

# Modernisation IBM i – Nouveautés 2014-2015

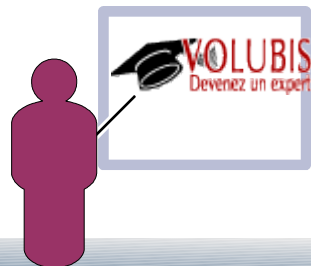
19 et 20 mai 2015 – IBM Client Center, Bois-Colombes

Volubis.fr

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

Dans nos locaux, vos locaux ou par Internet

*Christian Massé - cmasse@volubis.fr*



# Modernisation IBM i – Nouveautés 2014-2015

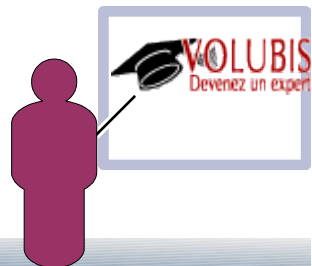
19 et 20 mai 2015 – IBM Client Center, Bois-Colombes

Volubis.fr

*Base de connaissance depuis 1995 (plus de 500 cours)*

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

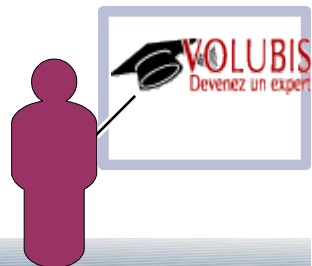
*Accédez à tout cela **gratuitement** pendant trois semaines cet été !*



# Modernisation IBM i – Nouveautés 2014-2015

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

Session 14 : R C A C



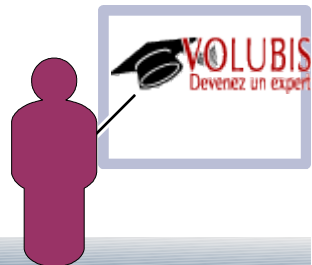
# Version 7.2

## Nouvelle clause VIOLATION sur les Check constraint

- **ON INSERT VIOLATION SET column-name = DEFAULT**  
L'erreur n'est pas signalée, la valeur par défaut est insérée
- **ON UPDATE VIOLATION PRESERVE column-name**  
L'erreur n'est pas signalée, la valeur précédente est conservée

## •Exemple

```
create table bdvin1/verif (  
  cle int as identity,  
  libelle char(50),  
  verifOK char(1) default 'o' check (verifok in ('o' , 'n') )  
                                on insert violation set verifOK = DEFAULT  
                                ON UPDATE VIOLATION PRESERVE verifOK  
)  
La table VERIF a été créée dans BDVIN1.
```



# Version 7.2

## Nouvelle clause VIOLATION sur les Check constraint

### •Vu pas System i Navigator

✓ Définition de contrainte de vérification - Ibmitest(E00dac4p)

Nom de contrainte : Q\_BDVIN1\_VERIF\_VERIFOK\_00001

Condition de vérification :

VERIFOK IN ('o', 'n')

Violation des données

Colonne : VERIFOK

Actions

- Sur violation d'insertion, définir sur la valeur par défaut de la colonne
- Sur violation de mise à jour, conserver la valeur de la colonne

Texte :

OK Annulation Aide ?



# Version 7.2

## Nouvelle clause VIOLATION sur les Check constraint

- suite à deux INSERT, dont l'un ne renseignant pas la colonne Verifok, nous avons bien les valeurs attendues

```
select * from VERIF
```

CLE	LIBELLE	VERIFOK
1	test1	0
2	test1	0

## Mais suite à cet INSERT qui aurait du être refusé

```
INSERT INTO BDVIN1/VERIF (LIBELLE, VERIFOK) VALUES('test2', 'x')  
1 lignes insérées dans VERIF de BDVIN1.
```

CLE	LIBELLE	VERIFOK
1	test1	0
2	test1	0
4	test2	0



# Version 7.2

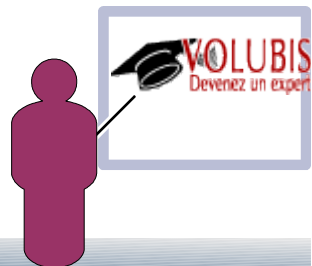
## Nouvelle clause VIOLATION sur les Check constraint

- Enfin, suite à cet UPDATE, lui aussi invalide

```
update bdvin1/verif set verifok = ' '  
where cle = 1  
1 lignes mises à jour dans VERIF de BDVIN1.
```

CLE	LIBELLE	VERIFOK
1	test1	0
2	test1	0
4	test2	0

- NB : Aucun message dans la LOG pour signaler les "remplacement" de valeur





# Version 7.2

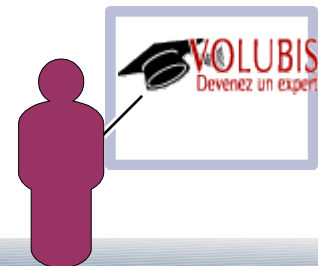
**L' option 47 de 5770SS1 (non facturable) apporte RCAC**

•Row and Column Access Control

```
Logiciels sous licence installés
```

Logiciel sous licence	Option produit	Description
5770SS1	47	IBM Advanced Data Security for i

Il s'agit de pouvoir indiquer des « droits » à la colonne ou à la ligne qui s'appliquent y compris aux personnes ayant les droits d'administrateur

