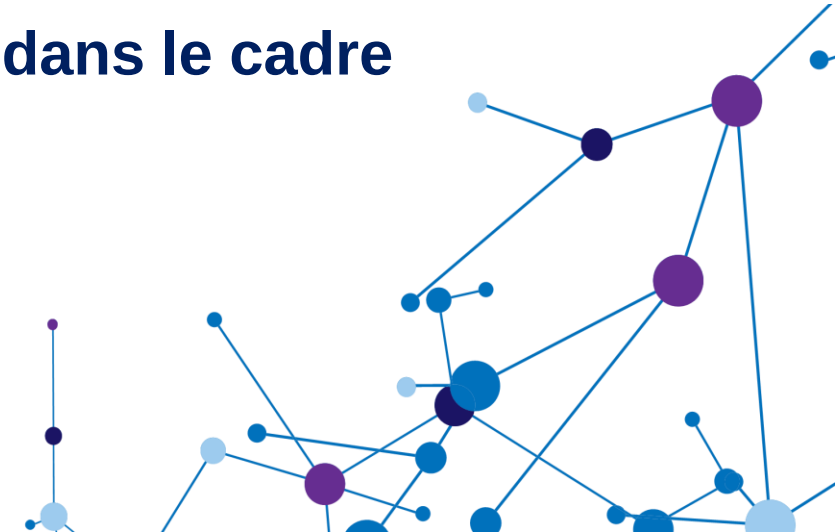
Université IBM i 2018

16 et 17 mai

IBM Client Center Paris

S20 – Nouveautés sécurité IBM i V7dans le cadre de la GDPR

Philippe Bourgeois
IBM France
pbourgeois@fr.ibm.com



Plan de la présentation

- Quelques mots sur la GDPR et l'IBM i
- Les fonctions de sécurité natives IBM i
  - Collectes de droits
  - Fonctions de cryptage SQL
  - FIELDPROC
  - RCAC
  - Tables temporelles

Quelques mots sur la GDPR et l'IBM i

La GDPR

- GDPR : **G**ENERAL **D**ATA **P**ROTECTION **R**EGULATION
  - En français : RGPD (Règlement Général sur la Protection des Données)
- Nouvelle règlementation européenne adoptée le 27 avril 2016 applicable à partir du **25 mai 2018**
  - Donne aux particuliers (« personnes physiques ») le **contrôle** de leurs informations personnelles et garantit la **protection** de ces informations dans le cadre du traitement et du mouvement des données
- Une obligation au sein de l'UE
- Pénalités et amendes : jusqu'à 4% du chiffre d'affaires global ou 20 M€

La GDPR

- Concerne toute entreprise, organisme, particulier qui collecte, manipule ou stocke des **informations personnelles** sur des citoyens ou des résidents de l'Union Européenne
  - **Information personnelle** : *"donnée identifiée ou identifiable d'une personne physique permettant une identification directe ou indirecte, par référence à un numéro d'identification ou à un ou plusieurs éléments qui lui sont propres"*

➔ Vous devez veiller à ce que des mesures **techniques**, **organisationnelles** et **juridiques** appropriées soient mises en place pour assurer conformité et transparence sur l'utilisation faite des données personnelles

La GDPR : les points-clés

- Identifier et maîtriser les **données**
- Identifier et maîtriser les **traitements**
- Mettre en place les moyens de **protection** et en faire la preuve
- Etre en capacité de **répondre** à toute **demande** (CNIL, usagers)

Traitement d'une donnée : toute utilisation de cette donnée (collecte, enregistrement, organisation, conservation, adaptation, modification, extraction, consultation, utilisation, communication, transmission, diffusion)

GDPR et IBM i

- **1. Protection** des données personnelles – **Confidentialité**
  - Contrôle des droits d'accès
  - Cryptage
  - Masquage
  - Anonymisation

- **2. Traçabilité** des accès aux données
  - Historisation des accès
  - Historisation des données

GDPR et IBM i – Protection des données personnelles

- **Contrôle des droits** d'accès aux fichiers contenant des données personnelles
- Les droits sur les fichiers et sur les données de ces fichiers proviennent :
  - Du droit spécial *ALLOBJ du profil
  - Des droits privés sur le fichier
  - Des droits issus du (ou des) groupe(s) auquel(s) appartient le profil
  - Des droits publics

- Deux fonctionnalités natives IBM i peuvent être utiles :
  - Les **collectes de droits** (authority collection)
    - Pour vous aider à identifier les droits réellement nécessaires
  - Le contrôle d'accès niveau ligne et colonne (**RCAC** - Row and Column Access Control)
    - Pour limiter l'accès à certaines lignes et certaines colonnes de tables DB2

GDPR et IBM i – Protection des données personnelles

- **Cryptage** des données personnelles

- Cryptage des **disques**
  - Cryptage des ASP (SS1 Option 45 – Encrypted ASP Enablement)

- Cryptage des **sauvegardes**
  - BRMS : BRMS Advanced Feature (BR1 Option 2)
  - SS1 Option 44 – IBM i Encrypted Backup Enablement

- Cryptage des **fichiers base de données**
  - Fonctions SQL de cryptage
  - Field Procedure (FIELDPROC)

- Cryptage de **données par programme**
  - APIs de cryptographie

GDPR et IBM i – Protection des données personnelles

- **Masquage** des données personnelles
  - Masquage du contenu de colonnes
    - Masques de colonnes de tables DB2 (Column Access Control – R**CAC**)
- **Anonymisation** et pseudonymisation de données
  - Pseudonymisation : réversible
  - Anonymisation : non réversible → les données deviennent non personnelles
  - Pour les données sur les environnements de développement, de test, de recette
  - Il existe de multiples solutions tierces pour anonymiser / pseudonymiser les données (Arcad Software, Itheis, Syncsort…)

ENJEUX

1. Vous devez protéger la prod

2. Aucune donnée ne doit sortir de la prod sans être anonymisée

GDPR et IBM i – Traçabilité des accès aux données

- Journaux bases de données
- Journal d'audit (QAUDJRN)
- Triggers
- Points d'exit
- Plusieurs outils tiers permettent d'exploiter ces données
- Il existe un outil IBM pour tracer l'activité des accès base de données : IBM Guardium (https://www-03.ibm.com/systems/data/flash/fr/resources/universite_i_2016/S27_-_Utiliser_IBM_Guardium_pour_tracker_l%20activite_de_DB2_for_i.pdf)
- L'historisation des données d'une table DB2 et son exploitation peut se faire en par la mise en œuvre des **tables temporelles**

GDPR et IBM i – Avant toute chose

Le top 10 des erreurs de configuration de sécurité IBM i

Les bonnes pratiques de sécurité IBM i

- A. Le top 10 des erreurs de configuration de sécurité IBM i
  - 1. Mots de passe par défaut
  - 2. Profils avec droits publics *USE
  - 3. Droits publics par défaut des objets
  - 4. Adoption de droits
  - 5. Droits spéciaux
  - 6. Ligne de commande
  - 7. Services réseau
  - 8. Profil invité NetServer
  - 9. Droits IFS
  - 10. DDM / DRDA

- B. Les bonnes pratiques de sécurité IBM i
  - 1. Niveau de sécurité système
  - 2. Valeurs système
  - 3. Mots de passe
  - 4. Profils utilisateur
  - 5. Sécurité des ressources
  - 6. Audit
  - 7. Autres bonnes pratiques

Les fonctions de sécurité natives IBM i

GDPR et IBM i – Focus sur quelques fonctionnalités

- 1. Collectes de droits
- 2. Fonctions de cryptage SQL
- 3. FIELDPROC
- 4. RCAC
- 5. Tables temporelles

1. Collectes de droits

- Rappel : lorsque l'on accède à un objet, les étapes suivantes sont réalisées pour contrôler les droits sur cet objet

Les étapes	Ordre
Le profil est il *ALLOBJ ?	1
Le profil a-t-il des droits privés sur l'objet ?	2
Le profil est-il indiqué dans une liste d'autorisation qui protège l'objet ?	3
Le groupe est il *ALLOBJ ?	4
Le groupe a-t-il des droits sur l'objet ?	5
Le groupe est-il indiqué dans une liste d'autorisation qui protège l'objet ?	6
*PUBLIC est-il indiqué sur l'objet ?	7
*PUBLIC est-il indiqué dans une liste d'autorisation qui protège l'objet ?	8

