CHGSRVA`, paramètres `ORDPTFAUTO` et `ORDPTFGRP`
  - http://ibmsystemsmag.com/Blogs/i-Can/December-2016/Automatic-Download-of-Group-PTFs-with-IBM-i-7-3/
- Application automatique des PTF à l'IPL
  - `PWRDWNSYS INSPTFDEV(*SERVICE)`
- IBM i strategy and roadmap
  - http://www-03.ibm.com/systems/power/software/i/smartpaper/index.html
  - http://www.ibmsystemsmag.com/Blogs/You-and-i/January-2017/IBM-i-Future-%E2%80%93-2017-Update/

Evènements

- Université IBM i
  - Paris : mi-mai sur 2 jours
  - Philippe Bourgeois
- Common
  - Romandie (Lausanne) : 2 mars
  - Europe (Bruxelles) : 18 au 21 juin
- Volubis / Pauses Café
  - Christian Massé

Audit/Sécurité

Authority Collection (audit des objets)

Rappel

- Historiquement le mécanisme d'accès aux ressources était de type contemporain.
  - Premier droit obtenu uniquement
  - Basé sur les Profils et les groupes
- Avec l'ouverture au monde UNIX, il est devenu complétif
  - C'est un cumul d'autorisations
  - Basé sur les profils additionnels
- Avec SQL a été introduit la notion d'autorisation aux niveaux des colonnes.
  - On peut voir des informations et en cacher d`autres
  - Par SQL ou HLL de manières traditionnelles

Concepts

- Les valeurs systèmes
  - Fixent le paramétrage de la machine
  - Principale valeur : QSECURITY
    - détermine le niveau de sécurité
  - Les accès aux ressources sont contrôlés à partir du niveau 30
- Le profil
  - Permet à la machine d'authentifier la personne

Concepts

- Profil de groupe
  - Référencement de profils devant avoir les mêmes droits
- Liste d'autorisations
  - Référencement de ressources qui doivent être protéger de manière identique
- Adoption de droits
  - Mécanisme permettant d'exécuter de manière temporaire un programme avec les droits du créateur de ce dernier
  - Adoption dynamique par API également possible (swap de profil)
- Dépositaire de droits
  - Permet de mettre en place des droits pour des objets qui n'existent pas encore

Le mécanisme d'accès original

Les Etapes	Ordre
Le profil est il *ALLOBJ	1
Le profil a-t-il des droits sur l'objet	2
Le profil est-il indiqué dans la liste	3
Le groupe est il *ALLOBJ	4
Le groupe a-t-il des droits sur l'objet	5
Le groupe est-il indiqué dans la liste	6
*PUBLIC est-il indiqué sur l'objet	7
*PUBLIC EST indiqué dans la liste	8

Suivi

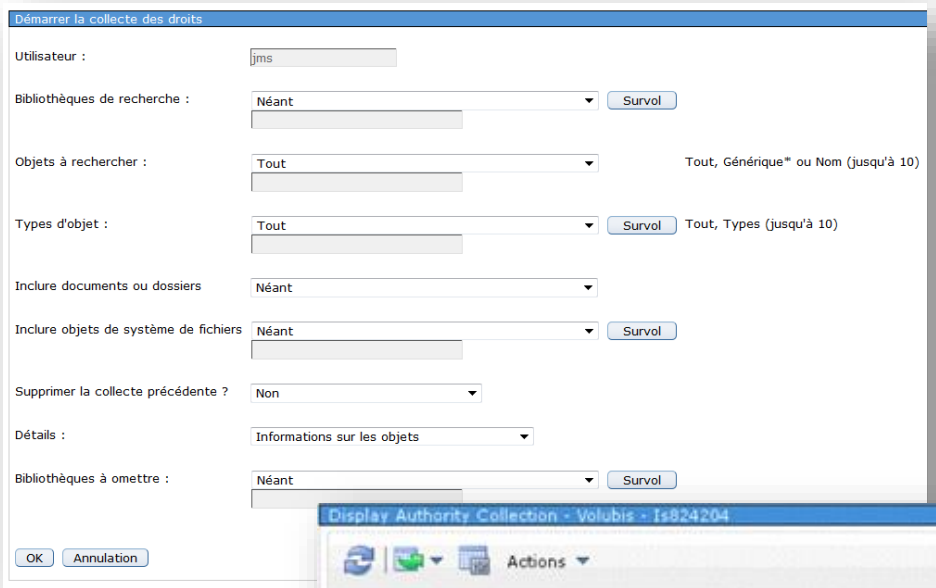
- Problématique
  - Jusqu'ici on avait un problème de suivi : on ne savait pas toujours pourquoi
    - un utilisateur accédait à un objet
    - si certains utilisateurs accédaient à des objets
- La V7R3 apporte un outil d'audit
  - Authority Collection
  - qui permet de tracer les accès pour un profil donné
- On savait à l'inverse par les audits, tracer les refus
  - De nombreux outils sont basés sur cette techno

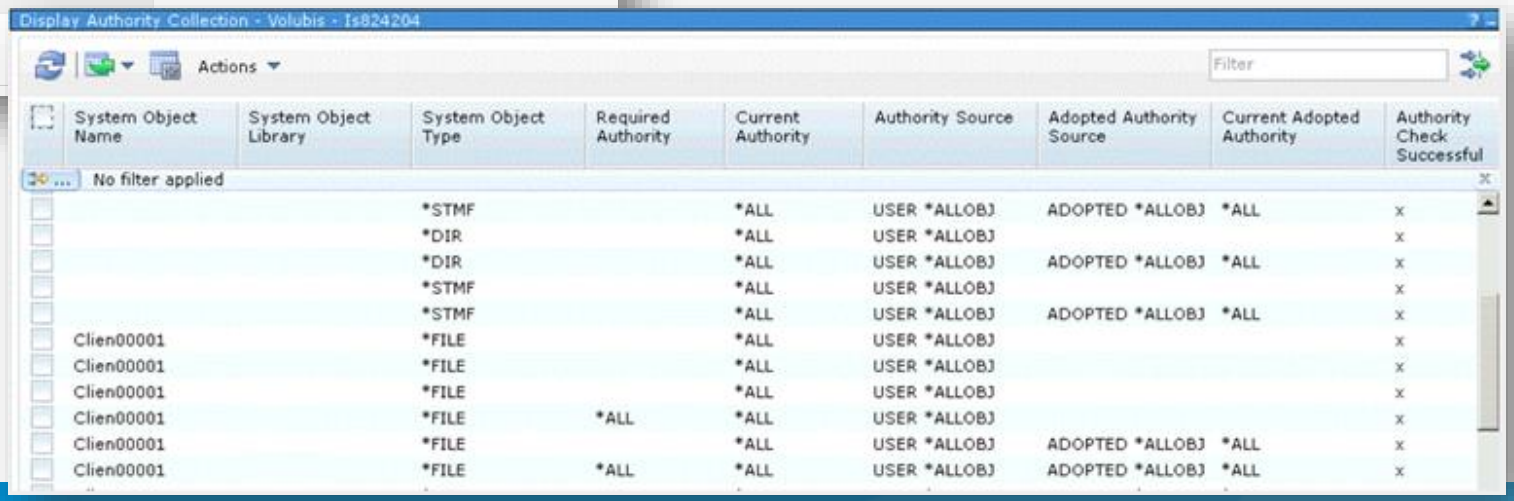
Mise en œuvre

- STRAUTCOL
  - Permet de démarrer l'audit pour un profil
- QSYS2/AUTHORITY_COLLECTION
  - Vue contenant le résultat de l'audit
- ENDAUTCOL
  - Permet de l'arrêter l'audit pour un profil
- DLTAUTCOL
  - Suppression des données d'audit collectées pour un profil

Mise en œuvre

- Accès via IBM Navigator for i





System Object Name	System Object Library	System Object Type	Required Authority	Current Authority	Authority Source	Adopted Authority Source	Current Adopted Authority	Authority Check Successful
		*STMF		*ALL	USER *ALLOBJ	ADOPTED *ALLOBJ	*ALL	x
		*DIR		*ALL	USER *ALLOBJ			x
		*DIR		*ALL	USER *ALLOBJ	ADOPTED *ALLOBJ	*ALL	x
		*STMF		*ALL	USER *ALLOBJ			x
		*STMF		*ALL	USER *ALLOBJ	ADOPTED *ALLOBJ	*ALL	x
Clien00001		*FILE		*ALL	USER *ALLOBJ			x
Clien00001		*FILE		*ALL	USER *ALLOBJ			x
Clien00001		*FILE		*ALL	USER *ALLOBJ			x
Clien00001		*FILE	*ALL	*ALL	USER *ALLOBJ			x
Clien00001		*FILE		*ALL	USER *ALLOBJ	ADOPTED *ALLOBJ	*ALL	x
Clien00001		*FILE	*ALL	*ALL	USER *ALLOBJ	ADOPTED *ALLOBJ	*ALL	x

