

Indicateur NULL et base de données

Introduction

Ce document a pour but de décrire l'utilisation des valeurs **NULL** en base de données.

 Seule une **table** d'une base de données **SQL** peut contenir des colonnes avec une valeur **NULL**.

La présentation ci-dessous concerne les programmes **Visual Adelia** et **Web Adelia**.

Si vous devez accéder à une base de données qui utilise des valeurs **NULL** dans ses colonnes, vous devez alors mettre en place la gestion correspondante.

Cependant, nous décrivons un [cas particulier valable pour des fonctions sur les colonnes en fin de page que nous vous invitons à lire](#).

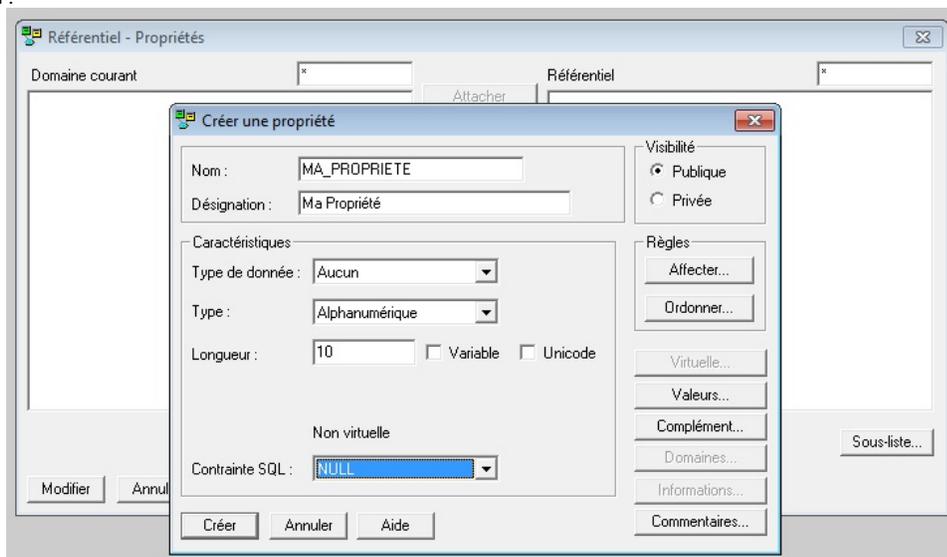
Pour cela, vous avez plusieurs aspects à prendre en compte :

- **Décrire** dans votre **modèle de données** Adelia les colonnes qui autorisent la valeur **NULL** (Propriété conceptuelle / contrainte SQL = [NULL ; NULL WITH DEFAULT] ou bien Propriété logique / onglet complément / contrainte SQL = [NULL ; NULL WITH DEFAULT])
- **Vérifier** que la colonne contient ou non une valeur **NULL** après une lecture
- **Positionner** ou non une valeur **NULL** dans la colonne avant de l'écriture
- **Comparer** la valeur d'une colonne à **NULL** dans la condition d'un ordre SQL.

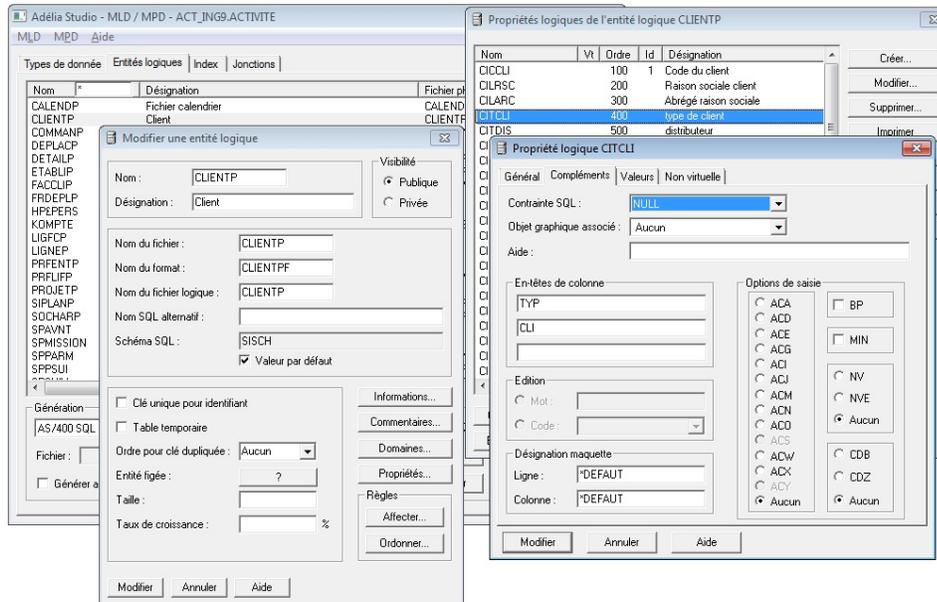
Décrire les données

Exemples de saisie de la contrainte SQL :

- Niveau conceptuel :



- Niveau logique :