- Jusqu'à la 7.3 on avait un problème de suivi : on ne savait pas toujours quel mécanisme était utilisé quand un utilisateur accédait à un objet

1. Collectes de droits

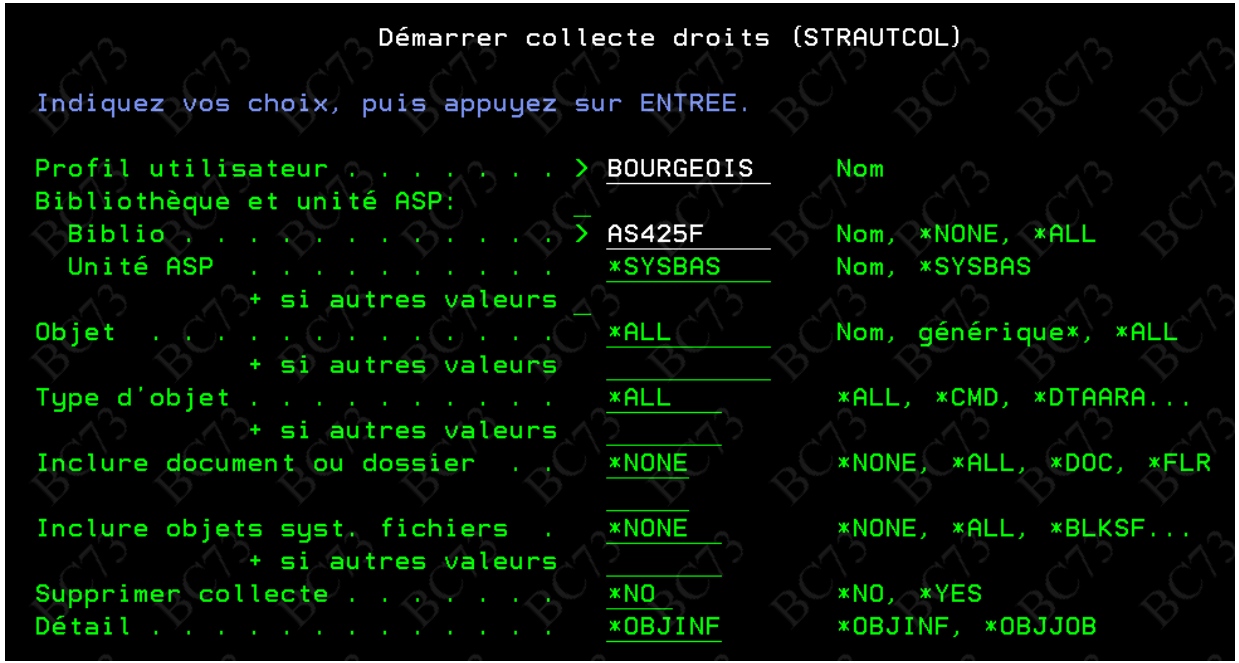
- Permettent de mieux comprendre les mécanismes d'attribution des droits réellement mis en œuvre (utile pour n'octroyer que les droits nécessaires aux utilisateurs)
- Une collecte de droits permet de tracer les accès d'un utilisateur à certains objets
  - Démarrage de la collecte de droits pour un utilisateur
  - Exécution des applications
  - Arrêt de la collecte
  - Affichage de la collecte
- Pour un objet donné, une collecte permet de connaître
  - Les droits utilisés par le profil
  - La source de ces droits
  - Les droits minimum nécessaires
- Prérequis : IBM i **7.3**

1. Collectes de droits

- 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

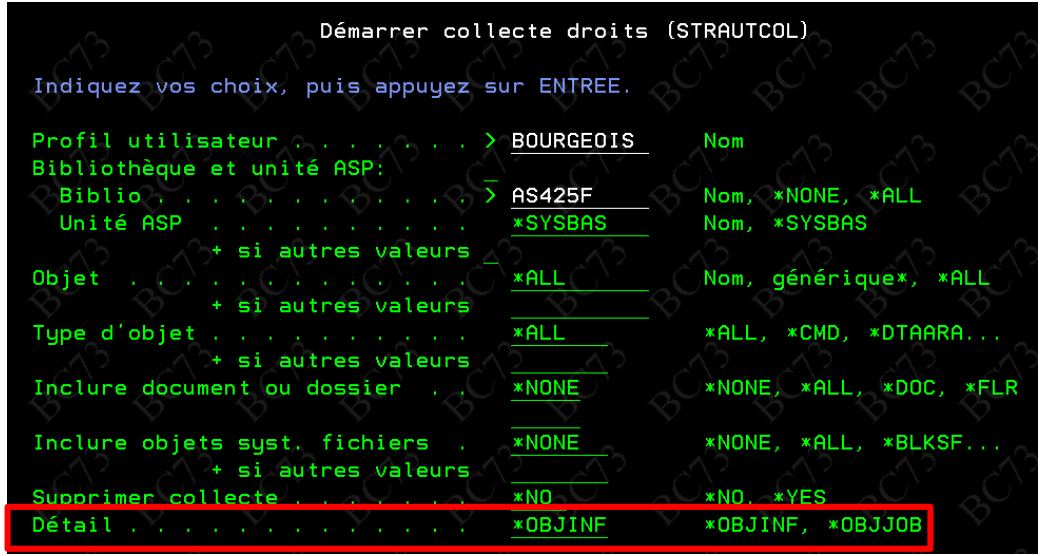
1. Collectes de droits

- Utilisateurs et groupes
  - Utilisateurs
  - Groupes
  - Création utilisateur
  - Création groupe
  - Utilisateur - Propriétés
  - Gestion des collectes
    - Démarrer la collecte des droits
    - Arrêter la collecte des droits
    - Afficher la collecte des droits
    - Supprimer la collecte des droits



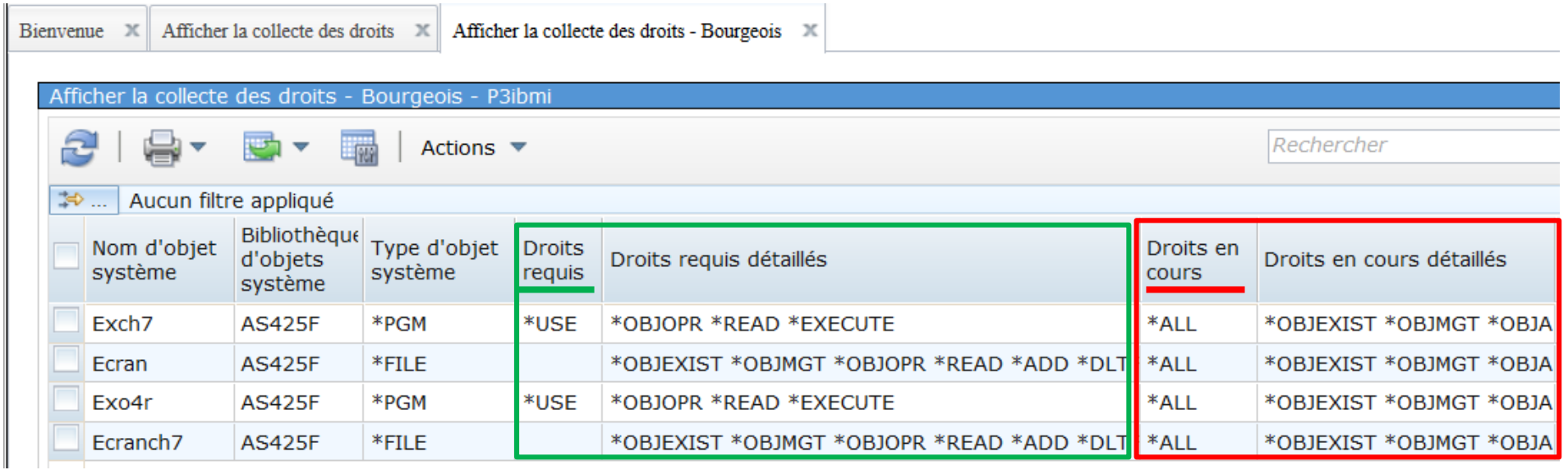
1. Collectes de droits

- Paramètre DETAIL :
  - OBJINF : si un objet est accédé plusieurs fois par le profil avec les mêmes droits, un seul enregistrement est créé
  - *OBJJOB : si un objet est accédé plusieurs fois par le profil avec les mêmes droits, tous les accès sont tracés