STRAUTCOL

- Exemple :

```
Collecte des droits démarrée pour l'utilisateur PLB.
```

  - Tracer tout pour le user PLB, mais pas à chaque JOB !
  - Le système créé une *DTAARA QSYS/QP0FNOCRED

Vue AUTHORITY_COLLECTION

- Exemple :

```
SELECT SYSTEM_OBJECT_NAME, SYSTEM_OBJECT_SCHEMA,
       AUTHORITY_SOURCE
FROM QSYS2/AUTHORITY_COLLECTION
WHERE AUTHORIZATION_NAME = 'PLB'
```

SYSTEM_OBJECT_NAME	SYSTEM_OBJECT_SCHEMA	AUTHORITY_SOURCE
QBATCH	QGPL	GROUP OWNERSHIP
QBATCH	QGPL	GROUP OWNERSHIP
QCLSRC	QGPL	GROUP OWNERSHIP
QCLSRC	QGPL	GROUP OWNERSHIP
QCLSRC	QGPL	GROUP OWNERSHIP
QCLSR	QGPL	GROUP OWNERSHIP
QCLSR	QGPL	GROUP OWNERSHIP
QCLSRC	QGPL	GROUP OWNERSHIP
QCLSRC	QGPL	GROUP OWNERSHIP
QCLSRC	QGPL	GROUP OWNERSHIP
QAUOOPT	QGPL	GROUP OWNERSHIP
QGPL	QSYS	GROUP PRIVATE
QGPL	QSYS	GROUP PRIVATE

Comment on obtient l'autorisation ?

Vue AUTHORITY_COLLECTION

- Pour connaitre les profils audités !

```
SELECT AUTHORIZATION_NAME, TEXT_DESCRIPTION
FROM USER_INFO
WHERE AUTHORITY_COLLECTION_ACTIVE = 'YES'
```

DSPUSRPRF, ENDAUTCOL, DLTAUTCOL

- Nouvelles informations dans la commande

```
Collecte des droits active . . . . . :   Oui
Le référentiel de collecte des droits
  existe déjà  . . . . . :   Oui
```

  - Indique si la collecte est active et s'il existe une collecte, même inactive
- ENDAUTCOL
  - Ne supprime pas la collecte
- DLTAUTCOL
  - Supprimer la collecte

Exemples de requête

- Pour un profil

```
SELECT * FROM QSYS2/AUTHORITY_COLLECTION
WHERE USER_NAME = 'USR'
```

- Pour un objet et un profil

```
SELECT * FROM QSYS2.AUTHORITY_COLLECTION
WHERE USER_NAME = 'USR' AND
      SYSTEM_OBJECT_NAME = 'OBJETX' AND
      SYSTEM_OBJECT_SCHEMA = 'LIBX'
```

- Pour un objet

```
SELECT * FROM QSYS2.AUTHORITY_COLLECTION
WHERE SYSTEM_OBJECT_NAME = 'OBJETX' AND
      SYSTEM_OBJECT_SCHEMA = 'LIBX'
```

Restauration d'un profil

- RSTUSRPRF
  - Si vous restaurez un profil à partir de la version 7.3 vers une version au moins équivalente l'attribut de collecte sera conservé à l'identique
  - Par contre la collecte ne sera pas restaurée

Restrictions

- Pour pouvoir Administrer "Authority collection"
  - *ALLOBJ nécessaire
- Ou être autorisé à "Database Security Administrator"
  - Fonction QIBM_DB_SECADM
  - Ces fonctions sont administrables via
    - Navigator for i - Application Administration
    - WRKFCNUSG - Work with Function commande
- En cas d'IPL anormal
  - Le référentiel peut être inutilisable, l'arrêter, le supprimer et le recréer

Annexes

- Description du fichier de sortie à analyser
  - http://www.ibm.com/support/knowledgecenter/ssw_ibm_i_73/rzarl/rzarlautcolview.htm

Audit réseau

Mise en œuvre

- Mise en œuvre
  - Création du premier récepteur

```
CRTJRNRCV JRNRCV(JRNLIB/AUDRCV0001)
```

  - Création du Journal d'audit

```
CRTJRN  JRN(QSYS/QAUDJRN)
        JRNRCV(JRNLIB/AUDRCV0001)
        MNGRCV(*SYSTEM)
        DLTRCV(*NO)
        AUT(*EXCLUDE)
        TEXT('Auditing Journal')
```

Mise en œuvre

- Démarrage de l'audit
  - Valeur système QAUDCTL
- 2 valeurs principales
  - *OBJAUT pour auditer les objets et les utilisateurs
    - Vous devez ensuite déterminer le périmètre à auditer
      - CHGAUD Modifier la valeur d'audit
      - CHGDLOAUD Modifier niveau d'audit DLO
      - CHGOBJAUD Modifier l'audit d'objet
      - CHGSECAUD Modifier audit de sécurité
      - CHGUSRAUD Modifier audit d'utilisateur
  - *AUDLVL pour auditer les actions
    - A définir dans les valeurs systèmes QAUDLVL et QAUDLVL2

Mise en œuvre

- Exemple
  - `*AUTFAIL` pour les erreurs d'authentification
- Liste complète
  - https://www.ibm.com/support/knowledgecenter/ssw_ibm_i_73/rzarl/rzarlaudlev2.htm

Mise en œuvre

- Avec *AUTFAIL
  - vous obtenez des postes de TYPE PW
- que vous pouvez visualiser par
  - la commande DSPAUDJRNE, qui ne permet pas la sortie fichier
- Si vous avez besoin d'un fichier, vous devrez créer votre fichier à l'image du fichier Système ici

```
CRTDUPOBJ OBJ(QASYPWJ5)
          FROMLIB(QSYS)
          OBJTYPE(*FILE)
          TOLIB(biblio)
          NEWOBJ(objet)
```

  - Pour trouver le fichier qui vous convient

```
DSPOBJD OBJ(QSYS/QASY*)
        OBJTYPE(*FILE)
        OUTPUT(*OUTFILE)
        OUTFILE(QTEMP/LISTEAUD)
```

Exemple

```
SELECT ODOBNM, ODOBTX
FROM liste
where odobtx like('%type PW%')
```

- remplacer par votre type
  - `Object` Text description
  - `QASYPWJE` Outfile for journal entry type `PW`
  - `QASYPWJ4` Outfile for journal entry type `PW`
  - `QASYPWJ5` Outfile for journal entry type `PW`
  - Le chiffre 4 ou 5 indiquant le niveau de détails

Extraction

- Vous remplissez votre fichier de travail

```
DSPJRN JRN(QAUDJRN)
       ENTTYP(PW)
       OUTPUT(*OUTFILE)
       INCHIDENT(*YES)
       OUTFILFMT(*TYPE5)
       OUTFILE(biblio/fichier)
```

  - Pour requêter vous pouvez utiliser n'importe quel outil SQL

Directement par SQL

```
SELECT *
FROM TABLE (
  QSYS2.Display_Journal( 'biblio', 'journal') )
    AS ANALYSE_JRN
WHERE journal_entry_type = ('PW')
```

- Attention le poste est donné brut
  - Permet une recherche rapide !