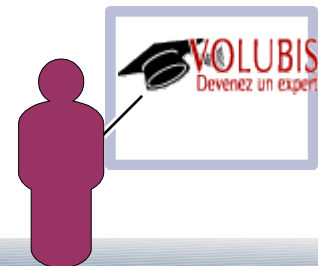
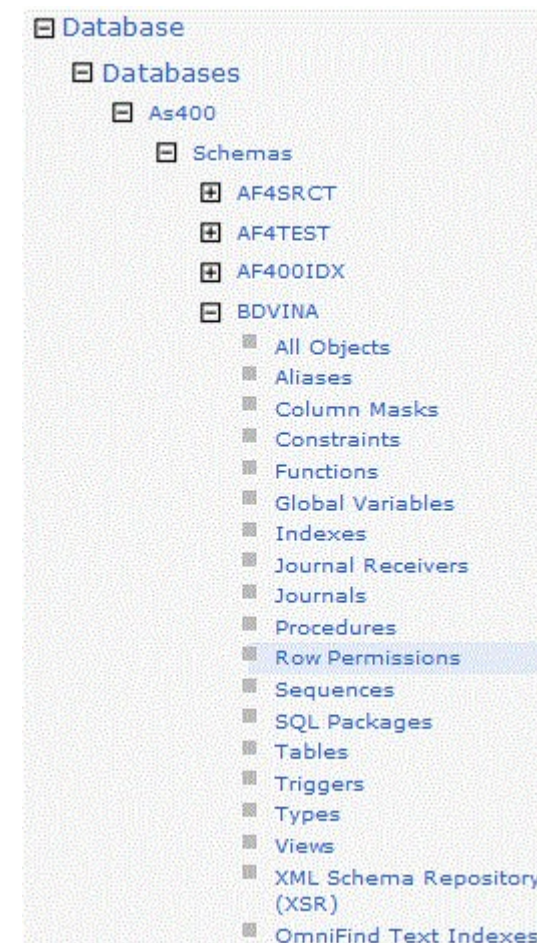
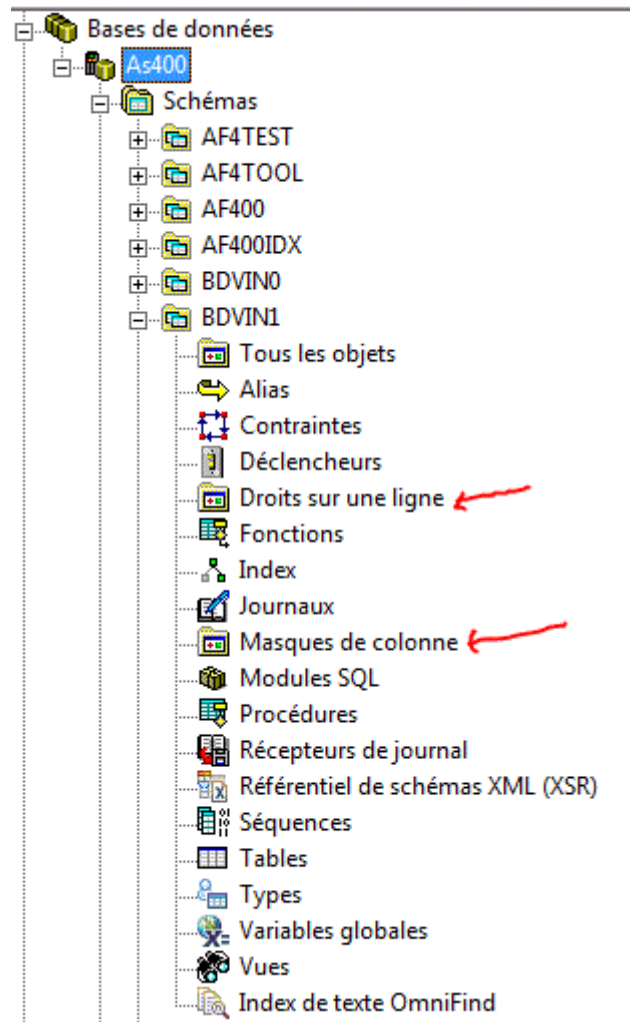




# RCAC

Ces deux nouvelles options sont accessibles via  
System i Navigator ou

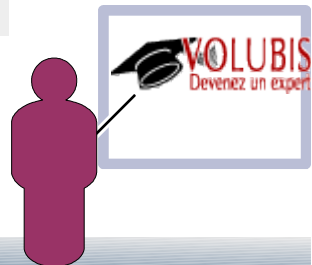
Navigator for i (version Web).



# RCAC

**CREATE MASK** indique si une colonne est retournée tel que ou totalement/partiellement masquée ('xxx-xxx-xxx-1234' pour un n° de CB)

```
CREATE [or REPLACE] MASK tel_MASK ON bdvin1/producteurs
FOR COLUMN pr_tel RETURN
CASE
  WHEN SESSION_USER = 'QSECOFR'
    THEN PR_TEL
  WHEN SESSION_USER = 'CM'
    THEN left(pr_tel , 3) concat 'XXXXXXXXXXXXXX'
  ELSE NULL
END
ENABLE
```





# RCAC

## Sous System i Navigator

Nouveau masque de colonne - As400(As400)

Nom : tel\_mask

Schéma de table : BDVIN1

Nom de table : PRODUCTEURS

Nom de corrélation pour une table : Non spécifié

Nom de colonne	Nom de ...	Type de donn...	Long...	Val...	Valeur p...	Texte	CCSID	Procédu...	En-tête 1	En-tête 2
PR_TEL	PR_TEL	CHARACTER	20	Oui	Valeur in...	tel produ...	297		PR_TEL	
PR_FAX	PR_FAX	CHARACTER	20	Oui	Valeur in...	fax produ...	297		PR_FAX	
PR_VENTE	PR_VEN...	CHARACTER	3	Oui	Valeur in...	vente oui/...	297		PR_VENTE	
PR_VISITE	PR_VISITE	CHARACTER	3	Oui	Valeur in...	visite oui/...	297		PR_VISITE	

Pour la colonne :

Retour

Expression CASE :

```
CASE
WHEN SESSION_USER = 'QSECOFR' THEN PR_TEL
  WHEN SESSION_USER = 'CM' THEN left(pr_tel , 3) concat 'XXXXXXXXXXXXXXXX'
ELSE NULL
END
```

Valeur de prévisualisation

Vérification de syntaxe

Activé

Texte :

Affichage de SQL

OK Annulation Aide ?

# RCAC

Puis

```
ALTER TABLE bdvin1/producteurs  
  ACTIVATE COLUMN ACCESS CONTROL
```

ou

BDVIN1.PRODUCTEURS - Ibmittest(E00dac4p)

Table	Colonnes	Contraintes de clé	Contraintes de clé associée
Nom :	PRODUCTEURS		
Schéma :	BDVIN1		
Nom de système :	PRODUCTEUR		

- Le support de stockage préférentiel est une unité SSD
- Données non rémanentes
- Contrôle d'accès de ligne
- Contrôle d'accès de colonne



# RCAC

Attention vous devez avoir les droits QIBM\_DB\_SECADM (même QSECOFR !)

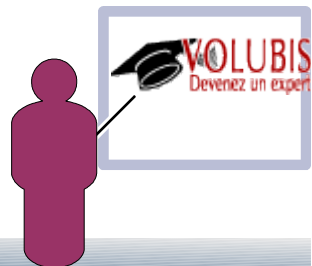
•sinon vous recevrez SQL0552 **Non autorisé à utiliser CREATE MASK.**

pour donner ce droit : WRKFCNUSG

```
Gestion de l'utilisation de fonctions

Indiquez vos options, puis appuyez sur ENTREE.
 2=Modifier l'utilisation   5=Utilisation

Opt  ID fonction                               Nom de la fonction
--  -
_    QIBM_DIRSRV_ADMIN                         IBM Tivoli Directory Server Administrator
_    QIBM_ACCESS_ALLOBJ_JOBLOG                 Accès à l'historique de travail du travail
_    QIBM_ALLOBJ_TRACE_ANY_USER               Trace any user
_    QIBM_WATCH_ANY_JOB                       Watch any job
_    QIBM_DB_DDMDRDA                           DDM & DRDA Application Server Access
_    QIBM_DB_SECADM                           Administrateur sécurité base de données
_    QIBM_DB_SQLADM                           Administrateur de base de données
_    QIBM_DB_SYSMON                           Informations de base de données
```





# RCAC

## Option 2 pour modifier

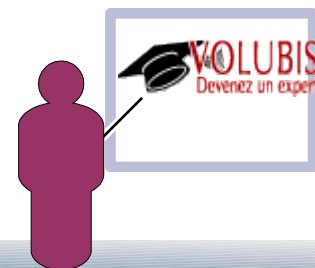
```
Modifier utilisation fonction (CHGFCNUSG)

Indiquez vos choix, puis appuyez sur ENTREE.

ID fonction      . . . . . > QIBM_DB_SECADM
Utilisateur      . . . . . _____ Nom
Utilisation      . . . . . *ALLOWED, *DENIED, *NONE
Droit par défaut . . . . . *DENIED      *SAME, *ALLOWED, *DENIED
Droit spécial *ALLOBJ . . . . . *NOTUSED  *SAME, *USED, *NOTUSED
```