1. Collectes de droits

- Affichage d'une collecte



Nom d'objet système	Bibliothèque d'objets système	Type d'objet système	Droits requis	Droits requis détaillés	Droits en cours	Droits en cours détaillés
Exch7	AS425F	*PGM	*USE	*OBJOPR *READ *EXECUTE	*ALL	*OBJEXIST *OBJMGT *OBJA
Ecran	AS425F	*FILE		*OBJEXIST *OBJMGT *OBJOPR *READ *ADD *DLT	*ALL	*OBJEXIST *OBJMGT *OBJA
Exo4r	AS425F	*PGM	*USE	*OBJOPR *READ *EXECUTE	*ALL	*OBJEXIST *OBJMGT *OBJA
Ecranch7	AS425F	*FILE		*OBJEXIST *OBJMGT *OBJOPR *READ *ADD *DLT	*ALL	*OBJEXIST *OBJMGT *OBJA

1. Collectes de droits

- Choix des objets à collecter

```
-- Démarrage de la collecte
cl:STRAUTCOL USRPRF(LJ) LIBINF((POTGDPRA)) OBJ(N05 G22) OBJTYPE(*FILE) DLTCOL(*YES);

-- Exécution en 5250 sous le profil LJ : GO POTMENU options 1 et 2

-- Arrêt de la collecte
cl:ENDAUTCOL USRPRF(LJ);
```

Afficher la collecte des droits - Lj - Ibmi73

Actions | Rechercher

Aucun filtre appliqué

Nom d'objet système	Bibliothèque d'objets système	Type d'objet système	Droits requis	Droits requis détaillés	Droits en cours	Droits adoptés en cours détaillés	Source de droits
G22	POTGDPRA	*FILE		*READ	*CHANGE	*OBJOPR *READ *AD	PUBLIC
N05	POTGDPRA	*FILE		*READ	*CHANGE	*OBJOPR *READ *AD	PUBLIC
N05	POTGDPRA	*FILE		*READ *UPD	*CHANGE	*OBJOPR *READ *AD	PUBLIC

1. Collectes de droits

- Requêtage SQL

  - Pour un **profil**

```
SELECT * FROM QSYS2.AUTHORITY_COLLECTION
WHERE USER_NAME = 'xxx'
```

  - Pour un **objet**

```
SELECT * FROM QSYS2.AUTHORITY_COLLECTION
 WHERE SYSTEM_OBJECT_NAME = 'xxx' AND
       SYSTEM_OBJECT_SCHEMA = 'yyy'
```

1. Collectes de droits

- Droits requis et droits en cours

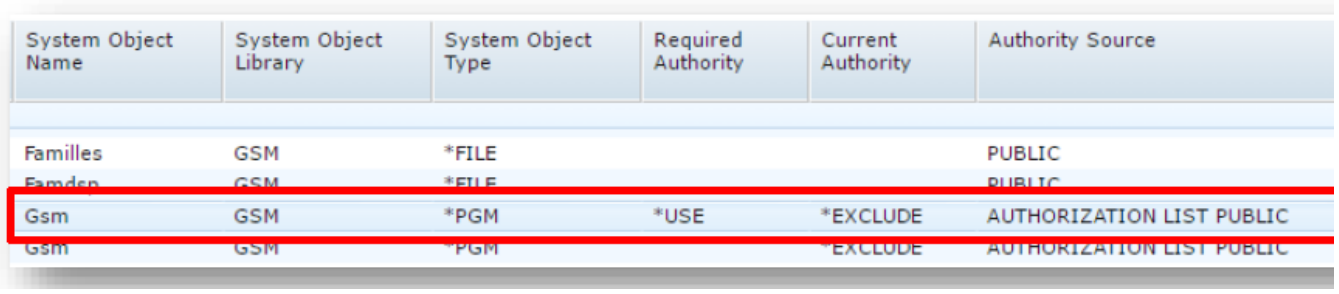
```
13 -- Droits requis et droits en cours
14 SELECT system_object_name,
15          required_authority, detailed_required_authority,
16          current_authority, detailed_current_authority
17  FROM qsys2.authority_collection WHERE authorization_name = 'LJ';
18
```

SYSTEM_OBJECT_NAME	REQUIRED_AUTHORITY	DETAILED_REQUIRED_AUTHORITY	CURRENT_AUTHORITY
G22	-	*READ	*CHANGE
N05	-	*READ	*CHANGE
N05	-	*READ *UPD	*CHANGE

DETAILED_CURRENT_AUTHORITY
*OBJOPR *READ *ADD *DLT *UPD *EXECUTE
*OBJOPR *READ *ADD *DLT *UPD *EXECUTE
*OBJOPR *READ *ADD *DLT *UPD *EXECUTE

1. Collectes de droits

- Source des droits
  - Exemple 1 - Non autorisé par liste d'autorisations



System Object Name	System Object Library	System Object Type	Required Authority	Current Authority	Authority Source
Familles	GSM	*FILE			PUBLIC
Famdsp	GSM	*FILE			PUBLIC
Gsm	GSM	*PGM	*USE	*EXCLUDE	AUTHORIZATION LIST PUBLIC
Gsm	GSM	*PGM		*EXCLUDE	AUTHORIZATION LIST PUBLIC

  - Exemple 2 – Droits *EXCLUDE – Autorisé par liste d'autorisation – Héritage de *ALLOBJ

System Object Name	System Object Library	System Object Type	Required Authority	Current Authority	Authority Source	Adopted Authority Source	Current Adopted Authority	Authority Check Successful
Familles	GSM	*FILE			PUBLIC			x
Famdsp	GSM	*FILE			PUBLIC			x
Gsm	GSM	*PGM	*USE	*EXCLUDE	AUTHORIZATION LIST PUBLIC			
Gsm	GSM	*PGM		*EXCLUDE	AUTHORIZATION LIST PUBLIC			
Gsm	GSM	*PGM	*USE	*EXCLUDE	AUTHORIZATION LIST PUBLIC			
Gsm	GSM	*PGM		*EXCLUDE	AUTHORIZATION LIST PUBLIC			
Gsm	GSM	*PGM	*USE	*USE	PUBLIC			x
Securinit	GSM	*PGM		*EXCLUDE	AUTHORIZATION LIST PUBLIC	ADOPTED *ALLOBJ	*ALL	x
Securinit	GSM	*PGM		*EXCLUDE	AUTHORIZATION LIST PUBLIC	ADOPTED *ALLOBJ	*ALL	x