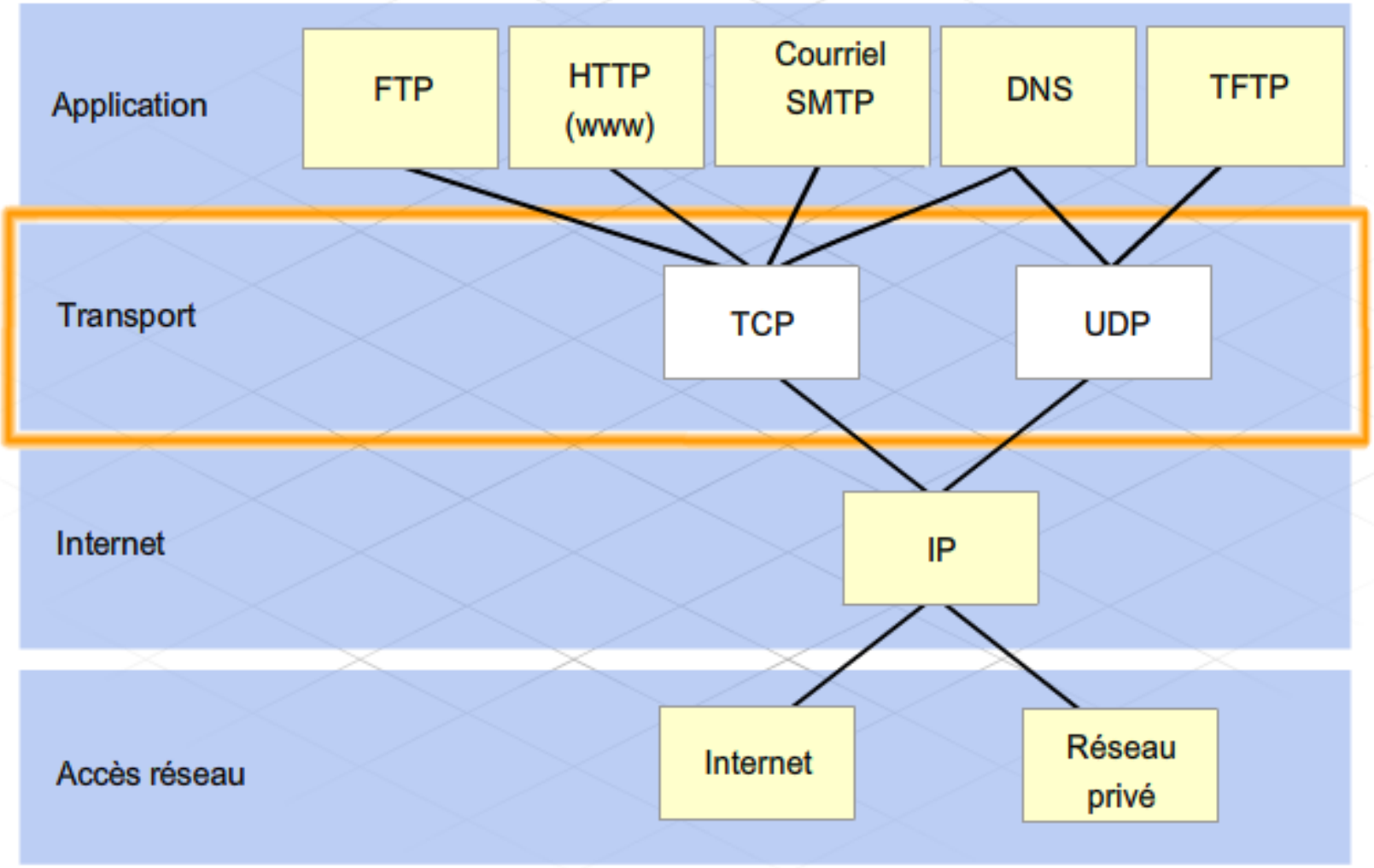
Valeurs Système QAUDLVL2

- Avec la version 7.3 : 3 nouvelles options d'audit du réseau
  - *NETUDP pour auditer le trafic UDP
  - *NETTELSVR pour auditer le flux telnet
  - *NETSECURE pour auditer les connexions sécurisées
- La valeur suivante existait déjà mais a été enrichie
  - *NETSCK pour auditer le trafic TCP
    - trace maintenant en plus SMTP et DHCP

```
Valeur système . . . . :   QAUDLVL2
Description  . . . . . :   Extension du niveau d'audit de sécurité


Options                                        Options
audit                                          audit
*NETSCK
*NETUDP
*NETTELSVR
*NETSECURE
```

Rappel TCP/IP



Rappel TCP/IP

- Protocole UDP
  - UDP est un protocole stateless (sans état)
  - on peut le comparer au courrier
  - TFTP, DNS, etc...
- Protocole TCP
  - TCP est un protocole stateful (avec état)
  - Fonctionne un peu comme le téléphone
    - il faut d'abord établir une connexion TCP entre les 2 hosts
    - cette connexion est un socket
  - FTP, HTTP, SMTP, TELNET etc…

Rappel TCP/IP

- SSL / TLS (Secure Socket Layer / Transport Layer Security)
  - C'est le protocole de sécurité le plus répandu qui créé un canal sécurisé entre deux machines
  - Souvent utilisé pour accéder a des ressources externe au LAN

Tableau des valeurs systèmes

- Valeurs d'audit des sockets de connexion

Valeur d'audit	Description	QAUDLVL QAUDLVL2	CHGUSRAUD	Entrée SK	Description détaillée
*NETSCK	Connexions TCP auditées	✔	✔	A C	Socket de connexion TCP accepté (**A**) ou connexion établie (**C**)
*NETUDP	Audit trafic UDP	✔	✔	I O	Un paquet UDP entrant (**I**nbound) ou sortant (**O**utband) a été reçu
*NETTELSVR	Audit des connexions Telnet	✔	✘	A	Une connexion Telnet entrante a été acceptée
*NETSECURE	Audit des connexions sécurisées	✔	✔	S X	Une connexion sécurisée a été négociée avec succès (**S**) ou a échoué (**X**)

Nature de la sortie

- Le type de poste généré est SK
  - Sockets Connections
- Le fichier modèle pour la sortie
  - QASYSKJ4/J5
- Exemple
  - type 5

```
DSPJRN JRN(QAUDJRN)
       ENTTYP(SK)
       OUTPUT(*OUTFILE)
       INCHIDENT(*YES)
       OUTFILFMT(*TYPE5)
       OUTFILE(biblio/fichier)
```

Annexes

Action auditing value	Description	Detailed SK journal entry type	Detailed description
*NETSCK	TCP connections are audited. Note: Telnet server connections are not audited.	A	Accept - A TCP socket connection was accepted.
		C	Connect - A TCP socket connection was established.
*NETUDP	UDP traffic is audited.	I	Inbound - An inbound UDP packet was received.
		O	Outbound - An outbound UDP packet was sent.
*NETTELSVR	Telnet server connections are audited.	A	Accept - An inbound Telnet connection was accepted.
*NETSECURE	Secure socket connections are audited.	S	Success - A secure connection was negotiated successfully.
		X	Fail - A secure connection failed to negotiate.

Routage des travaux serveurs

Principe

- Depuis la version 7.1
  - La procédure SQL `SET_SERVER_SBS_ROUTING` permet de router des travaux servers vers un sous système particulier
  - de nouveaux servers sont routables (voir tableau ci joint en version 7.3)
- Et vous pouvez indiquer une action dans le cas ou le sous-système indiqué serait indisponible
  - Soit démarrer dans le sous-système par défaut (valeur par défaut)
  - Soit ne pas démarrer le job

Listes des serveurs

Table 1. Servers and default subsystems

Server Description	Server Name	Default subsystem
Central server	QZSCSRVS	QUSRWRK
Database server	QZDASOINIT	QUSRWRK
Data queue server	QZHQSSRV	QUSRWRK
DDM	QRWTSRVR	QUSRWRK
DRDA	QRWTSRVR	QUSRWRK
File server	QPWFSERVSO	QSERVER
Network print server	QNPSERVS	QUSRWRK
Remote command server	QZRCSRVS	QUSRWRK

Routages

- Pour voir les routages existants : vue `QSYS2.SERVER_SBS_ ROUTING`
- Nouvelles zones
  - `QZHQSSRV_SUBSYSTEM`
  - `QZSCSRVS_SUBSYSTEM`
  - `QNPSERVS_SUBSYSTEM`
  - `QPWFSERVSO_SUBSYSTEM`
  - `QRWTSRVR_ROLLOVER`
  - `QZDASOINIT_ROLLOVER`
  - `QZRCSRVS_ROLLOVER`
  - `QZHQSSRV_ROLLOVER`
  - `QZSCSRVS_ROLLOVER`
  - `QNPSERVS_ROLLOVER`
  - `QPWFSERVSO_ROLLOVER`
- Exemple
  - `SELECT * FROM QSYS2.SERVER_SBS_ROUTING`

Exemples

- Voir un routage pour un user

```
SELECT substr(AUTHORIZATION_NAME, 1, 10) as user,
       QZDASOINIT_SUBSYSTEM, QZDASOINIT_ROLLOVER
FROM QSYS2.SERVER_SBS_ROUTING
WHERE = AUTHORIZATION_NAME = 'PLB'
```

- Voir les routages pour un job

```
SELECT substr(AUTHORIZATION_NAME, 1, 10) as user,
       QZDASOINIT_SUBSYSTEM, QZDASOINIT_ROLLOVER
FROM  QSYS2.SERVER_SBS_ROUTING
WHERE QZDASOINIT_SUBSYSTEM <> ' '
```