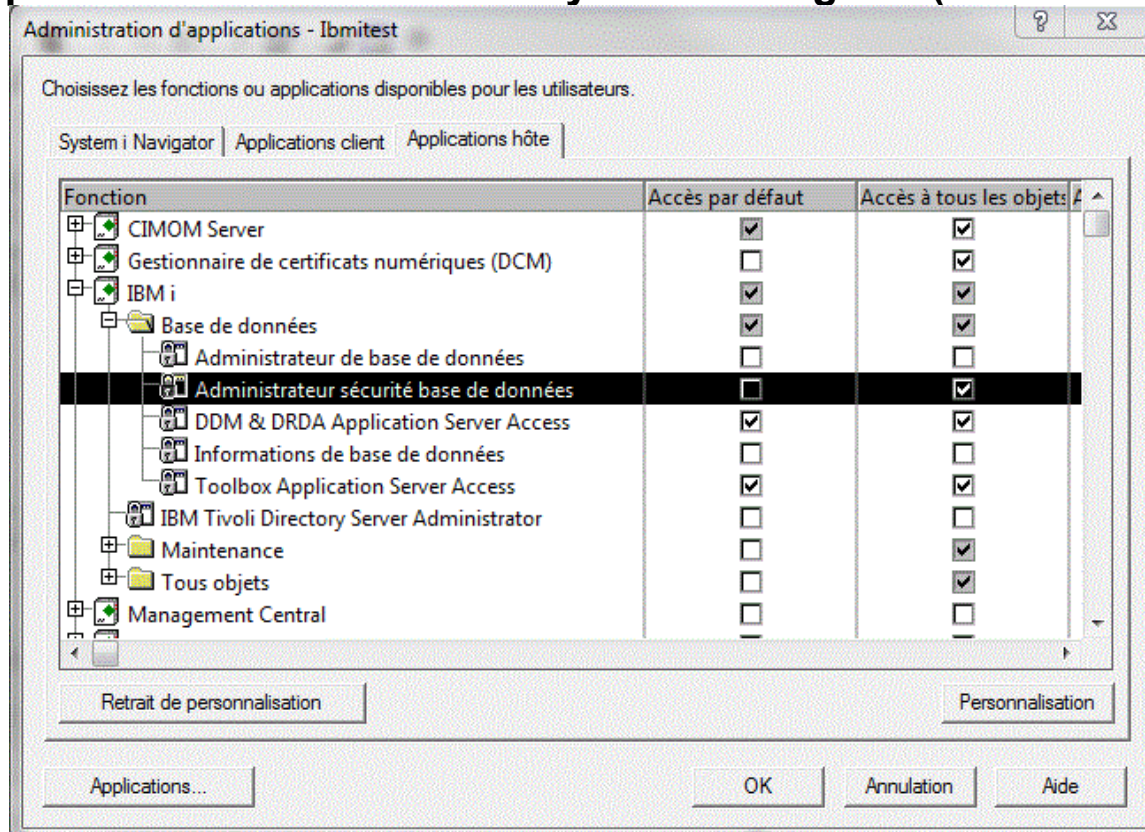
Indiquez :

- **\*USED** face à *Droits spécial* **\*ALLOBJ** pour que QSECOFR puisse manipuler ces notions
- **\*ALLOWED** face à *Droits par défaut* pour que TOUT LE MONDE puisse manipuler ces notions (déconseillé !)
- sinon indiquez un profil à ajouter (Utilisateur) et **\*ALLOWED** (Utilisation) pour autoriser un profil ou un groupe



# RCAC

**Vous pouvez aussi le faire avec System i Navigator (administration d'applications)**



**la restriction étant posée, vous pouvez la modifier, retrouver la source et la supprimer**

Nom	Nom de table	Nom de colonne	Activé	Créateur	Date de création
TEL_MASK	PR_TEL	PR_TEL	Oui	QSECOFR	07/08/14 16:52:54

Menu contextuel pour TEL\_MASK:

- Définition
- Génération d'instructions SQL...
- Commentaires...
- Suppression...





# RCAC

Les restrictions RCAC s'appliquent dans tous les contextes :

## DSPPFM sous QSECOFR

```

                                Membre de fichier physique
Fichier . . . . . :  PRODUCTEUR          Bibliothèque . . . :  BDVIN1
Membre . . . . . :  PRODU00001         Enregistrement . . :  1
Contrôle . . . . . :  _____         Colonne . . . . . :  235
Recherche . . . . .
+ . . . 4 . . . + . . . 5 . . . + . . . 6 . . . + . . . 7 . . . + . . . 8 . . . + . . . 9 . . . + . . . 0 . . .
  Jean-Pierre Hugon          05 57 88 30 01
  Philippe Dourthe          05 56 58 01 23

```

## DSPPFM sous CM

```

                                Membre de fichier physique
Fichier . . . . . :  PRODUCTEUR          Bibliothèque . . . :  BDVIN1
Membre . . . . . :  PRODU00001         Enregistrement . . :  1
Contrôle . . . . . :  _____         Colonne . . . . . :  235
Recherche . . . . .
+ . . . 4 . . . + . . . 5 . . . + . . . 6 . . . + . . . 7 . . . + . . . 8 . . . + . . . 9 . . . + . . . 0 . . .
  Jean-Pierre Hugon          05 XXXXXXXXXXXXXX
  Philippe Dourthe          05 XXXXXXXXXXXXXX

```



# RCAC

Bien sur la valeur retournée doit être compatible

•1<sup>er</sup> test refusé le prix est numérique

```
Le type de données du littéral *N n'est pas compatible avec la colon  
create or replace MASK prixMASK ON bdvin1/ma_cave  
FOR COLUMN cav_prix RETURN  
CASE  
WHEN SESSION_USER = 'QSECOFR'  
THEN cav_prix  
WHEN SESSION_USER = 'CM'  
THEN 'xxxxxxxxxx'  
ELSE NULL  
END  
ENABLE
```

•2eme, OK, on retourne 0

```
> create or replace MASK prixMASK ON bdvin1/ma_cave  
FOR COLUMN cav_prix RETURN  
CASE  
WHEN SESSION_USER = 'QSECOFR'  
THEN cav_prix  
WHEN SESSION_USER = 'CM'  
THEN 0  
ELSE NULL  
END  
ENABLE  
Instruction CREATE MASK terminée pour PRIXMASK de BDVIN1.
```



# RCAC

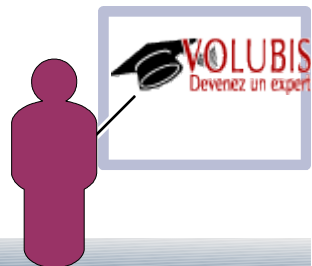
**Un MASK n'empêche pas les insertions**

**(par contre vous ne retrouvez pas forcément la donnée telle que vous l'avez insérée, mais masquée)**

**•Le problème se pose éventuellement lors des mises à jour :**

**Exemple , le pgm suivant lit et modifie le producteur 1 en RPG  
( fichier qui possède un MASK sur PR\_TEL)**

```
.....  
■ .....  
ctl-opt alwnull(*usrctl);  
dcl-F producteur disk usage(*update);  
chain 1 producteur;  
pr_fax = 'modifié';  
update prodf ;  
*inLR = *on;
```

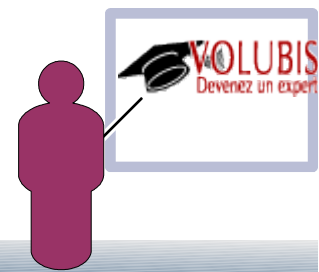


# RCAC

lors de l'Update RPG, il met à jour la ligne suivant les données qu'il a lui même reçu.