Vérifier la valeur de la colonne

Pour cela, on dispose de la fonction prédéfinie **&RECUPERER_INDIC** qui s'utilise comme suit :

```

W_NUM_CLIENT = 123
Lire_Sql Client *Cond(CL_NUM_CLIENT = :W_NUM_CLIENT)
Si *SqlCode = *Normal

/* ___ La colonne nom du client peut contenir une valeur nulle.
/* ___ Pour le vérifier, utiliser la fonction avec comme paramètres :
/* ___ - le nom de l'entité Adelia
/* ___ - le nom de la propriété logique

Si &RECUPERER_INDIC(CLIENT;CL_NOM_CLIENT) = 0

/* ___ La colonne CL_NOM_CLIENT n'a pas la valeur NULL
Lire_Sql PRESIDENT *Cond(PR_NOM = :CL_NOM_CLIENT)
...
Sinon
/* ___ La colonne contient une valeur NULL
...
Fin
Sinon
...
Fin

```

Positionner une valeur NULL dans une colonne

Afin de définir si la valeur d'une colonne contient NULL ou pas, il faut commencer par utiliser l'ordre L4G **AFFECTER_INDIC**. Lorsque le premier paramètre contient ***VRAI**, la valeur qui sera attribuée à la propriété sera positionnée à **NULL** lors de l'écriture (création ou mise à jour). Dans le cas contraire, elle conservera la valeur contenue dans la variable. Dans l'utilisation des ordres SQL de création ou de mise à jour, il est obligatoire d'utiliser le mot réservé ***INDIC_NULL** en paramètre. Le fait d'utiliser une valeur nulle dépend uniquement de ce qui doit être mise en oeuvre au niveau du métier et permet d'indiquer que la donnée n'est pas renseignée.



Pour une simplification du mode de gestion, vous pourriez mettre en place des règles de gestion qui simplifieraient la gestion des valeurs **NULL** en utilisant systématiquement une zone de donnée en relation avec les colonnes.

Ci après, deux exemples de mise en oeuvre :

Exemple 1, sans règle de gestion

```

Si W_NBR_ENFANT = 0
  Affecter_Indic *Vrai CLIENT CL_NOM_CLIENT
Sinon
  Affecter_Indic *Faux CLIENT CL_NOM_CLIENT
Fin
...
Creer_Sql CLIENT *Indic_Null

```

Exemple 2, avec une règle de gestion (génération avec RG implicite)

Il faut créer deux **règles de gestion** :

- une RG de classe **Création**
- une RG de classe **Mise à jour**.

Contenu de la règle de gestion implicite :

```

Decl BOOL W_BOO_CL_NOM_CLIENT /* --- zone de donnée pour la propriété CL_NOM_CLIENT
Affecter_Indic W_BOO_CL_NOM_CLIENT CLIENT CL_NOM_CLIENT

```

Traitement dans le programme :

```

W_BOO_CL_NOM_CLIENT = (W_NBR_ENFANT = 0)
...
Creer_Sql CLIENT *Indic_Null

```

Comparer à NULL la valeur d'une colonne dans une requête SQL

Afin de tester si une colonne vaut **NULL** dans une requête Sql, on utilisera la **fonction scalaire** SQL **EST_INDIC_NULL** directement dans la clause conditionnelle :

```

W_NUM_CLIENT = 123
Lire_Sql CLIENT *Cond( EST_INDIC_NULL(CL_NOM_CLIENT) Et CL_NUM_CLIENT = :W_NUM_CLIENT )
Si *SqlCode = *Normal
  /* ___ Le client existe et le nom n'est pas renseigné ___
  ...
Fin

```

Cas particulier des fonctions sur les colonnes

Certaines fonctions telles que MAX ou bien MIN peuvent retourner une valeur **NULL** alors que la colonne elle-même n'en gère pas. Ce cas se produit à partir du moment où la sélection ne retourne pas d'enregistrement.

Pour éviter une erreur SQL (et un éventuel enregistrement dans la log du travail iSeries), nous recommandons d'utiliser une **variable indicatrice** dans l'ordre SQL.

Par exemple :

```

Lire_Sql *Col(MAX(CL_MNT_CREDIT) :W_MNT_CREDIT_MAX) *Cond(CL_CAT_CLIENT = 'A')

```

Dans le cas où il n'y a pas de catégorie de client égal à 'A' alors une erreur SQL est renvoyée.

Pour éviter cela, il faut écrire :

Dans le paragraphe de **déclaration**

```

Ref(CL_MNT_CREDIT) W_MNT_CREDIT_MAX
Num_Bin_2          W_IND_MAX

```

Dans un paragraphe de **traitement**

```
Lire_Sql *Col(MAX(CL_MNT_CREDIT) :W_MNT_CREDIT_MAX W_IND_MAX) *Cond(CL_CAT_CLIENT = 'A')
Si W_IND_MAX < 0
  /* ____ La valeur renvoyée vaut NULL ____
Sinon
  /* ____ La valeur de W_MNT_CREDIT_MAX est renseignée avec la valeur max des enregistrements liés à la
condition ____
Fin
```