USER	QZDASOINIT_SUBSYSTEM	QZDASOINIT_ROLLOVER
FORM01	QBATCH	YES
FORM02	QBATCH	YES
PLB	QBATCH	YES

Suppression

- Pour supprimer toutes les entrées d'un profil

```
CALL QSYS2/SET_SERVER_SBS_ROUTING ('profil', '*ALL', '')
```

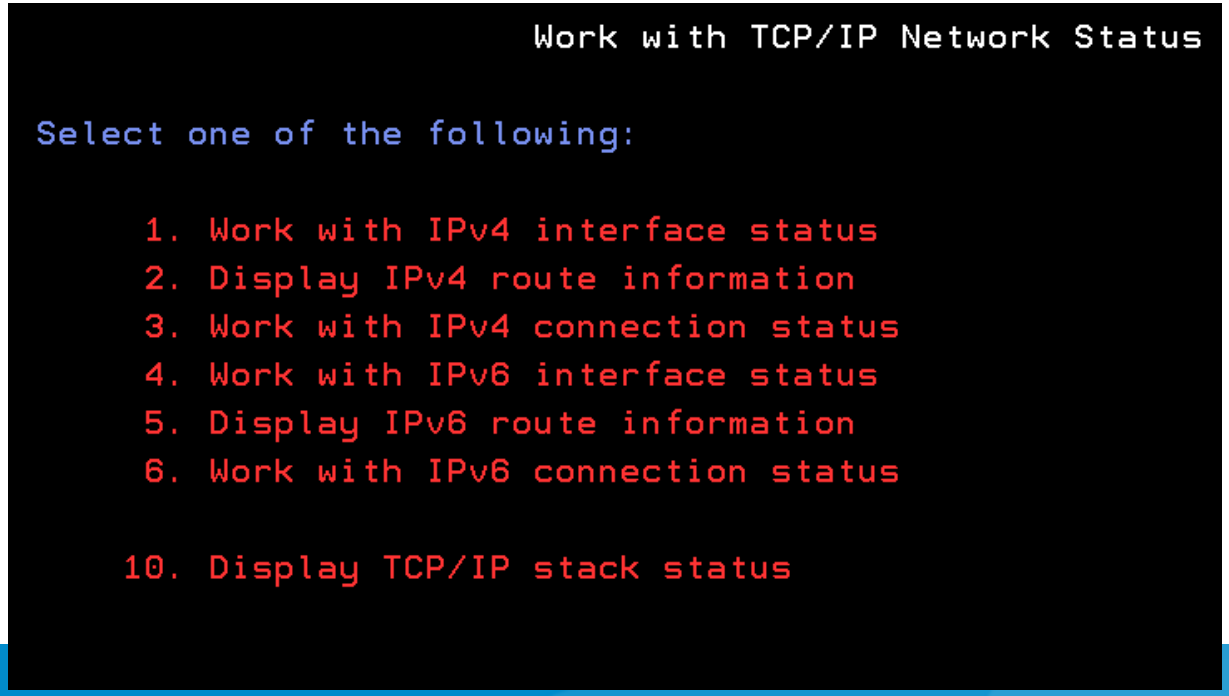
Exemple

- Création environnement étanche pour job OBBC
  - CHGSHRPOOL POOL(*SHRPOOL10) SIZE(10000) ACTLVL(5) PAGING(*CALC) TEXT('ODBC FOR SERVERS')
  - CRTSBSD SBSD(*biblio*/ODBCSVR) POOLS((1 *SHRPOOL16)) TEXT('ODBC FOR SERVERS')
  - CRTJOBD JOBD(*biblio* /ODBCSVR) TEXT('ODBC FOR SERVERS') RTGDTA(ODBCSVR) INLLIBL(VOTREBD QGPL QSYS)
  - CRTJOBQ *biblio* /ODBCSVR TEXT('ODBC FOR SERVERS')
  - ADDJOBQE SBSD(*biblio* /ODBCSVR) JOBQ(*biblio* /ODBCSVR) MAXACT(99) SEQNBR(10)
  - CRTCLS CLS(*biblio* /ODBCSVR) RUNPTY(55) TIMESLICE(100) TEXT('ODBC FOR SERVERS')
  - ADDRTGE SBSD(*biblio* /ODBCSVR) SEQNBR(10) CMPVAL('ODBCSVR') PGM(QCMD)
  - ADDPJE SBSD(*biblio* /ODBCSVR) PGM(QSYS/QZDASOINIT) JOBD(*biblio* /ODBCSVR) CLS(*biblio* /ODBCSVR)
  - STRSBS SBSD(*biblio* /ODBCSVR)
  - runsql sql('CALL QSYS2.SET_SERVER_SBS_ROUTING(''ODBCSVRU'',''QZDASOINIT'',''ODBCSVR'')') COMMIT(*NONE)

Netstat

Usage

- L'outil Netstat permet d'avoir des informations sur les connexions en cours
  - Pendant longtemps on devait utiliser la commande NETSTAT
    - permet d'avoir ces informations
    - il était difficile de faire des statistiques, des rapports, voir des actions



Depuis la version 7.1

- Il existe des vues dans la bibliothèque QSYS2
  - NETSTAT_INFO
  - NETSTAT_JOB_INFO
  - NETSTAT_ROUTE_INFO
  - NETSTAT_INTERFACE_INFO

En V7R3

- Ces vues on été complétées
  - Nouvelle colonne dans la vue `NETSTAT_JOB_INFO`
  - `JOB_TYPE` qui contient le type de travail
    - `AUTOSTART` The job is an autostart job
    - `BATCH` The job is a batch job
    - `INTERACTIVE` The job is an interactive job
    - `MONITOR` The job is a subsystem monitor job
    - `READER` The job is a spooled reader job
    - `SCPF` The job is the SCPF system job
    - `SYSTEM` The job is a system job
    - `WRITER` The job is a spooled writer job

En V7R3

- Nouvelle colonne dans la vue `NETSTAT_INFO`
- `PROTOCOL` : deux valeurs possibles
  - `TCP`
  - `UDP`

Exemples de requête

- Pour les jobs interactifs en cours

```
SELECT substr(REMOTE_ADDRESS, 1, 15) as REMOTE_ADDRESS,
       AUTHORIZATION_NAME, JOB_NAME
FROM QSYS2.NETSTAT_JOB_INFO
WHERE JOB_TYPE = 'INTERACTIVE'
```

REMOTE_ADDRESS	AUTHORIZATION_NAME	JOB_NAME
192.168.5.51	FA	168015/FA/QPADEV000K
192.168.5.51	FA	168033/FA/QPADEV000M
192.168.5.51	FA	167963/FA/QPADEV000Q
192.168.5.51	QSECOFR	168031/QSECOFR/QPADEV0008
192.168.5.51	PLB	168047/PLB/QPADEV0009
192.168.5.51	FA	168048/FA/QPADEV000C

- Pour les sessions d'un travail

```
SELECT substr(REMOTE_ADDRESS) as REMOTE_ADDRESS
       JOB_NAME, AUTHORIZATION_NAME
FROM QSYS2.NETSTAT_JOB_INFO
WHERE JOB_NAME like ('%QRWTSRVR%')
```

Exemples de requête

- Pour les sessions de l'utilisateur PLB

```
SELECT substr(REMOTE_ADDRESS, 1, 15) as REMOTE_ADDRESS,
       JOB_NAME, JOB_TYPE
FROM QSYS2.NETSTAT_JOB_INFO
WHERE AUTHORIZATION_NAME = 'PLB'
```

- Pour voir les connexions établies depuis un host

```
SELECT *
FROM QSYS2.NETSTAT_INFO
WHERE REMOTE_ADDRESS = '192.168.253.151'
```