1. Collectes de droits

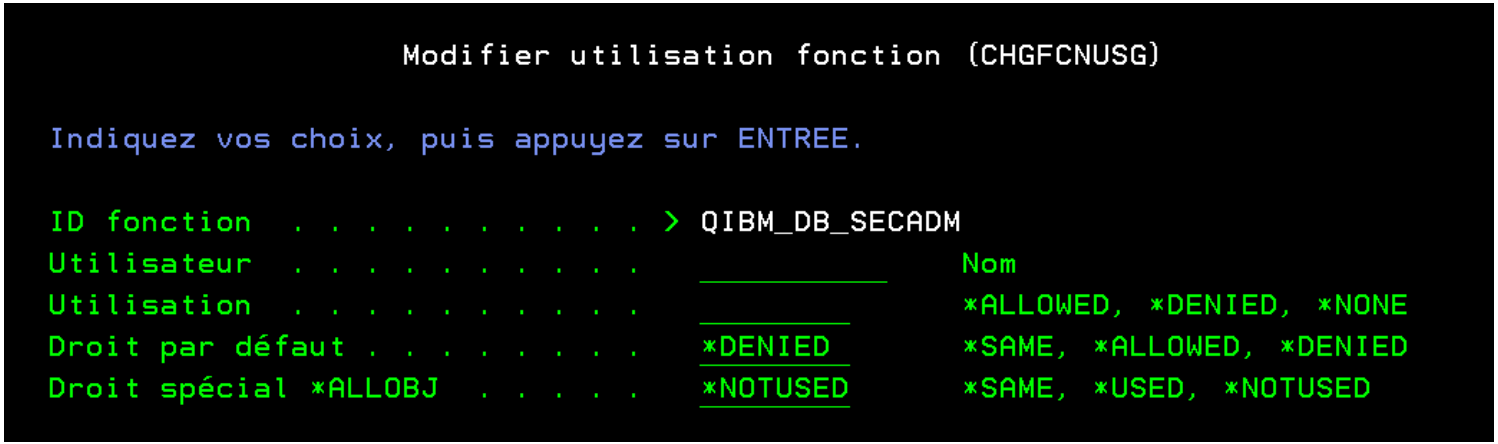
- Pour connaitre les profils audités :

```
SELECT AUTHORIZATION_NAME, TEXT_DESCRIPTION
FROM QSYS2.USER_INFO
WHERE AUTHORITY_COLLECTION_ACTIVE = 'YES'
```

- La commande DSPUSRPRF a deux nouveaux attributs permettant de savoir :
  - si une collecte est active pour ce profil
  - s'il existe une collecte, même inactive

1. Collectes de droits

- Restrictions - Pour pouvoir gérer les collectes de droits :
  - *ALLOBJ nécessaire
  - Ou être autorisé à la fonction **QIBM_DB_SECADM** (“Database Security Administrator”). Pour autoriser cette fonction :
    - IBM Navigator for i (Administration d'applications)
    - WRKFCNUSG



2. Les fonction de cryptage SQL

- Fonctions **ENCRYPT_xxx** et **DECRYPT_yyy** pour crypter et décrypter le contenu d'une colonne
- Fonction SET ENCRYPTION PASSWORD pour définir le mot de passe
  - Ou directement dans les fonctions ENCRYPT_xxx et DECRYPT_xxx
- Fonctionnalité **non** Data Centric :
  - N'est valable qu'en SQL
  - Il faut modifier les applications et la base de données
  - Il faut gérer applicativement les accès natifs (RPG, COBOL…)
- Prérequis : IBM i **5.4**

2. Les fonction de cryptage SQL

```
SET ENCRYPTION PASSWORD 'gdpr_ibmi';

INSERT INTO pbol39.patients3 VALUES('MAX', 'Maxime', ENCRYPT_AES('1701144055080'), 'Codeine', 'LEE');
INSERT INTO pbol39.patients3 VALUES('MIKE', 'Mike', ENCRYPT_AES('1630256077020'), 'Doliprane', 'JAMES');
INSERT INTO pbol39.patients3 VALUES('SAM', 'Samantha', ENCRYPT_AES('2780687012070'), 'Efferalgan', 'LEE');
```

```
SELECT * FROM pbol39.patients3;
```

PROFIL	PRENOM	NUMSECU	TRAITEMENT	MEDECIN
MAX	Maxime	4C946CFF0129D5A6B96180506FFE4F23B96180506FFE4F23EEA0D62C61DA2FB1F86EE59A21FA5265	Codeine	LEE
MIKE	Mike	4C7F81FF0129D5A6B96180506FFE4F23B96180506FFE4F238B6CA25E0B6DD80AC53145A0BB1994F7	Doliprane	JAMES
SAM	Samantha	4CAA56FF0129D5A6B96180506FFE4F23B96180506FFE4F23CA9D40168E5900AAFB5A4D009125C14F	Efferalgan	LEE

```
SET ENCRYPTION PASSWORD 'gdpr_ibmi';

SELECT profil, prenom, DECRYPT_CHAR(numsecu) as numsecu, traitement, medecin FROM pbol39.patients3;
```

PROFIL	PRENOM	NUMSECU	TRAITEMENT	MEDECIN
MAX	Maxime	1701144055080	Codeine	LEE
MIKE	Mike	1630256077020	Doliprane	JAMES
SAM	Samantha	2780687012070	Efferalgan	LEE

2. Les fonction de cryptage SQL

- La zone doit être définie avec la clause FOR BIT DATA
  - Et il faut modifier sa longueur pour stocker le mot de passe
- Le mot de passe peut être défini par la clause SET ENCRYPTION PASSWORD
  - L'ajout de la clause WITH HINT xxx permet de définir une astuce
    - Et l'instruction GETHINT permet de récupérer l'astuce
- Il faut gérer applicativement les accès natifs (RPG, COBOL…)
  - Pour crypter : créer un trigger BEFORE qui va lancer la fonction ENCRYPT_xxx
  - Pour décrypter : créer une vue SQL qui va lancer la fonction DECRYPT_yyy et déclarer cette vue dans les programmes

3. FIELDPROC

- Programme de cryptage/décryptage rattaché à une colonne d'une table, qui sera appelé :
  - Lors des insertions et mises à jour pour crypter le contenu de la colonne
  - Lors des lectures pour décrypter le contenu de la colonne
- Fonctionnalité Data-Centric
  - N'implique pas de modifier les applications, ni la base de données
  - Est valable quelle que soit l'interface d'accès à la table (accès natifs, accès SQL, commandes CL…)
- Le décryptage peut se faire sous conditions
- Le programme peut être
  - Écrit par vous-même
  - Acheté (Syncsort, Linoma, Raz-Lee, Townsend Security)
- Prérequis : IBM i **7.1**

3. FIELDPROC

```
ALTER TABLE pbol39.patients2 ALTER numsecu SET FIELDPROC pbol39.fldprocpat;
```

```
SELECT * FROM pbol39.patients2;
```

PROFIL	PRENOM	NUMSECU	TRAITEMENT	MEDECIN
MAX	Maxime	0805504411071	Codeine	LEE
MIKE	Mike	0207706520361	Doliprane	JAMES
SAM	Samantha	0702107860872	Efferalgan	LEE
DOUG	Doug	0606606370951	Dafalgan	JAMES
AMY	Amy	0807704350882	Voltarene	LEE

Ici le programme de cryptage ne fait "qu'inverser les caractères"

```
....+....1....+....2....+....3....+....4....+...
PROFIL     PRENOM     NUMSECU
MAX        Maxime     08055044110 71
MIKE       Mike       02077065203 61
SAM        Samantha   07021078608 72
DOUG       Doug       06066063709 51
AMY        Amy        08077043508 82
```

3. FIELDPROC

```
ALTER TABLE patients2b ALTER numsecu SET FIELDPROC pbol39.fldprocc;
```

```
SET SESSION_USER = 'BOURGEOIS';
SELECT * FROM patients2b;
```

PROFIL	PRENOM	NUMSECU	TRAITEMENT	MEDECIN
MAX	Maxime	□□□□□□□□□□□□	Codeine	LEE
MIKE	Mike	□□□□□□□□□□	Doliprane	JAMES
SAM	Samantha	□□□□□□□□□□	Efferalgan	LEE
DOUG	Doug	□□□□□□□□□□□□	Dafalgan	JAMES
AMY	Amy	□□□□□□□□□□□	Voltarene	LEE

```
SET SESSION_USER = 'QSECOFR';
SELECT * FROM patients2b;
```

PROFIL	PRENOM	NUMSECU	TRAITEMENT	MEDECIN
MAX	Maxime	1701144055080	Codeine	LEE
MIKE	Mike	1630256077020	Doliprane	JAMES
SAM	Samantha	2780687012070	Efferalgan	LEE
DOUG	Doug	1590736066060	Dafalgan	JAMES
AMY	Amy	2880534077080	Voltarene	LEE

