La PROC SQL « pour les nuls »

Présenté par Louis-René Rheault
Pour le Club des Utilisateurs SAS de Québec

7 juin 2022

Club des utilisateurs SAS de Québec

UNIVERSITÉ LAVAL

Objectifs de la présentation

Au terme de cette présentation, vous serez en mesure d’ :

- ☺ Extraire et transformer des données à l’aide de requêtes SQL simples ou en utilisant l’assistant d’Enterprise Guide
- ☺ Intégrer la procédure SQL au sein de vos projets pour en simplifier certaines étapes

Plan de la présentation

1. Usages de la procédure SQL
2. Données utilisées
3. Structure de l'énoncé SELECT
4. Calculs et création de variables
5. DISTINCT… pour éliminer les doublons
6. Regroupements et balayage vertical
7. Requêtes sur plusieurs tables
8. Assistant dans Enterprise Guide
9. Conclusion

1. Usages de la procédure SQL

- SQL (Structured Query Language) est un langage de gestion et d'interrogation de données
- Implanté de manière très semblable dans plusieurs logiciels, notamment la plupart des systèmes de gestion de bases de données (Sybase, Oracle, DB2, Informix, SQL Server, MS-Access, MySQL, Teradata, Netezza, …).
- SAS a implanté le langage SQL sous la forme d'une procédure: **PROC SQL**
- Ne <u>remplace pas</u> l'étape **DATA**, mais plutôt complémentaire à celle-ci, et un atout <u>essentiel</u> pour progresser en SAS...

1. Usages de la procédure SQL (suite)

- Principales tâches réalisées via la procédure SQL de SAS

Énoncé	Usage
SELECT	Requête d'extraction avec filtres, fusion et contrôle des attributs de colonnes
CREATE	Créer une table, une vue ou un index
ALTER TABLE	Modifier les attributs d'une table ou des variables
INSERT	Ajouter des rangées dans une table
DELETE	Supprimer des rangées d'une table
DROP	Supprimer des tables, des vues ou des index

1. Usages de la procédure SQL (suite)

- La procédure SQL diffère de la plupart des autres procédures:
  - pas de paramètre DATA= pour nommer la table lue
  - plusieurs requêtes dans la même procédure
  - requête faite de plusieurs "clauses", qui figurent dans un ordre précis
  - dans les listes de variables ou d'éléments, séparer ceux-ci par des virgules
  - utiliser le symbole * pour référer à tous les éléments d'une table
  - pas d'énoncé **WEIGHT** disponible, car ne peut pondérer les observations
  - pas d'énoncé **RUN** pour terminer - utiliser plutôt **QUIT**
- Ces différences ne marquent pas une inconsistance chez SAS mais sont dictées par les standards du langage SQL

2. Données utilisées

- Base de données d'un services d'activités sportives comportant:
  - Table des cours disponibles à chaque session
  - Table des personnes
  - Table des sessions
  - Table des inscriptions

2. Données utilisées

INSCRIPTIONS

Where | Query Builder | Tasks | Share

	Cours	Session	Personne
1	Hockey	2022E	John
2	Hockey	2022E	Robert
3	Hockey	2022E	Alice
4	Hockey	2022E	Henry
5	Hockey	2022E	James
6	Hockey	2022E	Judy
7	Hockey	2022E	Louise
8	Hockey	2022E	Thomas
9	Hockey	2022E	William
10	Hockey	2022E	Janet
11	Hockey	2022E	Jeffrey

SESSION

Where | Query Builder | Tasks | Share

	Session	DateDébut	DateFin
1	2021E	28JUN2021	27AUG2021
2	2021A	06SEP2021	17DEC2021
3	2022H	17JAN2022	10JUN2022
4	2022E	27JUN2022	26AUG2022
5	2022A	05SEP2022	16DEC2022
6	2023H	16JAN2023	09JUN2023

PERSONNES

Where | Query Builder | Tasks

	Nom	Sexe	Age	Grandeur	Poids
1	Alfred	M	14	69	112.5
2	Alice	F	13	56.5	84
3	Barbara	F	13	65.3	98
4	Carol	F	14	62.8	100.9
5	Henry	M	14	63.5	104.8
6	James	M	12	57.3	83
7	Jane	F	12	59.8	84.5
8	Janet	F	15	62.5	114
9	Jeffrey	M	13	61.4	84
10	John	M	12	59	99.5
11	Joyce	F	11	51.3	50.5
12	Judy	F	14	64.3	90
13	Louise	F	12	56.3	77
14	Mary	F	15	67	113
15	Philip	M	16	72	150
16	Robert	M	12	64.8	128
17	Ronald	M	15	67	133
18	Thomas	M	11	57.5	85
19	William	M	15	66.5	112