- Pour connaitre les passerelles

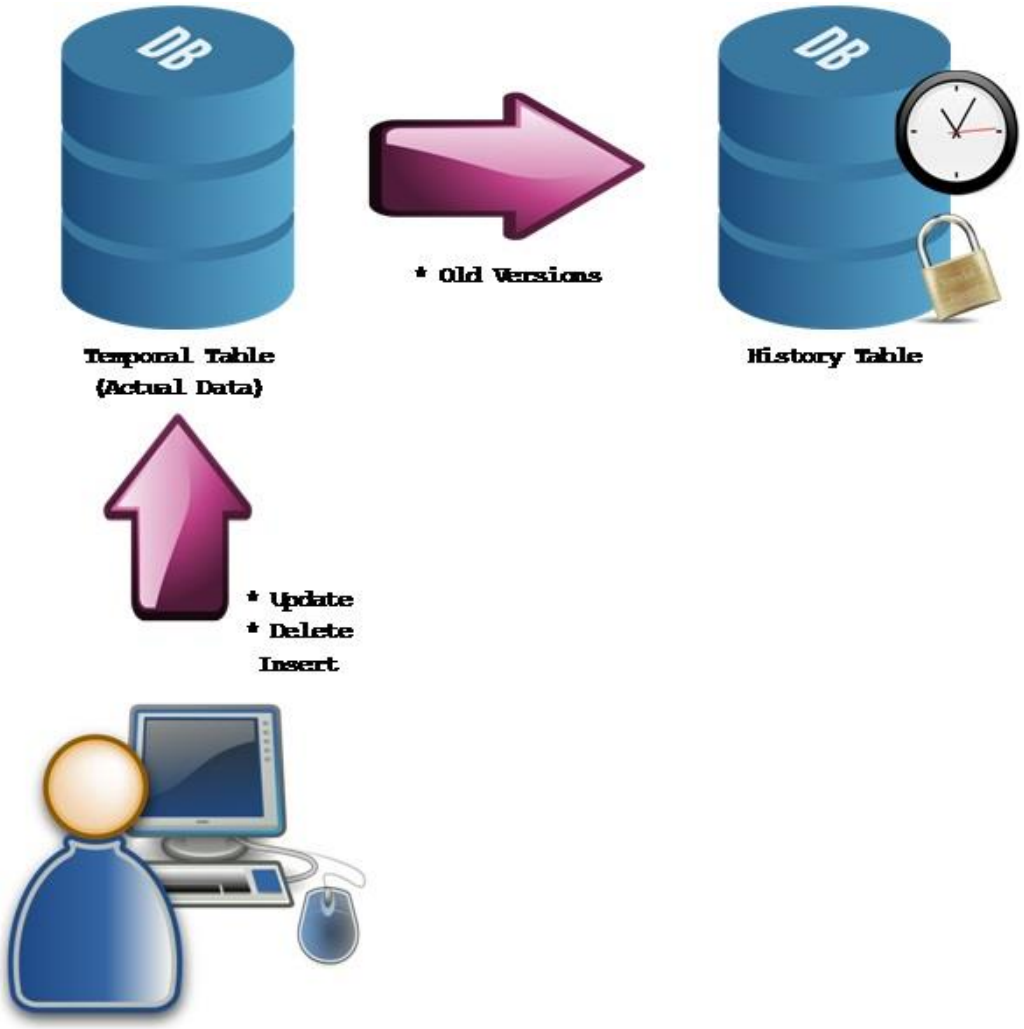
```
SELECT *
FROM QSYS2.NETSTAT_ROUTE_INFO
```

Tables temporelles

Introduction

- Table temporelle sur une période système
  - Table maintenant les versions datées des valeurs
    - Elle contient la version en cours des données
    - Elle stocke les versions précédentes (supprimées, mises à jour) dans une table d'historisation associée
  - Créée uniquement via SQL
    - Mais peut être un fichier créé via DDS et modifié via SQL
    - Table classique mais avec des colonnes ayant des définitions particulières

Schéma de principe



Mise en oeuvre

Table *base*

- Tables temporelles

```
create or replace table clients(
  nocli int as identity primary key,
  nomcli char(50),
  depcli dec(2, 0),
  datcrt date not null with default current date,
  usrcrt varchar(128) generated always as (session_user),
  action char(1) generated always as (data change operation),
  debut timestamp(12) not null generated always as row begin,
  fin timestamp(12)   not null generated always as row end,
  tsid timestamp(12)  generated always as transaction start id,
  period system_time (debut , fin) ) ;
```

Table *historique*

- Création de la table historique

```
create or replace table clients_histo like clients ;
```

- Liaison

```
alter table clients add versioning use history table
clients_histo on delete add extra row ;
```

- Arrêt de l'historisation

```
alter table clients drop versioning ;
```

Détail des colonnes

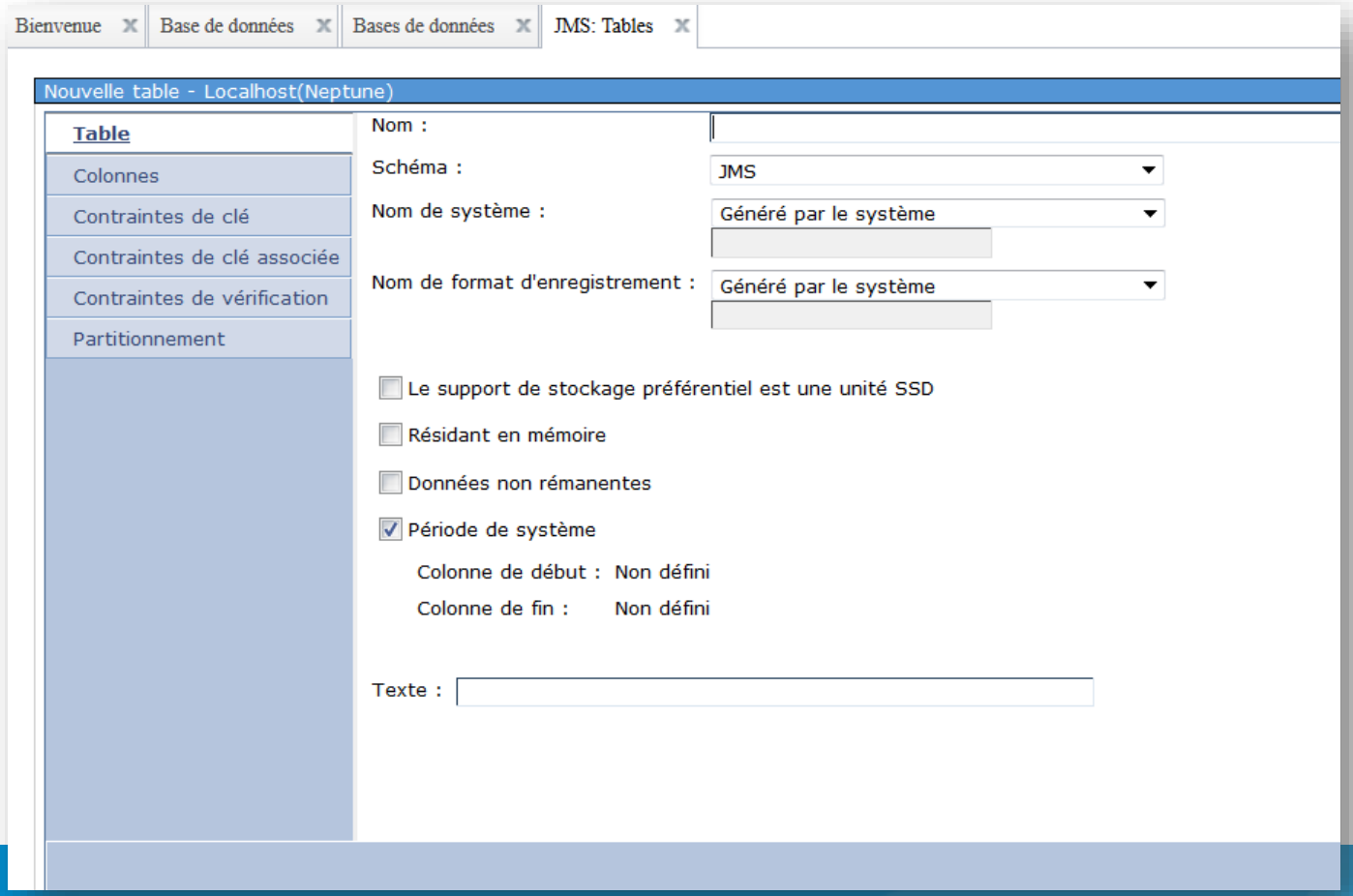
- **`Row-begin`**
  - instant ou la donnée devient la version courante. Dans le cas où une table est modifiée par **ALTER TABLE** pour intégrer la gestion des versions de ses enregistrements la valeur est par défaut : `0001-01-01-00.00.00.000000000000`
  - Donnée identique si plusieurs lignes générées par une requête de masse
  - `Row-begin` doit être inférieure à `row-end.`
- **`Row-end`**
  - instant où la donnée n'est plus la version en cours. Par défaut : `9999-12-30-00.00.00.000000000000`. A ce moment la donnée est transférée à la table d'historisation.
  - Donnée identique si plusieurs lignes générées par une requête de masse
- **`Transaction ID`**
  - Date de la première opération de modification de données dans la transaction.
- **`SYSTEM_TIME`**
  - Intervalle de temps pendant lequel une donnée était courante.
- **`Change-op`**
  - Opération ayant effectuée la modification de la donnée
    - I : ajout/insert
    - U : Mise à jour/Update
    - D : suppression/Delete

ON DELETE ADD EXTRA ROW

- Lors du ALTER TABLE…ADD VERSIONING
  - Permet de provoquer l'ajout d'un enregistrement dans la table **historique** lorsqu'une ligne est supprimée.
  - Sans cela, l'opération de suppression ne serait pas insérée dans la table **historique**.