• Suite à un CALL par CM (c'est QSECOFR qui regarde le contenu de la table) :

```
Affichage des données
                                Largeur des données . . : 796
Première ligne à afficher . . . :
Première colonne à afficher . . : 265
+...27...+...28...+...29...+...30...+...31...+...32...+...33...+...34...
PONSABLE                                PR_TEL                                PR_FAX
Pierre Hugon                            05 XXXXXXXXXXXXX                          modifié
pe Dourthe                              05 56 58 01 23                            05 56 58 00
```



# RCAC

Y compris en utilisant la fonction %fields()

```
ctl-opt alwnull(*usrctl);  
dcl-F producteur disk usage(*update);  
chain 2 producteur;  
pr_fax = 'modifié';  
update prodf %fields(pr_fax);  
*inLR = *on;
```

Affichage des données

Largeur des données . . . : 796		
Première ligne à afficher . . .	Première colonne à afficher . . . 265	
+...27...+...28...+...29...+...30...+...31...+...32...+...33...+...34...		
PONSABLE	PR_TEL	PR_FAX
ierre Hugon	05 XXXXXXXXXXXXX	modifié
pe Dourthe	05 XXXXXXXXXXXXX	modifié
t Strass	05 56 58 27 63	05 56 58 22



# RCAC

Seul un UPDATE SQL (toujours réalisé par CM) , ne met à jour que certains champs  
(donc ne touche pas aux autres)

## Entrée d'instructions SQL

```
Saisissez l'instruction SQL, puis appuyez sur ENTREE.  
> UPDATE BDVIN1/PRODUCTEURS SET PR_FAX = 'Modifié' WHERE pr_code = 3  
1 lignes mises à jour dans PRODUCTEUR de BDVIN1.  
===> _____
```

## Affichage des données

```
Largeur des données . . . : 796  
Première ligne à afficher . . . :  
Première colonne à afficher . 265  
+...27...+...28...+...29...+...30...+...31...+...32...+...33...+...34.  
PONSABLE PR_TEL PR_FAX  
ierre Hugon 05 XXXXXXXXXXXXX modifié  
pe Dourthe 05 XXXXXXXXXXXXX modifié  
t Strass 05 56 58 27 63 Modifié  
t Vonderbouden 05 56 58 80 04 05 56 58
```



# RCAC

La documentation conseille dans le cas d'entrées/sorties « natives » :

1/ de faire un trigger, qui rétablisse l'ancienne valeur

```
IF substr(new.PR_TEL, 4, 13) = 'XXXXXXXXXXXXXX' THEN  
  new.PR_TEL = old.PR_TEL;
```

... /...

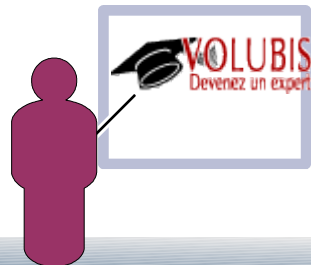
2/ de faire une contrainte qui refuse la valeur produite par le MASK

```
CHECK substr(PR_TEL, 4, 13) <> 'XXXXXXXXXXXXXX'
```

Ou **mieux**, faites une contrainte qui utilise la nouvelle clause  
ON UPDATE VIOLATION

```
CHECK substr(PR_TEL, 4, 13) <> 'XXXXXXXXXXXXXX'  
  ON UPDATE VIOLATION PRESERVE PR_TEL
```

L'ancienne valeur sera alors remplacée automatiquement,  
sans message d'erreur





# RCAC

Enfin, les données sont masquées, juste avant l'affichage, c'est à dire après jointure et GROUP BY

par exemple, le SELECT suivant, qui donne le nombre de clients par indicatif téléphonique

```
SELECT LEFT(PR_TEL, 5) , count(*) as  
nombre  
FROM producteurs  
GROUP BY LEFT(PR_TEL, 5)
```

affiche

si la colonne n'est pas masquée		si la colonne est masquée	
05 55	3	05 xx	3
05 56	7	05 xx	7
05 58	4	05 xx	4



# RCAC

**CREATE PERMISSION** indique la(les) règles(s) qui font qu'une ligne peut être vue

• Toute ligne ne correspondant pas à la règle n'est pas retournée :

**Exemple CM ne doit pas voir l'appellation 13**

```
CREATE [or REPLACE] PERMISSION VINS_ROW_ACCESS ON bdvin1/vins
FOR ROWS
WHERE
  SESSION_USER <> 'CM'
  OR (SESSION_USER = 'CM'
      and (appel_code <> 13 or appel_code IS NULL)
  )
ENFORCED FOR ALL ACCESS
ENABLE
```

*rappelez vous, on indique ce qui peut être vu (une affirmation, donc)*

Puis

```
ALTER TABLE bdvin1/vins
  ACTIVATE ROW ACCESS CONTROL
```



# RCAC

**CREATE PERMISSION** indique la(les) règles(s) qui font qu'une ligne peut être vue

## •Sous System i Navigator

Nouveaux droits sur une ligne - As400.volubis.intra(As400)

Nom : VINS\_ROW\_ACCESS

Schéma de table : BDVIN1

Nom de table : VINS

Nom de corrélation pour une table : Non spécifié

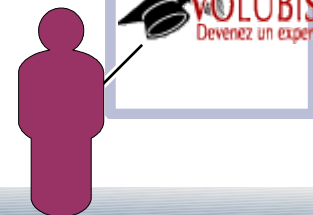
Pour les lignes où

Condition de recherche : 

```
SESSION_USER <> 'CM'  
OR (SESSION_USER = 'CM'  
    and (appel_code <> 13 or appel_code IS NULL)  
)
```

Activé

Texte :



# RCAC

On aurait aussi pu faire deux permissions

```
CREATE PERMISSION VINS_ROW_ACCESS1 ON bdvin1/vins
FOR ROWS
WHERE
  SESSION_USER <> 'CM'

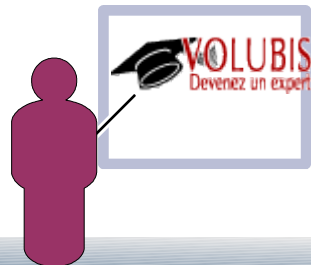
ENFORCED FOR ALL ACCESS
ENABLE ;

-----

CREATE PERMISSION VINS_ROW_ACCESS2 ON bdvin1/vins
FOR ROWS
WHERE
  SESSION_USER = 'CM'
  and (appel_code <> 13 or appel_code IS NULL)

ENFORCED FOR ALL ACCESS
ENABLE ;
```

un peu moins performant en temps de réponse...