3. FIELDPROC

FLDPROCC.RPGLE

```
 Ligne 35      Colonne 31    Remplacement
      ....+....1....+....2....+....3....+....4....+....5....+....6....+....7....
000100       ctl-opt dftactgrp(*no) actgrp(*caller);
000200       ctl-opt stgmdl(*inherit) thread(*concurrent);
000300
000400      /copy QSYSINC/QRPGLESRC,SQLFP
000500
000600       dcl-s lg uns(5);
000700
000800       dcl-ds *n psds;
000900         profil char(10) pos(358);
001000       end-ds;
001100
001200       dcl-pi *n;
001300         contexte bindec(2);          // Contexte
001400         optparms likeds(sqlfopvd);   // Description des paramètres
001500         decoded_attr likeds(sqlfpd); // Attributs de la donnée décodée
001600         decoded_data char(512);      // Donnée décodée
001700         encoded_attr likeds(sqlfpd); // Attributs de la donnée encodée
001800         encoded_data char(512);      // Donnée encodée
001900         sqlstate char(5);            // SQLSTATE
002000         sqlmsg likeds(sqlfmt);       // Texte de message
002100       end-pi;
002200
002300       SQLSTATE = '00000';
002400       select;
002500         when contexte = 8;  // Enregistrement du FIELDPROC
002600          // Le type retourné est le même, donc copie de la définition
002700          encoded_attr = decoded_attr;
002800
002900         when contexte = 0;  // Ecriture => encodage
003000           lg = decoded_attr.SQLFL;
003100           encode_decode(decoded_data : encoded_data : lg);
003200
```

3. FIELDPROC

```
003300          when contexte = 4 ; // Lecture => décodage
003400            lg = encoded_attr.SQLFL;
003500            if profil = 'QSECOFR';
003600              encode_decode(encoded_data : decoded_data : lg);
003700            else;
003800              %subst(decoded_Data:1:lg) = %subst(encoded_Data:1:lg);
003900            endif;
004000          other;
004100            SQLSTATE = '38001';
004200            sqlmsg = 'Demande non prise en charge';
004300        endsl;
004400        *inlr = *on;
004500
004600        dcl-proc encode_decode;
004700        dcl-pi *n;
004800          data1 char(512);
004900          data2 char(512);
005000          lg uns(5);
005100        end-pi;
005200
005300        dcl-s i uns(5);
005400        for i = 1 to lg;
005500          if %subst(data1:i:1)  = ' ';   // la doc déconseille de crypter les espaces
005600            %subst(data2:i:1) = %subst(data1:i:1);
005700          else;
005800            %subst(data2:i:1) = %bitnot(%subst(data1:i:1));
005900          endif;
006000        endfor;
006100        end-proc;
```

3. FIELDPROC

- Le programme doit être de type *PGM ILE et ne pas contenir de SQL
- Il est appelé dans 3 cas :
  - Création/modification de la colonne (validation du type)
  - Ecriture (codage du contenu, compression…)
  - Lecture (décodage sous conditions…)
- Les index sur les colonnes avec FIELDPROC sont encodés
  - Attention : les tris sur la colonne peuvent être perturbés
- Un CHGPF perd les FIELDPROC

4. RCAC

- **RCAC** : Row and Column Access Control
- Couche additionnelle de sécurité, complémentaire à la sécurité niveau table
- Permet de limiter l'accès à certaines données (certaines lignes et/ou certaines colonnes d'une table). Ajout par SQL de règles :
  - PERMISSION : pour restreindre l'accès aux lignes
  - MASK : pour restreindre l'accès aux colonnes
- Se définit au niveau de la base de données (approche « Data Centric »)
  - S'applique quelque soit l'interface d'accès à la table
  - Ne nécessite pas la modification des applications
  - Personne n'y échappe

4. RCAC

- Exemple : Assurance santé – On veut sécuriser la table PATIENTS

PROFIL	PRENOM	NUMSECU	TRAITEMENT	MEDECIN
MAX	Maxime	1701144055080	Codeine	LEE
MIKE	Mike	1630256077020	Doliprane	JAMES
SAM	Samantha	2780687012070	Efferalgan	LEE
DOUG	Doug	1590736066060	Dafalgan	JAMES
AMY	Amy	2880534077080	Voltarene	LEE

- Scénario pour définir les règles :
  - Les **patients** ne peuvent voir que leurs propres données
  - Les **médecins** ne peuvent voir les données que de leurs patients
  - Les **chercheurs** peuvent visualiser toutes les données
  - Les **autres** personnes ne peuvent pas visualiser les données
  - Seul le **patient** peut voir son **numéro de sécurité sociale**

Permissions (droits sur les lignes)

Masque de colonne

4. RCAC

- Exemple : Assurance santé – On veut sécuriser la table PATIENTS

PROFIL	PRENOM	NUMSECU	TRAITEMENT	MEDECIN
MAX	Maxime	1701144055080	Codeine	LEE
MIKE	Mike	1630256077020	Doliprane	JAMES
SAM	Samantha	2780687012070	Efferalgan	LEE
DOUG	Doug	1590736066060	Dafalgan	JAMES
AMY	Amy	2880534077080	Voltarene	LEE

- Les profils IBM i
  - Trois profils de groupe :
    - PATIENTS
    - MEDECINS
    - CHERCHEURS
  - Les patients (MAX, MIKE, DOUG…) sont rattachés au profil de groupe PATIENTS
  - Les médecins (LEE et JAMES) sont rattachés au profil de groupe MEDECINS
  - Le chercheur BOB est rattaché au profil de groupe CHERCHEURS

4. RCAC

- Exemple – Droits sur les **lignes** : création d'une **permission**

```
CREATE PERMISSION pbol39.row_access ON pbol39.patients FOR ROWS WHERE

 (VERIFY_GROUP_FOR_USER(SESSION_USER, 'PATIENTS') = 1 AND profil = SESSION_USER)
 OR
 (VERIFY_GROUP_FOR_USER(SESSION_USER, 'MEDECINS') = 1 AND medecin = SESSION_USER)
 OR
 (VERIFY_GROUP_FOR_USER(SESSION_USER, 'CHERCHEURS') = 1)

ENFORCED FOR ALL ACCESS ENABLE;


ALTER TABLE pbol39.patients ACTIVATE ROW ACCESS CONTROL;
```

- On définit des règles
- Toutes les lignes qui ne correspondent pas à la règle ne sont pas renvoyées

4. RCAC

- Exemple – Droits sur les **lignes** – Création d'une **permission**
  - Connexion sous le profil BOURGEOIS (*ALLOBJ – rattaché à aucun profil de groupe)

```
  Affichage d'un travail

Utilisateur:   BOURGEOIS
```

```
CREATE PERMISSION pbol39.row_access ON pbol39.patients FOR ROWS WHERE

(VERIFY_GROUP_FOR_USER(SESSION_USER, 'PATIENTS') = 1 AND profil = SESSION_USER)
OR
(VERIFY_GROUP_FOR_USER(SESSION_USER, 'MEDECINS') = 1 AND medecin = SESSION_USER)
OR
(VERIFY_GROUP_FOR_USER(SESSION_USER, 'CHERCHEURS') = 1)
```

```
> select * from pbol39.patients
```

BOURGEOIS n' aucun droit sur les lignes du fichier PATIENTS

```
                                Affichage des données
                                                          Largeur des données . . :    91
Première ligne à afficher . . _____                      Première colonne à afficher  . _____
....+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8....+....9.
PROFIL      PRENOM              NUMSECU        TRAITEMENT                    MEDECIN
********  Fin de données  ********
```

4. RCAC

- Exemple – Droits sur les **lignes** – Création d'une **permission**
  - Connexion sous le profil BOB (rattaché au profil de groupe CHERCHEURS)

```
  Affichage d'un travail

Utilisateur:   BOB
```

```
> select * from pbol39.patients
```

```
CREATE PERMISSION pbol39.row_access ON pbol39.patients FOR ROWS WHERE

 (VERIFY_GROUP_FOR_USER(SESSION_USER, 'PATIENTS') = 1 AND profil = SESSION_USER)
 OR
 (VERIFY_GROUP_FOR_USER(SESSION_USER, 'MEDECINS') = 1 AND medecin = SESSION_USER)
 OR
 (VERIFY_GROUP_FOR_USER(SESSION_USER, 'CHERCHEURS') = 1)
```