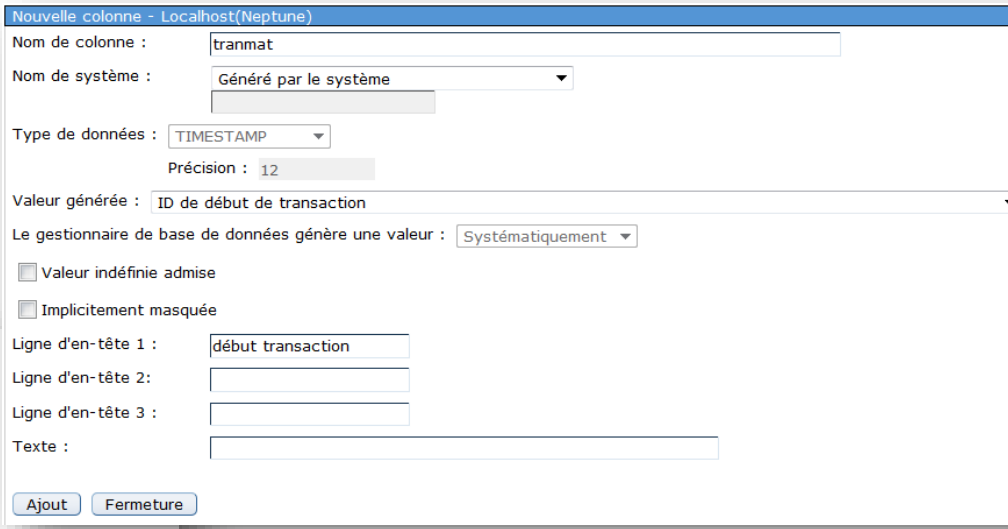
Via Navigator for i 1/2

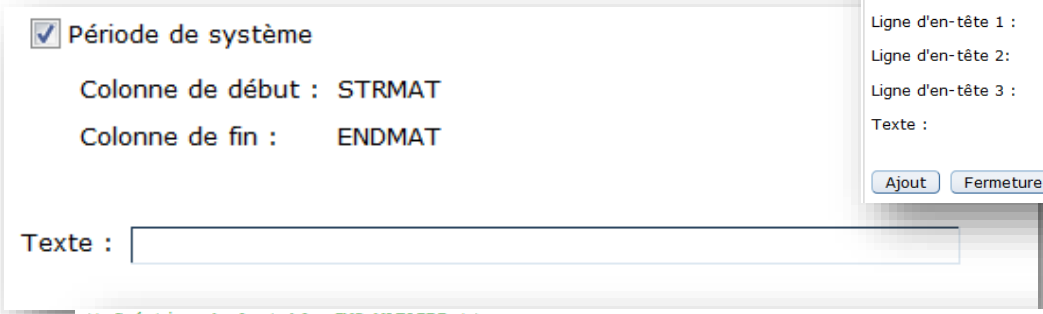
- A la création d'une table
  - Les colonnes obligatoire ne sont pas encore créées



Via Navigator for i 2/2

- Création des colonnes
  - Création des colonnes
    - Row-begin, Row-end…






```
/* Création de la table JMS.MATAIRP */
CREATE TABLE JMS.MATAIRP (
IDMAT INTEGER GENERATED ALWAYS AS IDENTITY (START WITH 1, INCREMENT BY 1, NO ORDER, NO CYCLE, NO MINVALUE, NO MAXVALUE, CACHE 20)  NOT NULL IMPLICITLY HIDDEN ,
DESCMAT CHARACTER(50) NOT NULL NOT HIDDEN ,
IMMAT CHARACTER(10) NOT NULL NOT HIDDEN ,
ENERGMAT CHARACTER(3) NOT NULL NOT HIDDEN ,
STRMAT TIMESTAMP(12) GENERATED ALWAYS AS ROW BEGIN NOT NULL NOT HIDDEN ,
ENDMAT TIMESTAMP(12) GENERATED ALWAYS AS ROW END NOT NULL NOT HIDDEN ,
TRANMAT TIMESTAMP(12) GENERATED ALWAYS AS TRANSACTION START ID NOT NULL NOT HIDDEN ,
PERIOD SYSTEM_TIME (STRMAT, ENDMAT)  ) NOT VOLATILE UNIT ANY KEEP IN MEMORY NO ;
/* Définition d'en-têtes de colonne pour JMS.MATAIRP */
LABEL ON COLUMN JMS.MATAIRP ( IDMAT IS 'Code ID' ,
DESCMAT IS 'Description matériel' ,
IMMAT IS 'Immatriculation matériel' ,
ENERGMAT IS 'Energie materiel' ,
STRMAT IS 'Horodatage courant ' ,
ENDMAT IS 'Horodatage fin ' ,
TRANMAT IS 'début transaction'  );
```

Usage

Requêtes sur les tables temporelles

- Ajout de quelques clients, puis modification du client n°1

```
select nocli, action, debut, fin from clients

NOCLI  ACTION  DEBUT                          FIN
 1     U       2016-02-11-14.18.48.915480000244  9999-12-30-00.00.00.000000000000
 2     I       2016-02-08-14.30.33.608585000244  9999-12-30-00.00.00.000000000000
 3     I       2016-02-08-14.30.58.621274000244  9999-12-30-00.00.00.000000000000
 4     I       2016-02-08-14.31.14.640404000244  9999-12-30-00.00.00.000000000000


select nocli, action, debut, fin from clients_histo

NOCLI ACTION   DEBUT                          FIN
 1     I       2016-02-08-14.29.58.915480000244  2016-02-11-14.18.48.915480000244
```

Requêtes sur les tables temporelles

- Les clients comme si on était le 10/02 à 12h (les deux tables sont lues)

```
select * from clients
for system_time as of '2016-02-10-12.00.00.00000000000'
```

- Les clients entre 1/02 et 10/02

```
select * from clients
for system_time from '2016-02-01-00.00.00.00000000000'
                  to '2016-02-10-23.59.59.00000000000'
```

- Bornes incluse

```
select * from clients
for system_time between '2016-02-01-00.00.00.00000000000'
                    and '2016-02-10-23.59.59.00000000000'
```

Requêtes sur les tables temporelles

- Modification du registre `CURRENT TEMPORAL SYSTEM_TIME`

```
SET CURRENT TEMPORAL SYSTEM_TIME = CURRENT TIMESTAMP - 1 YEAR

SELECT * FROM clients
```

- Identique à

```
SELECT * FROM clients
FOR SYSTEM_TIME AS OF CURRENT TEMPORAL SYSTEM_TIME;
```

Requêtes sur les tables temporelles

- Clause FROM
  - **… FROM table FOR SYSTEM_TIME…**
    - **AS OF *valeur***
    - Inclure les lignes dont la valeur de **row-begin** est inférieure ou égale à ***valeur***
    - **FROM *valeur1* TO *valeur2***
    - Inclure les lignes dont la valeur de **row-begin** est inférieure à ***valeur2*** et **row-end** supérieure à ***valeur1***
    - **BETWEEN *valeur1* AND *valeur2***
    - Inclure les lignes dont la valeur de **row-begin** est inférieure ou égale à ***valeur2*** et **row-end** supérieure à ***valeur1*** (valeur nulle non prise en compte)

- Une ligne peut apparaitre plusieurs fois si elle a été modifiée plusieurs fois (et éventuellement supprimée)

QAQQINI : SYSTIME_PERIOD_ADJ

- Conduite à tenir lorsque row-end est inférieure à row-begin
  - `*ERROR` (défaut)
    - Code erreur `SQLCODE -20528 / SQLSTATE 57062`
  - `*ADJUST`
    - Ajustement de `row-end` et `row-begin` pour résoudre le problème
    - Un problème se posera dans ce cas avec transaction `start-ID` qui diffèrera de `row-begin`

Vue

- Utilisation dans des vues

```
create view clients2017 as
  select * from clients for system_time
  between '2017-01-01 00:00:00.000000000000' and
          '2017-12-31 23:59:59.999999999999'
```