# RCAC

**Le système ajoute alors une permission implicite QIBM\_DEFAULT\_nomdetable\_schema où la permission est si 0=1, donc toujours fausse.**

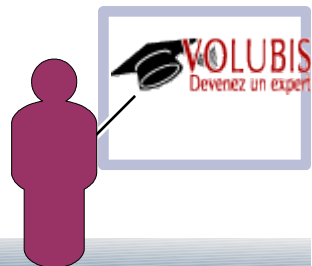
- seules les permissions explicites autorisent des lignes à être vues (en bref tout ce qui n'est pas autorisé est interdit)**
- Donc, Attention, si vous enlever la permission sans désactiver ROW ACCESS CONTROL plus aucune ligne ne peut être extraite (le fichier apparaît toujours comme vide !)**

**Avec notre PERMISSION "VINS\_ROW\_ACCESS" :**

**SELECT COUNT(\*) from VINS , sous QSECOFR affiche 25.221**

**SELECT Count(\*) from VINS WHERE appel\_code = 13 indique un nombre de 811 vins pour cette appellation**

**SELECT COUNT(\*) from VINS , sous CM affiche 24.410**



# RCAC

Une **PERMISSION**, peut empêcher une insertion ou une mise à jour, qui ne respecte pas la règle

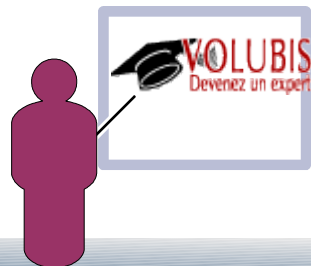
•Exemple avec **APPEL\_CODE** à 13 (vous recevez SQ20471)

```
> INSERT INTO BDVIN1/VINS VALUES(999999, 1, 'test', 'Cabernet', null,
null, null, null, null, null, 13, 1, null)
Instruction INSERT ou UPDATE non conforme aux droits de ligne.
```

•Administration :

- pour modifier : **ALTER MASK | PERMISSION**
- pour retirer : **DROP MASK | PERMISSION**

tant que vous n'avez pas activé les droits par **ALTER TABLE**,  
les **MASK** et les **PERMISSIONS** sont inopérants





# RCAC

nouvel ordre ALTER TRIGGER en version 7.2

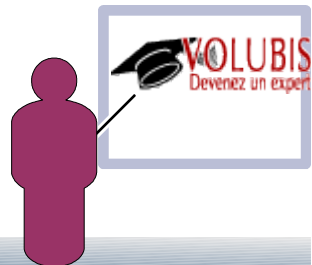
pour modifier certains paramètres d'un TRIGGER :

**ENABLE**, le trigger est actif (dft)  
**DISABLE**, le trigger n'est plus actif

**SECURED** ce trigger est sécurisé (compatible) avec les droits RCAC  
**NOT SECURED** ce trigger n'est pas compatible avec les droits RCAC (le défaut)

Il est impossible de modifier cet attribut quand des droits RCAC sont actifs

- il est impossible de créer un trigger NOT SECURED quand des droits RCAC sont actifs





# RCAC

```
> create trigger bdvin1.prdrtrg
  AFTER INSERT ON BDVIN1.PRODUCTEUR
  BEGIN
    call QCMDEXC('sndmsg msg(''producteur ajouté'') tousr(cm)');
  END
PRDTRG de BDVIN1 ne peut pas être utilisé pour le contrôle d'accès d
```

## Détail du message SQ20470

```
ID message . . . . . : SQ20470

Message . . . . . : PRDTRG de BDVIN1 ne peut pas être utilisé pour le contrôle
d'accès de ligne ou de colonne.

Cause . . . . . : PRDTRG de BDVIN1 type TRIGGER ne peut pas être créé ou
modifié pour l'une des raisons suivantes car PRODUCTEUR de BDVIN1 type FILE
dépend de celui-ci pour le contrôle d'accès de ligne ou de colonne.
  -- Une fonction définie par l'utilisateur doit être sécurisée si elle est
référéncée dans un DROIT ou un MASQUE ou dans une vue elle-même référéncée
par un DROIT ou un MASQUE.
  -- Un déclencheur doit être sécurisé s'il est défini sur une table avec un
DROIT ou un MASQUE actif.
  -- Un déclencheur doit être sécurisé s'il est défini sur une vue basé sur
une table avec un DROIT ou un MASQUE actif.
```

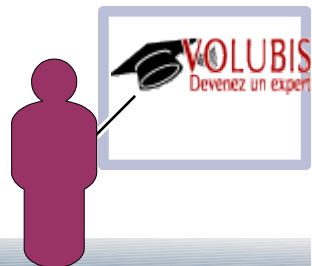
# RCAC

Par contre, cet ordre fonctionne

```
> create trigger bdvin1.prdrtrg
  AFTER INSERT ON BDVIN1.PRODUCTEUR SECURED
  BEGIN
    call QCMDEXC('sndmsg msg(''producteur ajouté'') tousr(cm)');
  END
Déclencheur PRDTRG créé dans BDVIN1.
```

Lors d'un insert, le trigger est lancé sans problème

```
Messages ne nécessitant pas de réponse
- producteur ajouté
  De . . . : QSEC0FR          23/07/14   16:02:07
```



# RCAC

Les mêmes paramètres sont proposés sur CREATE TRIGGER / CREATE FUNCTION, y compris sur les assistants de création :

The image displays two overlapping screenshots from SQL Developer. The top screenshot, titled 'Navigator for I', shows the 'New SQL Trigger' wizard. It includes a section titled 'Specify whether this trigger is considered secure for row and column access control:' with two radio buttons: 'Not secured' (selected) and 'Secured'. The bottom screenshot, titled 'Nouvelle fonction SQL - As400(As400)', shows the 'Options' tab of the function wizard. It features several dropdown menus and checkboxes. A red arrow points to the 'Considéré comme sécurisé pour le contrôle d'accès de colonne et de ligne' checkbox, which is currently unchecked. Another red arrow points to the 'Considéré comme sécurisé pour le contrôle d'accès de ligne et de colonne' checkbox in the top screenshot, which is also selected.