BOB peut visualiser toutes les lignes du fichier PATIENTS

```
                                Affichage des données
                                                    Largeur des données . . :    91
Première ligne à afficher . . _____                Première colonne à afficher . _____
....+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8....+....9.
PROFIL     PRENOM              NUMSECU         TRAITEMENT                MEDECIN
MAX        Maxime              1701144055080   Codeine                   LEE
MIKE       Mike                1630256077020   Doliprane                 JAMES
SAM        Samantha            2780687012070   Efferalgan                LEE
DOUG       Doug                1590736066060   Dafalgan                  JAMES
AMY        Amy                 2880534077080   Voltarene                 LEE
********  Fin de données  ********
```

4. RCAC

- Exemple – Droits sur les **lignes** – Création d'une **permission**
  - Connexion sous le profil LEE (rattaché au profil de groupe MEDECINS)

```
  Affichage d'un travail

Utilisateur:   LEE
```

```
> select * from pbol39.patients
```

```
CREATE PERMISSION pbol39.row_access ON pbol39.patients FOR ROWS WHERE

  (VERIFY_GROUP_FOR_USER(SESSION_USER, 'PATIENTS') = 1 AND profil = SESSION_USER)
  OR
  (VERIFY_GROUP_FOR_USER(SESSION_USER, 'MEDECINS') = 1 AND medecin = SESSION_USER)
  OR
  (VERIFY_GROUP_FOR_USER(SESSION_USER, 'CHERCHEURS') = 1)
```

LEE ne peut visualiser que les patients dont il s'occupe

```
                                Affichage des données
                                                        Largeur des données . . :    91
Première ligne à afficher . . _____                    Première colonne à afficher . _____
....+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8....+....9.
PROFIL      PRENOM                NUMSECU        TRAITEMENT                      MEDECIN
MAX         Maxime                1701144055080  Codeine                         LEE
SAM         Samantha              2780687012070  Efferalgan                      LEE
AMY         Amy                   2880534077080  Voltarene                       LEE
********  Fin de données  ********
```

4. RCAC

- Exemple – Droits sur les **lignes** – Création d'une **permission**
  - Connexion sous le profil LEE (rattaché au profil de groupe MEDECINS)

```
Connecté à la base de données relationnelle Ibmi73 sur 9.128.137.198 en tant que LEE
```

LEE ne peut pas mettre à jour une ligne qu'il n'a pas droit de lire

```
44  UPDATE pbol39.patients SET traitement = 'Codeine' WHERE profil = 'DOUG';
```

```
⚠ Etat SQL : 02000
Code fournisseur : 100
Message : [SQL0100] La ligne n'a pas été trouvée pour UPDATE. Cause . . . . . : L
```

PROFIL	PRENOM	NUMSECU	TRAITEMENT	MEDECIN
MAX	Maxime	1701144055080	Codeine	LEE
MIKE	Mike	1630256077020	Doliprane	JAMES
SAM	Samantha	2780687012070	Efferalgan	LEE
DOUG	Doug	1590736066060	Dafalgan	JAMES
AMY	Amy	2880534077080	Voltarene	LEE

4. RCAC

- Exemple – Droits sur les **lignes** – Création d'une **permission**
  - Connexion sous le profil MIKE (rattaché au profil de groupe PATIENTS)

```
  Affichage d'un travail

Utilisateur:   MIKE
```

```
> select * from pbol39.patients
```

```
CREATE PERMISSION pbol39.row_access ON pbol39.patients FOR ROWS WHERE

(VERIFY_GROUP_FOR_USER(SESSION_USER, 'PATIENTS') = 1 AND profil = SESSION_USER)
OR
(VERIFY_GROUP_FOR_USER(SESSION_USER, 'MEDECINS') = 1 AND medecin = SESSION_USER)
OR
(VERIFY_GROUP_FOR_USER(SESSION_USER, 'CHERCHEURS') = 1)
```

MIKE ne peut visualiser que ses données personnelles

```
                                Affichage des données
                                                          Largeur des données . . :    91
Première ligne à afficher . . :                     Première colonne à afficher . :
      ...+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8....+....9.
PROFIL     PRENOM          NUMSECU        TRAITEMENT                    MEDECIN
MIKE       Mike            1630256077020  Doliprane                     JAMES
******** Fin de données ********
```

4. RCAC

- Exemple – Masques de **colonne** : création d'un **masque**

```
CREATE MASK pbol39.mask_secu ON pbol39.patients FOR COLUMN numsecu RETURN
 CASE
   WHEN VERIFY_GROUP_FOR_USER(SESSION_USER, 'PATIENTS') = 1 THEN numsecu
   ELSE '*******' CONCAT RIGHT(numsecu, 6)
 END
 ENABLE;

 ALTER TABLE pbol39.patients ACTIVATE COLUMN ACCESS CONTROL;
```

4. RCAC

- Exemple – Masques de **colonne** : création d'un **masque**
  - Connexion sous le profil MIKE (rattaché au profil de groupe PATIENTS)

```
  Affichage d'un travail

Utilisateur:    MIKE
```

```
CREATE MASK pbol39.mask_secu ON pbol39.patients FOR COLUMN numsecu RETURN
 CASE
  WHEN VERIFY_GROUP_FOR_USER(SESSION_USER, 'PATIENTS') = 1 THEN numsecu
  ELSE '*******' CONCAT RIGHT(numsecu, 6)
 END
```

```
> select * from pbol39.patients
```

Un patient peut visualiser son numéro de sécurité sociale

```
                                  Affichage des données
                                                          Largeur des données . . :    91
Première ligne à afficher . . _____                      Première colonne à afficher . _____
....+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8....+....9.
PROFIL      PRENOM              NUMSECU          TRAITEMENT                MEDECIN
MIKE        Mike                1630256077020    Doliprane                 JAMES
********  Fin de données  ********
```

4. RCAC

- Exemple – Masques de **colonne** : création d'un **masque**
  - Connexion sous le profil LEE (rattaché au profil de groupe MEDECINS)

```
  Affichage d'un travail

Utilisateur:   LEE
```

```
CREATE MASK pbol39.mask_secu ON pbol39.patients FOR COLUMN numsecu RETURN
 CASE
   WHEN VERIFY_GROUP_FOR_USER(SESSION_USER, 'PATIENTS') = 1 THEN numsecu
   ELSE '*******' CONCAT RIGHT(numsecu, 6)
 END
```

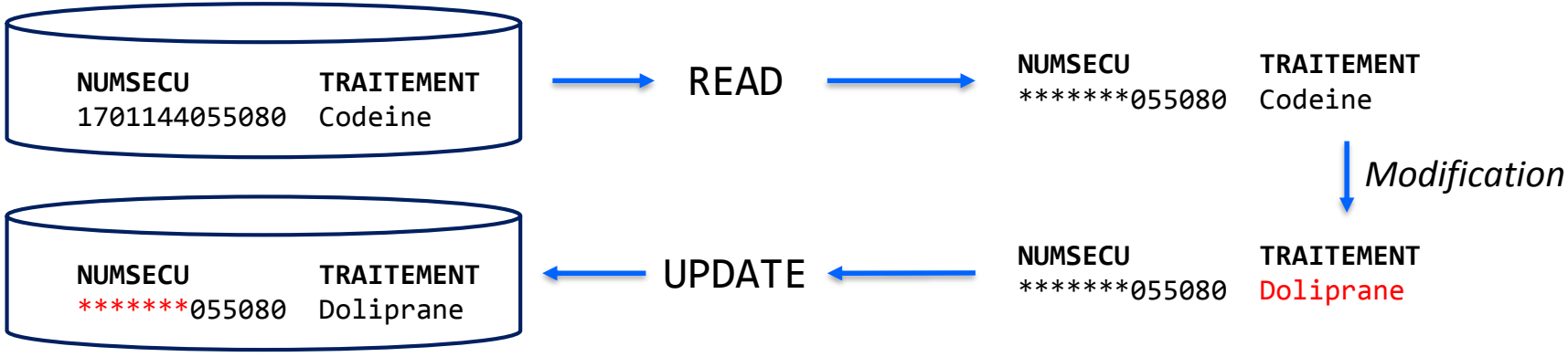
```
> select * from pbol39.patients
```

Une personne autre que le patient lui-même ne peut pas visualiser le numéro de sécurité sociale

```
                                  Affichage des données
                                                                Largeur des données . . :    91
Première ligne à afficher . . _____                  Première colonne à afficher . _____
....+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8....+....9.
PROFIL      PRENOM                NUMSECU          TRAITEMENT                MEDECIN
MAX         Maxime                *******055080    Codeine                   LEE
SAM         Samantha              *******012070    Efferalgan                LEE
AMY         Amy                   *******077080    Voltarene                 LEE
********  Fin de données  ********
```