Autre exemple

- Comparaison du fichier clients à deux dates distinctes

```
select c1.*, c2.*
from clients for system_time as of
  '2017-01-01' c1
full outer join clients for system_time as of
  '2017-02-01' c2
on c1.nocli = c2.nocli  ;
```

- Toutes les versions du client n°1

```
select *
from clients for system_time
  between '0001-01-01' and '9999-12-31'
where nocli = 1 ;
```

RPG

Comportement

- Un programme RPG (CL, COBOL, …) ne voit aucune différence avec une table (fichier) classique
  - La temporalité ne modifie
    - Ni l'ID de niveau de format
    - Ni l'ID de niveau de fichier
- RPG considère un fichier comme un buffer
  - Les données ne sont pas contrôlées
  - Un buffer contenant l'ensemble de l'enregistrement est « poussé » vers le fichier
- Impacts
  - Ignore les valeurs par défaut et colonnes cachées
  - Les colonnes générées (`GENERATED ALWAYS AS`) sont gérées

Exemple

```
ctl-opt option(*nodebugio) alwnull(*usrctl) actgrp(*new) ccsid(*char:*jobrun);

// fichier client
dcl-f clients keyed usage(*update:*output) rename( clients : fmtcli ) ;

// variables
dcl-ds enreg likerec(fmtCli:*all) ;
dcl-ds cle   likerec(fmtCli:*key) ;

// Lecture ------------------------------------------------------------------
cle.NOCLI = 1 ;
chain cle.NOCLI fmtCli enreg ;
if %found( clients ) ;
  dsply enreg.NOMCLI ;
else ;
  dsply 'non trouve' ;
endif ;

// Maj de l'enregistrement --------------------------------------------------
enreg.NOMCLI = %trim( enreg.NOMCLI ) + ' -- modif GESTCLI' ;
update fmtcli enreg ;

// Copie de l'enregistrement ------------------------------------------------
enreg.NOCLI = 1 ;
enreg.NOMCLI = 'Créé par GESTCLI' ;
write fmtcli enreg ;
```

Résultats

- Fichier `CLIENTS` avant exécution

NOCLI	NOMCLI	DEPCLI	DATCRT	USRCRT	ACTION	DEBUT	FIN	TSID
1	IBM France	69	2017-02-01	NB	I	2017-02-01 14:44:18.855777000244	9999-12-30 00:00:00.000000000000	-
2	Global Knowledge	75	2017-02-01	NB	I	2017-02-01 14:44:18.865301000244	9999-12-30 00:00:00.000000000000	-
3	Colruyt / Pro à Pro	82	2017-02-01	NB	I	2017-02-01 14:44:18.865639000244	9999-12-30 00:00:00.000000000000	-
4	April	69	2017-02-01	NB	I	2017-02-01 14:44:18.865832000244	9999-12-30 00:00:00.000000000000	-
5	Jaillance	26	2017-02-01	NB	I	2017-02-01 14:44:18.866027000244	9999-12-30 00:00:00.000000000000	-
6	Atos	69	2017-02-01	NB	I	2017-02-01 14:44:18.866211000244	9999-12-30 00:00:00.000000000000	-

- Fichiers `CLIENTS` et `CLIENTS_HISTO` après exécution

NOCLI	NOMCLI	DEPCLI	DATCRT	USRCRT	ACTION	DEBUT	FIN	TSID
1	IBM France -- modif GESTCLI	69	2017-02-01	NB	U	2017-02-01 14:45:02.934758000244	9999-12-30 00:00:00.000000000000	-
2	Global Knowledge	75	2017-02-01	NB	I	2017-02-01 14:44:18.865301000244	9999-12-30 00:00:00.000000000000	-
3	Colruyt / Pro à Pro	82	2017-02-01	NB	I	2017-02-01 14:44:18.865639000244	9999-12-30 00:00:00.000000000000	-
4	April	69	2017-02-01	NB	I	2017-02-01 14:44:18.865832000244	9999-12-30 00:00:00.000000000000	-
5	Jaillance	26	2017-02-01	NB	I	2017-02-01 14:44:18.866027000244	9999-12-30 00:00:00.000000000000	-
6	Atos	69	2017-02-01	NB	I	2017-02-01 14:44:18.866211000244	9999-12-30 00:00:00.000000000000	-
7	Créé par GESTCLI	69	2017-02-01	NB	I	2017-02-01 14:45:02.951515000244	9999-12-30 00:00:00.000000000000	-

NOCLI	NOMCLI	DEPCLI	DATCRT	USRCRT	ACTION	DEBUT	FIN	TSID
1	IBM France	69	2017-02-01	NB	I	2017-02-01 14:44:18.855777000244	2017-02-01 14:45:02.934758000244	-

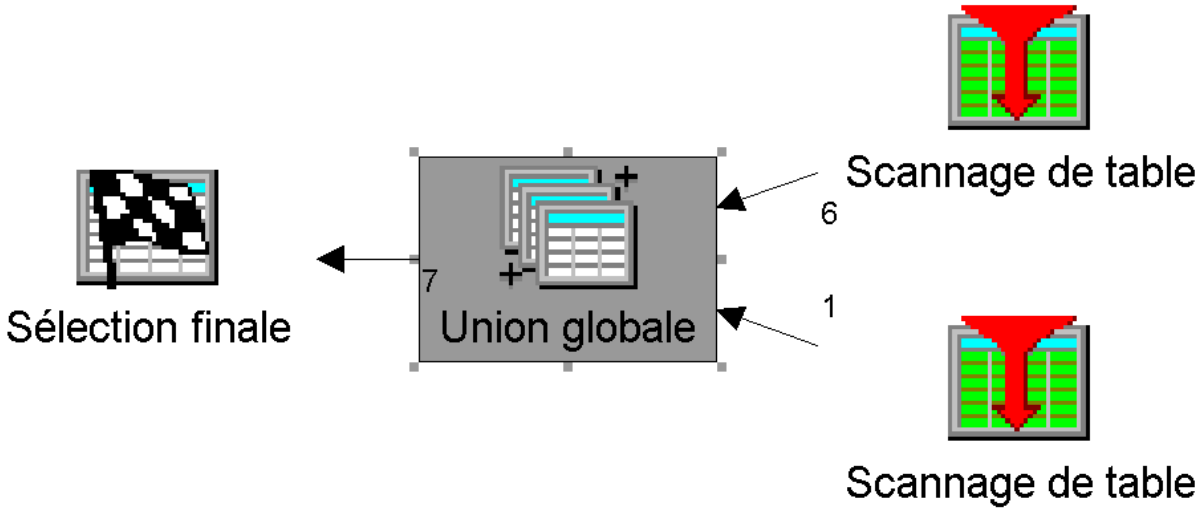
Autres effets

- Accès natifs vs SQL
  - Prend en compte la valeur SYSTIME_PERIOD_ADJ de QAQQINI
    - CPF503B (code raison 3) émis lorsque SYSTIME_PERIOD_ADJ a la valeur *DEFAULT ou *ERROR si une mise à jour provoque l'insertion dans la table historique d'un enregistrement avec une date de fin inférieure à la date de début
      - Cas d'application ?
  - Ignore CURRENT TEMPORAL SYSTEM_TIME
    - C'est-à-dire que les accès natifs travaillent exclusivement sur la table d'origine
    - Aucun accès implicite à la table historique n'est réalisé en lecture

Performance

- Table historique prise en charge par l'optimiseur SQE

```
SELECT *
FROM CLIENTS FOR SYSTEM_TIME AS OF
  '2017-02-01 14:45:02.934758000244'
```



Performance

- Mécanisme intégré à la BD
  - Plus performant que gestion manuelle avec un trigger
  - Temps observés : +25% par rapport à une table sans versionning pour les opérations de mise à jour et suppressions
- Indexation
  - Index sur « Row begin » et « Row end »
- Préférences mémoire et média
  - KEEP IN MEMORY YES
  - UNIT SSD