System i Navigator

Indiquez si ce déclencheur est considéré comme sécurisé pour le contrôle d'accès de ligne et de colonne :

- Non sécurisé
- Sécurisé

Nouvelle fonction SQL - As400(As400)

Fonction | Paramètres | Retours | Options | Corps de routine

Accès aux données ! Lit des données SQL

Résolution des accès simultanés : Par défaut

Exécution simultanée autorisée : Non spécifié

- CALLED ON NULL INPUT
- Même valeur renvoyée à partir d'appels successifs pour des paramètres identiques
- Exécute une action externe
- Sera exécuté dans une unité d'exécution distincte
- Considéré comme sécurisé pour le contrôle d'accès de colonne et de ligne

Specify whether this trigger is considered secure for row and column access control:

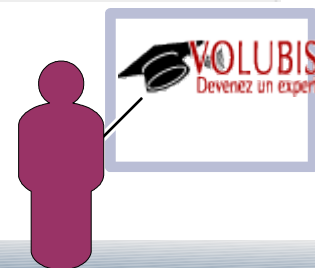
- Not secured
- Secured

Data access: Reads SQL data

Concurrent access resolution: Default

Can be run in parallel: Not specified

- Called on NULL input
- Same result returned from successive calls with identical
- Performs an external action
- Will run in a separate thread
- Considered secure for row and column access control



# RCAC

La mise en place ou le retraits des droits RCAC :

impose une allocation exclusive de la table

provoque un enregistrement AX dans le journal d'AUDIT.

et des enregistrements dans le journal de la table, de code D, type :

- M1 : CREATE MASK
- M2 : DROP MASK
- M3 : ALTER MASK
- P1 : CREATE PERMISSION
- P2 : DROP PERMISSION
- P3 : ALTER PERMISSION

Les droits RCAC sont visibles (mais non modifiables) par EDTOBJAUT/DSPOBJAUT

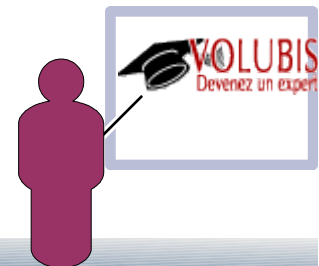
```
                Révision des droits sur un objet
Objet . . . . . :   PRODUCTEUR           Propriétaire . . . . . :   CM
  Bibliothèque . . . . . :       BDVIN1     Groupe principal . . . . . :   *NONE
Type d'objet . . . . . :   *FILE           Unité ASP . . . . . :   *SYSBAS
Contrôle d'accès de ligne ou de colonne . . . . . :   Actif
Indiquez les modifications sur les droits actuels, puis appuyez sur ENTREE.
```

# RCAC

Pour voir la liste des droits RCAC, regarder SYSCONTROLS et SYSCONTROLSDEP

• SysControls de QSYS2

RCAC_SCHEMA	<i>VARCHAR(128)</i>
RCAC_NAME	<i>VARCHAR(128)</i>
RCAC_OWNER	<i>VARCHAR(128)</i>
TABLE_SCHEMA	<i>VARCHAR(128)</i>
TABLE_NAME	<i>VARCHAR(128)</i>
TBCORRELATION	<i>VARCHAR(128)</i>
COLUMN_NAME	<i>VARCHAR(128)</i>
SYSTEM_COLUMN_NAME	<i>CHAR(10)</i>
SYSTEM_TABLE_NAME	<i>CHAR(10)</i>
SYSTEM_TABLE_SCHEMA	<i>CHAR(10)</i>
CONTROL_TYPE	<i>CHAR(1)</i> <b>M=Mask   R=Row permission</b>
ENFORCED	<i>CHAR(1)</i>
IMPLICIT	<i>CHAR(1)</i>
ENABLE	<i>CHAR(1)</i>
CREATE_TIME	<i>TIMESTAMP</i>
LAST_ALTERED	<i>TIMESTAMP</i>
IASP_NUMBER	<i>SMALLINT</i>
LABEL	<i>VARCHAR(50)</i>
LONG_COMMENT	<i>VARCHAR(2000)</i>
RULETEXT.	<i>DBCLOB -&gt; c'est ici qu'est la règle, en clair.</i>



# RCAC

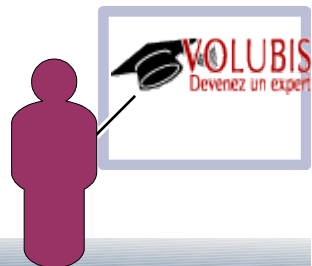
SYSCONTROLSDEP affiche les éléments dépendants, par exemple :

```
CREATE TABLE USEROK
(username char(10), appel_code int ) ;
-----
-- l'utilisateur CM ne doit voir que les appellations 13 et 144

INSERT INTO USEROK VALUES('CM', 13);
INSERT INTO USEROK VALUES('CM', 144);

CREATE PERMISSION PROD_ROW_ACCESS1 ON producteurs
FOR ROWS
WHERE
appel_code in (select appel_code from USEROK
               where username = SESSION_USER)
ENFORCED FOR ALL ACCESS
ENABLE ;
-----

ALTER TABLE producteurs
ACTIVATE ROW ACCESS CONTROL
ENFORCED FOR ALL ACCESS
ENABLE ;
```





# RCAC

SYSCONTROLSDEP contient alors :

```
SELECT SYSCONTROLSDEP.RCAC_NAME, SYSCONTROLSDEP.OBJECT_NAME, SYSCONTROLSDEP.OBJECT_TYPE,  
SYSCONTROLSDEP.COLUMN_NAME  
FROM QSYS2.SYSCONTROLSDEP AS SYSCONTROLSDEP;
```

RCAC_NAME	OBJECT_NAME	OBJECT_TYPE	COLUMN_NAME
PROD_ROW_ACCESS1	USEROK	TABLE	
PROD_ROW_ACCESS1	USEROK	COLUMN	APPEL_CODE
PROD_ROW_ACCESS1	USEROK	COLUMN	USERNAME



# RCAC

N'hésitez pas à créer vos propres fonctions pour retourner des informations complexes

Ex RTVUSRCLS

AF4TEST.RTVUSRCLS - As400(As400)

Fonction | Paramètres | Retours | Options

Nom : RTVUSRCLS

Schéma : AF4TEST

Nom spécifique : RTVUSRCLS

Langue : CL

Style de paramètre : SQL

Schéma de programme : AF4TEST

Programme ou programme de service : RTVUSRCLS

AF4TEST.RTVUSRCLS - As400(As400)

Fonction | Paramètres | Retours | Options

Paramètres :

Nombre	Nom	Type de données	Longueur
1	USER	CHARACTER	10

AF4TEST.RTVUSRCLS - As400(As400)

Fonction | Paramètres | Retours | Options

Valeur renvoyée à l'instruction d'appel

Type de données : CHARACTER

Longueur : 10





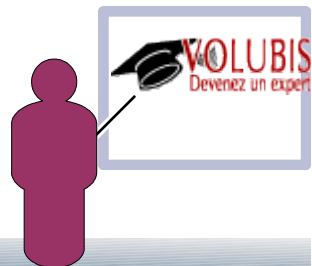
# RCAC

N'hésitez pas à créer vos propres fonctions pour retourner des informations complexes

Ex RTVUSRCLS

```
PGM parm(&profil &CLS &profilind &CLSind +  
        &SQLSTATE &functionQ &function &message)  
DCL     VAR(&PROFIL) TYPE(*CHAR) LEN(10)  
DCL     VAR(&CLS) TYPE(*CHAR) LEN(10)  
DCL     VAR(&PROFILIND) TYPE(*INT) LEN(2)  
DCL     VAR(&CLSIND) TYPE(*INT) LEN(2)  
DCL     VAR(&SQLSTATE) TYPE(*CHAR) LEN(5)  
DCL     VAR(&FUNCTIONQ) TYPE(*CHAR) LEN(519)  
DCL     VAR(&FUNCTION) TYPE(*CHAR) LEN(130)  
DCL     VAR(&MESSAGE) TYPE(*CHAR) LEN(1000)  
  
RTVUSRPRF  USRPRF(&PROFIL) USRCLS(&CLS)  
MONMSG CPF0000 EXEC(DO)  
  CHGVAR &CLSIND -1  
  ENDDO  
ENDPGM
```