4. RCAC

- Masques – Attention aux mises à jour accidentelles via les langages RPG et COBOL sur des colonnes en partie masquées

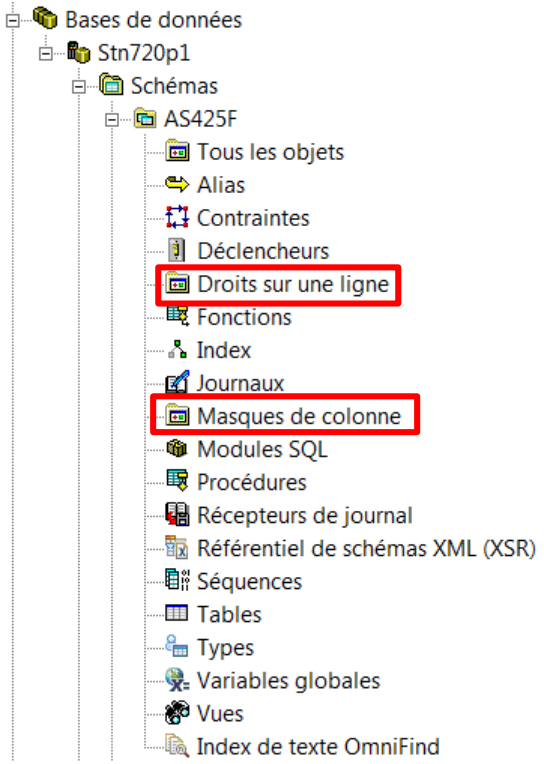


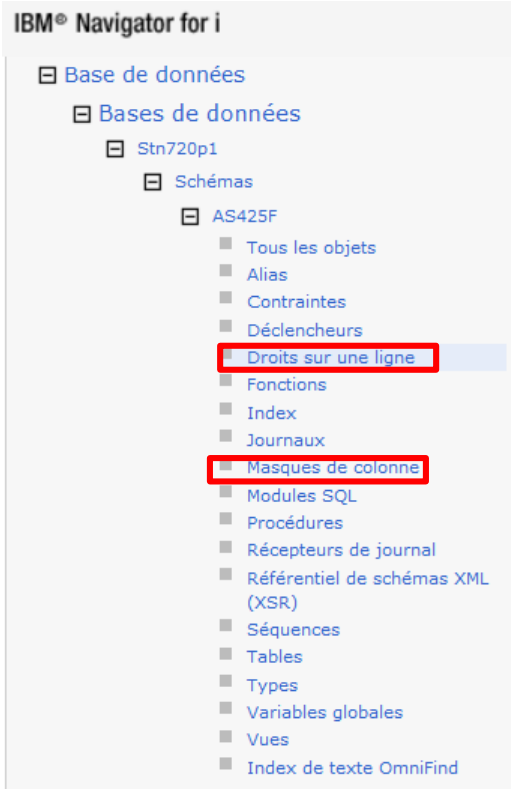
Solutions :
- UPDATE SQL
- Trigger
- Contrainte de vérification avec la clause ON UPDATE VIOLATION :

```
ALTER TABLE … CHECK SUBSTR(numsecu, 1, 6) <> '******'
   ON UPDATE VIOLATION PRESERVE numsecu
```

4. RCAC

- Gestion par System i Navigator et IBM Navigator for i





4. RCAC

- Prérequis
  - Option 47 de SS1 (IBM Advanced Data Security for i)
    - Non facturable

```
5770SS1    47      IBM Advanced Data Security for i
```

- La personne qui met en place la sécurité RCAC doit être "Administrateur de la sécurité base de données"
  - Même pour un profil *ALLOBJ
  - Par la commande WRKFCNUSG, fonction QIBM_DB_SECADM
  - Ces utilisateurs n'auront pas forcément accès aux données des tables

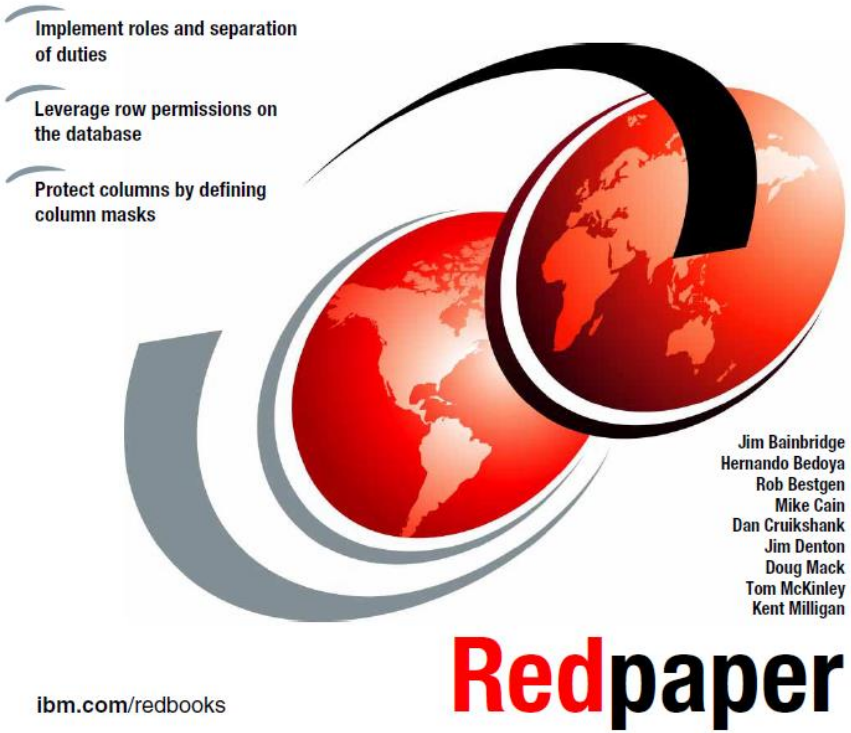
```
                Modifier utilisation fonction (CHGFCNUSG)

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
```

4. RCAC

- Les droits RCAC sont stockés dans la table elle-même
  - Ils sont donc sauvegardés par SAVLIB et SAVOBJ, déplacés par MOVOBJ, dupliqués (par défaut) par CRTDUPOBJ
- Une table (ou un fichier physique) avec des droits RCAC ne peut pas être sauvegardée dans une version d'OS précédente
- Une table (ou un fichier physique) avec des droits RCAC, restaurée sur un système ne possédant pas l'option 47 ne peut plus être ouverte
- Pour voir la liste des droits RCAC existants, interroger les vues du catalogue SYSCONTROLS et SYSCONTROLSDEP de QSYS2

4. RCAC

- Redbook



5. Tables temporelles

- Pour pouvoir répondre à ce type de questions :
  - Quel était le prix de cet article le mois dernier ?
  - Combien de fois a t-il été modifié les 6 derniers mois ?
  - Quels mouvements y-a-t-il eus sur ma table articles au 1er trimestre 2017 ?
  - Quel était l'état de nos comptes avant la fusion ?
  - Pendant combien de temps ce produit a t-il été vendu à ce tarif ?
  - Je voudrais reproduire l'inventaire comme si nous étions le 10 janvier 2017