Restrictions

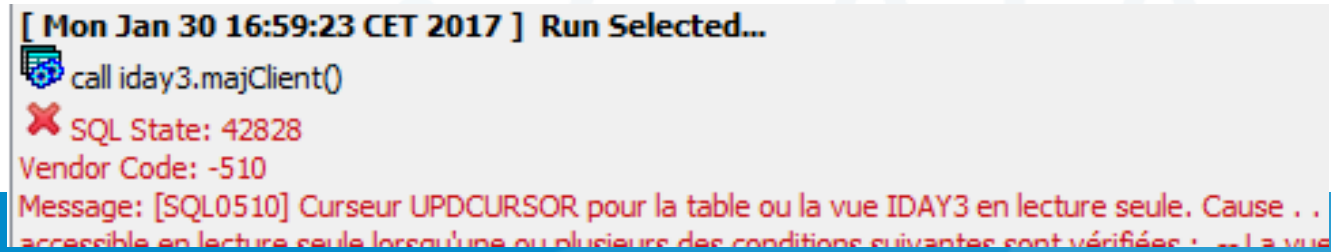
Opérations

- Les opérations suivantes ne sont pas supportées avec les tables temporelles
  - `CREATE OR REPLACE TABLE` *`base`*
    - `CREATE TABLE` uniquement lorsque la table *`base`* est liée à sa table *`historique`*
  - `DROP TABLE` *`historique`* | `ALTER TABLE` *`historique`*
    - Non autorisé
  - Les deux tables doivent être
    - Journalisées
    - Dans la même bibliothèque
- La table historique ne peut pas faire partie d'une contrainte référentielle

Curseur

- Il est impossible de supprimer ou mettre à jour un enregistrement avec un curseur
  - Avec ou sans `WHERE CURRENT OF`

```
CREATE OR REPLACE PROCEDURE majClient
LANGUAGE SQL
BEGIN
  DECLARE cur_nom CHAR(50);
  DECLARE updCursor CURSOR FOR
    SELECT nomcli
      FROM clients FOR SYSTEM_TIME AS OF '2017-01-30-17.00.00.000000';
  OPEN updCursor;
  FETCH updCursor INTO cur_nom ;
  UPDATE clients SET nomcli = upper( cur_nom )
    WHERE CURRENT OF updCursor;
  CLOSE updCursor;
END;
```



Comportement

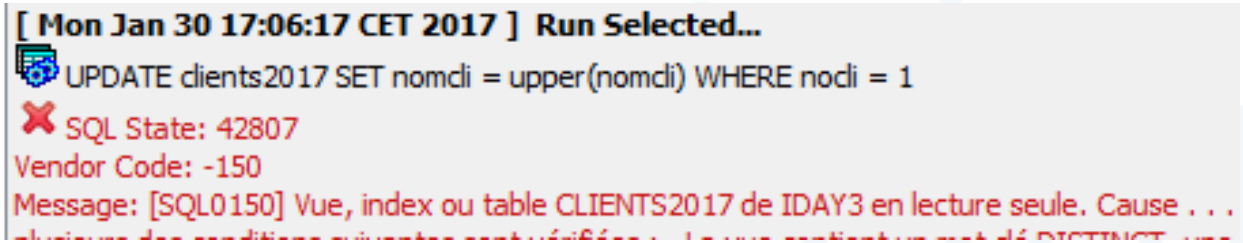
- Opérations supportées
  - `DROP TABLE base`
    - Supprime les deux tables : `base` et `historique`
  - `ALTER TABLE base ADD COLUMN`
    - Ajoute également la colonne à la table `historique`
    - Ne supporte pas les colonne GENERATED
  - `ALTER TABLE base DROP COLUMN`
    - Ou réduction de la taille de la colonne : impossible
  - `DELETE FROM historique`
    - Permet l'élagage des enregistrements

Vue

- Les insertions, suppressions et mises à jour ne sont pas autorisées dans une vue définie avec une période

```
CREATE OR REPLACE VIEW clients2017 AS
  SELECT nocli, nomcli
    FROM clients FOR SYSTEM_TIME AS OF
       TIMESTAMP('2017-01-30-18.00.00.000000');

UPDATE clients2017 SET nomcli = upper(nomcli) WHERE nocli = 1 ;
```



Catalogue

Vues

- SYSHISTORYTABLES
  - Une ligne pour chaque table 'historique`

```
select * from qsys2.SYSHISTORYTABLES
where history_table_schema = 'IDAY3' ;
```

HISTORY_TABLE_SCHEMA	HISTORY_TABLE_NAME	VERSIONING_STATUS	PERIOD_NAME	TABLE_SCHEMA	TABLE_NAME
IDAY3	CLIENTS_HISTO	E	SYSTEM_TIME	IDAY3	CLIENTS

SYSTEM_HISTORY_SCHEMA	SYSTEM_HISTORY_TABLE_NAME	SYSTEM_TABLE_SCHEMA	SYSTEM_TABLE_NAME
IDAY3	CLIEN00001	IDAY3	CLIENTS

- SYSPERIODS
  - Une ligne par période définie pour une table temporelle

PERIOD_NAME	TABLE_SCHEMA	TABLE_NAME	BEGIN_COLUMN_NAME	END_COLUMN_NAME	PERIOD_TYPE	HISTORY_TABLE_SCHEMA	HISTORY_TABLE_NAME
SYSTEM_TIME	IDAY3	CLIENTS	DEBUT	FIN	S	IDAY3	CLIENTS_HISTO

ON_DELETE_ADD_EXTRA_ROW	VERSIONING_STATUS	SYSTEM_TABLE_SCHEMA	SYSTEM_TABLE_NAME	SYSTEM_HISTORY_TABLE_SCHEMA	SYSTEM_HISTORY_TABLE_NAME
YES	E	IDAY3	CLIENTS	IDAY3	CLIEN00001

SYSTEM_HISTORY_TABLE_NAME	SYSTEM_BEGIN_COLUMN_NAME	SYSTEM_END_COLUMN_NAME
CLIEN00001	DEBUT	FIN

Vues

- SYSTABLES
  - Nouvelle colonne TEMPORAL_TYPE
    - 'N' : pas une table temporelle ni une table historique
    - 'H' : table historiques
    - 'S' : table temporelle

```
select table_name, table_type, file_type, temporal_type
from qsys2.systables where table_schema = 'IDAY3' ;
```

TABLE_NAME	TABLE_TYPE	FILE_TYPE	TEMPORAL_TYPE
CLIENTS	T	D	S
CLIENTS_HISTO	T	D	H
QRPGLESRC	P	S	N

Vues

- SYSCOLUMNS
  - La colonne HAS_DEFAULT indique les colonnes générées

```
select column_name, table_name, column_default from
qsys2.syscolumns where table_schema = 'IDAY3' and
table_name like 'CLIENTS%';
```

COLUMN_NAME	TABLE_NAME	COLUMN_DEFAULT
NOCLI	CLIENTS	-
NOMCLI	CLIENTS	-
DEPCLI	CLIENTS	-
DATCRT	CLIENTS	CURRENT_DATE
USRCRT	CLIENTS	SESSION_USER
ACTION	CLIENTS	DATA CHANGE OPERATION
DEBUT	CLIENTS	-

Intégration

Compilation SQL

- Nouveau paramètre sur les commandes RUNSQLSTM et RUNSQL
  - SYSTIME
    - Indique si les instructions SQL statiques et dynamiques sont sensibles au registre CURRENT TEMPORAL SYSTEM_TIME
    - Dépendant de l'heure système
      - *YES
      - *NO
- CRTSQL*
  - OPTION(*SYSTIME or *NOSYSTIME)
- SQL
  - SET OPTION SYSTIME = *YES or *NO

Commandes fichiers

- DSPFD et DSPFFD
  - N'indiquent pas la temporalité
- DSPDBR
  - N'indique pas le lien entre les tables
- CLRPFM
  - Provoque CPF3157
    - Des déclencheurs ou la table temporelle de période système bloquent l'opération demandée.
- CPYF
  - Comportement classique
- CRTDUPOBJ
  - Créé une table temporelle, mais sans table historique
  - Seule la lecture est possible
  - Les autres opérations échouent jusqu'à
    - Créer et associer la table historique
    - Supprimer la temporalité

SAV / RST

- SAV
  - La table historique doit être explicitement sauvegardée
- RST
  - Lorsque la table base est restaurée sans la table historique, elle n'est pas utilisable
    - État « defined »
  - Les seules opérations possibles sont alors
    - ALTER TABLE ADD VERSIONING
    - ALTER TABLE DROP VERSIONING
    - DROP TABLE
  - Dès que la table historique est restaurée, la table de base devient utilisable
    - État « versioned »

Bonnes pratiques

- Nommage
  - Respecter l'ordre alphabétique pour les sauvegarder et restaurer ensemble
    - CLIENTS
    - CLIENTS_HISTO
- Performances
  - Index (radix) sur la table historique
- Elagages
  - Pour limiter le nombre d'enregistrements dans la table historique
- Droits
  - Aucun droit nécessaire sur la table historique : *EXCLUDE
- Colonnes IMPLICITLY HIDDEN
  - Pour les colonnes de suivi