Retourne le paramètre USRCLS ou NULL en cas de problème



# RCAC

N'hésitez pas à créer vos propres fonctions pour retourner des informations complexes

Ex RTVUSRAUT

AF4TEST.RTVUSRAUT - As400(As400)

Fonction | Paramètres | Retours | Options

Nom : RTVUSRAUT

Schéma : AF4TEST

Nom spécifique : RTVUSRAUT

Langue : CL

Style de paramètre : SQL

Schéma de programme : AF4TEST

Programme ou programme de service : RTVUSRAUT

AF4TEST.RTVUSRAUT - As400(As400)

Fonction | Paramètres | Retours | Options

Paramètres :

Nombre	Nom	Type de données	Longueur
1	USER	CHARACTER	10
2	SPCAUT	CHARACTER	10

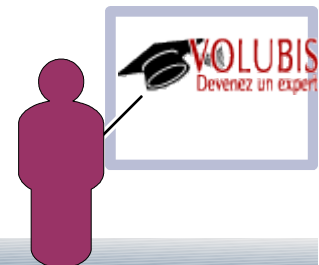
AF4TEST.RTVUSRAUT - As400(As400)

Fonction | Paramètres | Retours | Options

Valeur renvoyée à l'instruction d'appel

Type de données : DECIMAL

Précision : 1 Echelle : 0



# RCAC

N'hésitez pas à créer vos propres fonctions pour retourner des informations complexes

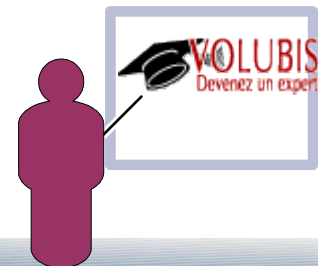
Ex RTVUSRAUT

```
PGM parm(&profil &aut &YES &profilind &autind &yesind +
        &SQLSTATE &functionQ &function &message)
DCL      VAR(&PROFIL) TYPE(*CHAR) LEN(10)
DCL      VAR(&AUT) TYPE(*CHAR) LEN(10)
DCL      VAR(&YES) TYPE(*DEC) LEN(1 0)
DCL      VAR(&PROFILIND) TYPE(*INT) LEN(2)
DCL      VAR(&AUTIND) TYPE(*INT) LEN(2)
DCL      VAR(&YESIND) TYPE(*INT) LEN(2)
DCL      VAR(&SQLSTATE) TYPE(*CHAR) LEN(5)
DCL      VAR(&FUNCTIONQ) TYPE(*CHAR) LEN(519)
DCL      VAR(&FUNCTION) TYPE(*CHAR) LEN(130)
DCL      VAR(&MESSAGE) TYPE(*CHAR) LEN(1000)

DCL      VAR(&SPCAUT) TYPE(*CHAR) LEN(100)

RTVUSRPRF  USRPRF(&PROFIL) SPCAUT(&SPCAUT)
MONMSG CPF0000 EXEC(DO)
  CHGVAR &YESIND -1
  RETURN
ENDDO
```

Retourne NULL en cas de problème



# RCAC

N'hésitez pas à créer vos propres fonctions pour retourner des informations complexes

Ex RTVUSRAUT

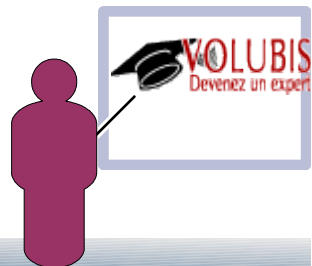
```
/* PROFIL EXISTE, LE DROIT SPÉCIAL EST-IL ATTRIBUÉ ? */  
IF          COND(%SCAN(&AUT &SPCAUT) > 0) THEN(DO)  
  CHGVAR    VAR(&YES) VALUE(1)  
ENDDO  
ELSE DO  
  CHGVAR    VAR(&YES) VALUE(0)  
enddo  
ENDPGM
```

Retourne 1 si l'utilisateur a le droit spécial demandé

Permettant CASE

```
When RTVUSRCLS(session_user) = '*SECOFR '  
Or RTVUSRAUT(session_user, '*ALLOBJ') = 1  
.../...
```

*Notez la fonction %scan disponible aujourd'hui en CL .... avec plein d'autres !*



# RCAC

Impact sur les requêtes SQL :

Autant les masques ne s'appliquent que lors de l'affichage (n'ont donc pas d'impact sur les jointures, par exemple)

Autant les permissions sont plus déterminantes :

- la jointure n'a lieu qu'avec les lignes autorisées.

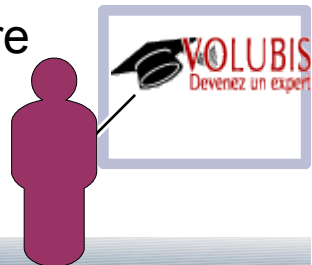
- INSERT into cible (SELECT \* from SOURCE)

vous subissez les MASK !

vous ne copiez que les lignes autorisés par les permissions

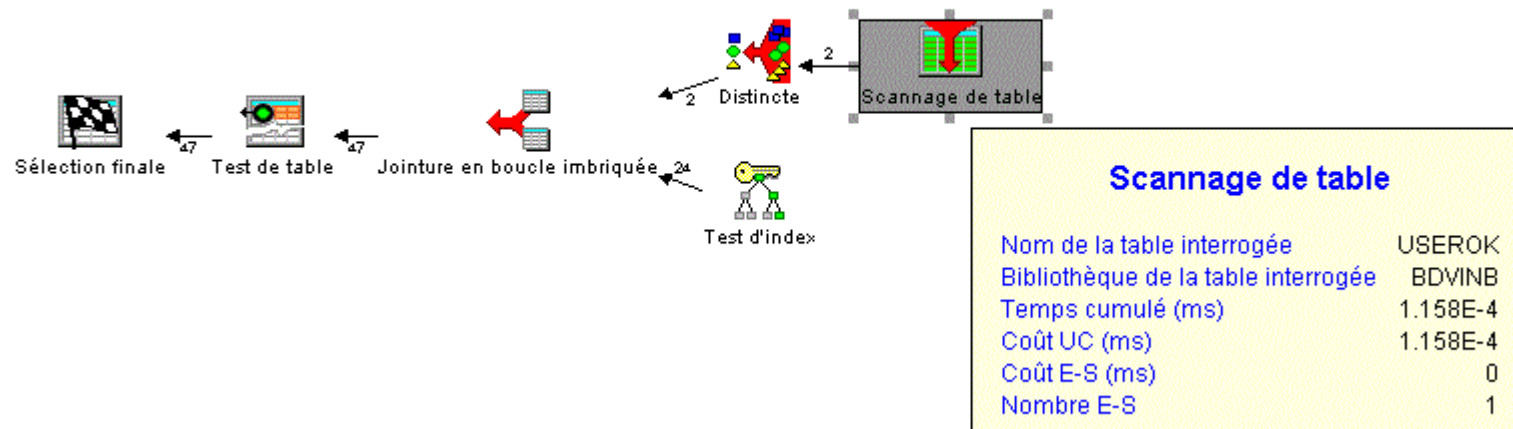
+ les lignes copiées doivent être valides dans la table cible  
(si elle même possède des droits RCAC)