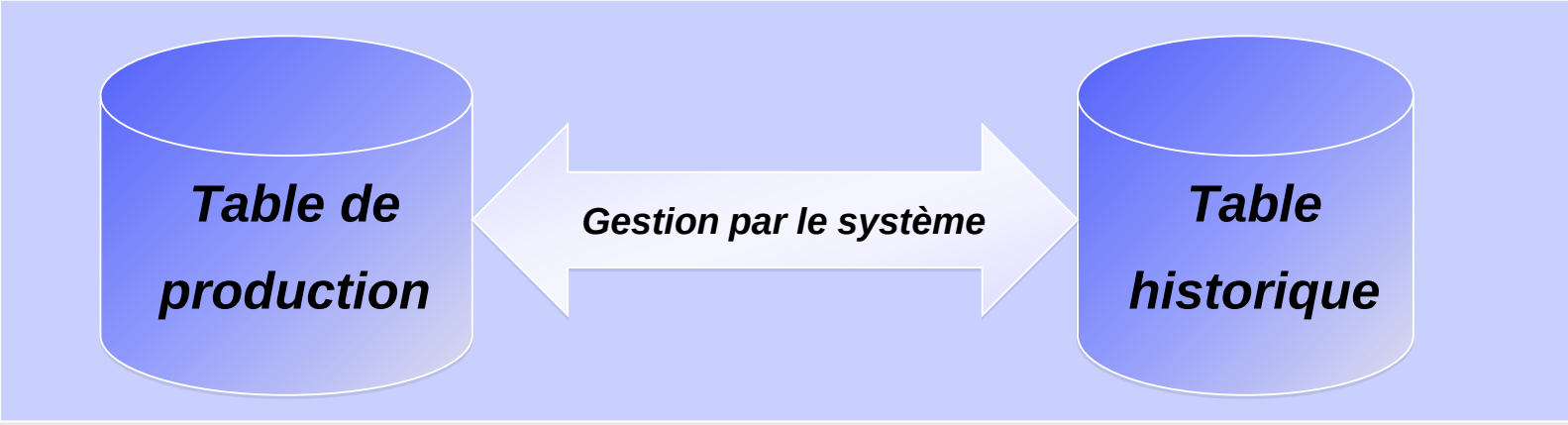
- **Avant** les tables temporelles
  - Etude du journal
  - Historisation manuelle (création d'archives par triggers)

- **Avec** les tables temporelles
  - DB2 garde automatiquement l'historique des données
  - L'interrogation de l'historique se fait :
    - par SQL
    - directement sur les tables de production

5. Table temporelle

- Table DB2 qui contient la version en cours des données et qui historise automatiquement les versions précédentes (données mises à jour et données supprimées) dans une table historique associée
  - Une table temporelle est une table DB2 classique (ou un fichier physique) à laquelle on ajoute 3 colonnes spécifiques
- Possibilité de visualiser, en interrogeant directement la table temporelle :
  - Les données telles qu'elles étaient à n'importe quel point précis du passé
  - Les informations qui ont changé sur une période de temps donnée
  - La date à laquelle les informations ont été modifiées
- Comparaison de données dans le temps, audit…
- Prérequis
  - IBM i **7.3**

5. Tables temporelles



Lecture des données
- SELECT

Modification des données
- INSERT
- UPDATE
- DELETE

Lecture des données
- SELECT

Modification des données
- ~~INSERT~~
- ~~UPDATE~~
- DELETE

5. Tables temporelles

```
ALTER TABLE pbol39.patients4
 ADD COLUMN debut    TIMESTAMP(12) NOT NULL GENERATED AS ROW BEGIN,
 ADD COLUMN fin      TIMESTAMP(12) NOT NULL GENERATED AS ROW END,
 ADD COLUMN periode  TIMESTAMP(12) GENERATED AS TRANSACTION START ID,
 ADD PERIOD SYSTEM_TIME (debut, fin);
```

```
CREATE OR REPLACE TABLE pbol39.pat4histo LIKE pbol39.patients4;

ALTER TABLE pbol39.patients4 ADD VERSIONING USE HISTORY TABLE pbol39.pat4histo;
```

Les données actuelles

```
SELECT * FROM pbol39.patients4;
```

PROFIL	PRENOM	NUMSECU	TRAITEMENT	MEDECIN
MAX	Maxime	1701144055080	Codeine	SMITH
MIKE	Mike	1630256077020	Doliprane	JAMES
SAM	Samantha	2780687012070	Efferalgan	LEE

```
SELECT * FROM pbol39.patients4
 FOR SYSTEM_TIME AS OF '2017-11-11';
```

OU

```
SELECT * FROM pbol39.patients4
 FOR SYSTEM_TIME AS OF CURRENT DATE - 1 DAYS;
```

Les données dans le passé

PROFIL	PRENOM	NUMSECU	TRAITEMENT	MEDECIN
MIKE	Mike	1630256077020	Doliprane	JAMES
SAM	Samantha	2780687012070	Efferalgan	LEE
MAX	Maxime	1701144055080	Codeine	LEE
DOUG	Doug	1590736066060	Dafalgan	JAMES
AMY	Amy	2880534077080	Voltarene	LEE

5. Tables temporelles

```
ALTER TABLE pbol39.patients4
 ADD COLUMN utilisateur VARCHAR(128) GENERATED AS (SESSION_USER),
 ADD COLUMN operation CHAR (1)       GENERATED AS (DATA CHANGE OPERATION),
 ADD COLUMN debut     TIMESTAMP(12) NOT NULL GENERATED AS ROW BEGIN,
 ADD COLUMN fin       TIMESTAMP(12) NOT NULL GENERATED AS ROW END,
 ADD COLUMN periode TIMESTAMP(12) GENERATED AS TRANSACTION START ID,
 ADD PERIOD SYSTEM_TIME (debut, fin);
```

```
SELECT * FROM pbol39.patients4
 FOR SYSTEM_TIME FROM CURRENT TIMESTAMP - 2 DAYS
                 TO CURRENT TIMESTAMP
 WHERE profil = 'MAX';
```

Les opérations réalisées sur le profil MAX les 2 derniers jours

PROFIL	PRENOM	NUMSECU	TRAITEMENT	MEDECIN	OPERATION	DEBUT	UTILISATEUR
MAX	Maxime	1701144055080	Codeine	SMITH	U	2017-11-12 11:46:08...	BOURGEOIS
MAX	Maxime	1701144055080	Codeine	LEE	I	2017-11-10 20:09:41...	BOURGEOIS

5. Tables temporelles

- Clause **FOR SYSTEM TIME**
  - AS OF xxx
  - FROM xxx TO yyy (bornes non comprises)
  - BETWEEN xxx AND yyy (bornes comprises)
- Autre possibilité
  - SET **CURRENT TEMPORAL SYSTEM_TIME** = xxx;
  - SELECT * FROM t1;
- Une vue peut encapsuler un SELECT avec la clause FOR SYSTEM TIME
  - Mais elle ne sera qu'en lecture seule
- On peut joindre 2 SELECT avec des clauses FOR SYSTEM TIME
- Un DROP TABLE de la table temporelle supprime également la table historique
- Un ALTER TABLE xxx ADD COLUMN sur la table temporelle ajoutera également la colonne sur la table historique

5. Tables temporelles

- Restrictions sur la table **temporelle** :
  - CREATE *OR REPLACE* TABLE impossible une fois le versionning activé
  - UPDATE impossible si le registre CURRENT TEMPORAL SYSTEM_TIME est différent de NULL
- Restrictions sur la table **historique**
  - DROP / ALTER TABLE impossibles
  - INSERT et UPDATE impossibles
  - Contrainte référentielle interdite
- Les 2 tables doivent être
  - Dans la même bibliothèque
  - Journalisées
- Arrêt de la temporalité (versionning)
  - ALTER TABLE xxx DROP VERSIONNING

5. Tables temporelles

- Vues du catalogue système
  - Attribut TEMPORAL_TYPE dans SYSTABLES

```
222 SELECT table_name, temporal_type FROM qsys2.systables WHERE table_schema = 'PBOL39' ORDER BY table_name;
223
```

TABLE_NAME	TEMPORAL_TYPE
PATIENTS2	N
PATIENTS2B	N
PATIENTS3	N
PATIENTS4	S
PAT4HISTO	H

  - Vues SYSHISTORYTABLES et SYSPERIODS

- DSPFD, DSPFFD et DSPDBR n'indiquent pas la temporalité


MERCI