SQE fait une jointure, le cas échéant : si le droit RCAC référence une autre table (cas d'utilisation de notre table USEROK)



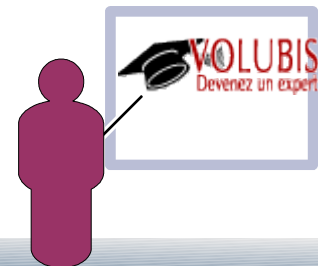
# RCAC

ici, sous VISUAL Explain, mais aussi dans les moniteurs SQL



Il signale qu'il y a des droits RCAC

AQP utilisé dans le plan d'accès	Non
Itération du plan d'accès AQP	1
Contrôle d'accès ←	Colonne





# RCAC

VISUAL Explain peut aussi proposer des index

Outil de conseil à la gestion des statistiques et des index - As400(As400)

Outil de conseil à la gestion des index | Outil de conseil à la gestion des statistiques

Il est recommandé de créer les index suivants :

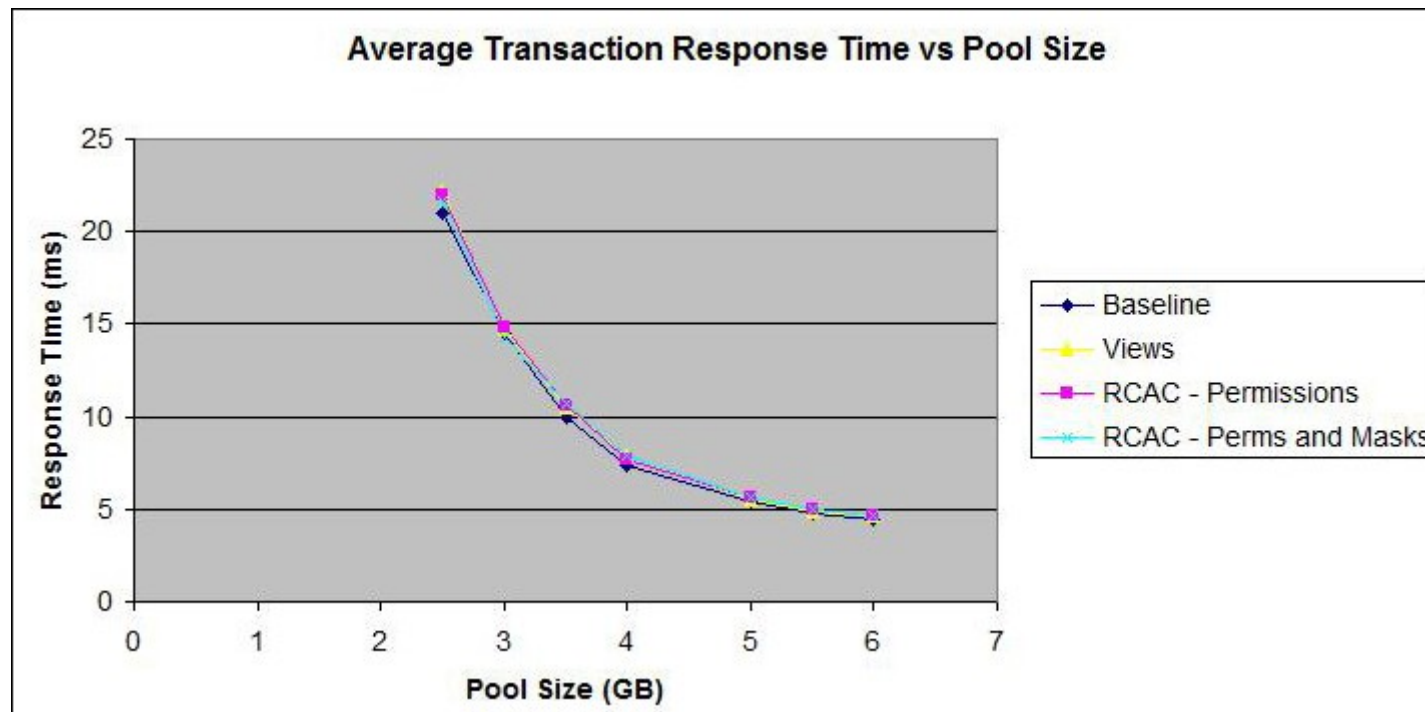
Création	Nom de table	Schéma	Type d'index	Colonnes
<input checked="" type="checkbox"/>	USEROK	BDVINB	Base binaire	APPEL_CODE
<input checked="" type="checkbox"/>	USEROK	BDVINB	Base binaire	USERNAME APPEL_CODE



# RCAC

*Developer Work's* signale un surplus en CPU

- Négligeable (< à 1 %) par rapport à une vue réalisant les mêmes services
- Moins de 5 %, par rapport à une lecture sans contrôle



# RCAC

## Commandes système

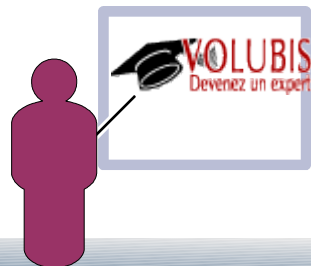
- les droits RCAC sont stockés dans la table elle même, ils sont donc :

sauvegardés par SAVLIB, SAVOBJ  
déplacés par MOV OBJ  
dupliqués (par défaut) par CRTDUPOBJ

## CRTDUPOBJ et CPYLIB

Quand les Data sont dupliquées, elles le sont à l'identique (en clair) , les droits RCAC étant aussi dupliqués, mais il y a un nouveau paramètre sur CRTDUPOBJ

```
Duplication des données . . . . DATA *NO
Dupliquer des contraintes . . . . CST *YES
Dupliquer des déclencheurs . . . . TRG *YES
ID de fichiers en double . . . . FILEID *NO
Contrôle d'accès en double . . . . ACCCTL ← *ALL
```



# RCAC

Commandes système

- CPYF

Seules les données étant copiées (pas les droits RCAC), ne sont dupliqués que les lignes autorisées, et éventuellement masquées.

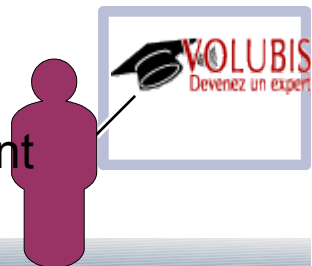
- Réplication manuelle de fichiers, ou ETL :

Le problème est le même que CPYF, il faut donc explicitement autoriser l'utilisateur OU subir les effets de RCAC !

- Attention aux droits

Les données dans le journal et dans une MQT sont en clair (la réplication basée sur la fonction journal se passera bien)

Mais il faut limiter le droit de lire le récepteur, et le droit de lire explicitement une MQT, sinon vous aurez une faille de sécurité !



# RCAC

## ATTENTION !

une table (ou fichier physique) avec des droits RCAC ne peut pas être sauvegardée pour version précédente.

une table (ou fichier physique) avec des droits RCAC, restaurée sur un système ne possédant pas l'option 47 ne peut plus être **ouverte**.

**Merci !**