COURSSESSIONS

Where | Query Builder | Tasks | Share

	Cours	Session	Animation	Jour	Heure
1	Hockey	2022E	Patrice	Lundi	17:00
2	Badminton	2022E	Hans	Mardi	18:00
3	Rugby	2022E	Louis-René	Mercredi	17:30
4	Escalade	2022E	Alain	Jeudi	12:00
5	Pilates	2022E	Christine	Jeudi	19:00

3. Structure de l'énoncé SELECT

L'énoncé SELECT sert à extraire des informations (listes ou agrégations) d'une ou plusieurs tables.

- Clauses de l'énoncé **SELECT** fournies dans un ordre précis:

  SELECT *variable-a, variable-b, …*

  FROM *table*

  WHERE *filtre-des-rangées*

  GROUP BY *variable(s)-d'agrégation-des-rangées*

  HAVING *filtre-pour-variable(s)-agrégées-dans-les-rangées*

  ORDER BY *variable(s)-de-tri*

3. Structure de l’énoncé SELECT (suite)

- Dans sa forme la plus simple, **SELECT** :
  - extrait certaines ou toutes les variables d'une table,
  - les ordonne horizontalement, en leur accolant des attributs et
  - produit une nouvelle table ou un rapport <u>sans contrôle de mise en page</u>

3. Structure de l'énoncé SELECT (suite)

Exemple: extraire toutes les variables de tous les enregistrements d'une table:

```
PROC SQL;
    SELECT *
    FROM CSAS2.PERSONNES;
QUIT;
```

Nom	Sexe	Age	Grandeur	Poids
Alfred	M	14	69	112.5
Alice	F	13	56.5	84
Barbara	F	13	65.3	98
Carol	F	14	62.8	100.9
Henry	M	14	63.5	104.8
James	M	12	57.3	83
Jane	F	12	59.8	84.5
Janet	F	15	62.5	114
Jeffrey	M	13	61.4	84
John	M	12	59	99.5
Joyce	F	11	51.3	50.5
Judy	F	14	64.3	90
Louise	F	12	56.3	77
Mary	F	15	67	113
Philip	M	16	72	150
Robert	M	12	64.8	128
Ronald	M	15	67	133
Thomas	M	11	57.5	85
William	M	15	66.5	112

3. Structure de l’énoncé SELECT (suite)

- Pas de table créée par défaut
- Un rapport s'affiche, sans véritable contrôle de mise en page
- Une erreur de syntaxe ne provoque pas l'arrêt de la procédure
- Dans SAS Base, si on ne spécifie pas la commande de fin « QUIT; », la procédure demeure ouverte, sans requérir de soumettre de nouveau `PROC SQL;`
  (Dans SAS Enterprise Guide, des commandes de fins sont automatiquement ajoutées à toutes les tâches exécutées)

3. Structure de l'énoncé SELECT (suite)

- Exemple - extraire certaines variables et rangées:

```
PROC SQL;
    SELECT NOM, SEXE, AGE
    FROM CSAS2.PERSONNES
    WHERE Age >= 14;
QUIT;
```

Nom	Sexe	Age
Alfred	M	14
Carol	F	14
Henry	M	14
Janet	F	15
Judy	F	14
Mary	F	15
Philip	M	16
Ronald	M	15
William	M	15

3. Structure de l’énoncé SELECT (suite)

- La clause CREATE peut précéder le SELECT et permet de créer une table, une vue ou un index.
- Pas de rapport affiché en sortie, mais une note au journal et une nouvelle table disponible pour de nouveaux traitements.

3. Structure de l'énoncé SELECT (suite)

- Exemple – créer une nouvelle table basée sur le résultat d'une requête:

```
CREATE TABLE work.publicite_sauveteurs AS
SELECT NOM, SEXE, AGE
FROM CSAS2.PERSONNES
WHERE Age >= 14;
```

```
10
11    PROC SQL;
12        CREATE TABLE work.publicite_sauveteurs AS
13        SELECT NOM, SEXE, AGE
14        FROM CSAS2.PERSONNES
15        WHERE Age >= 14;
NOTE: Table WORK.PUBLICITE_SAUVETEURS created, with 9 rows and 3 columns.

16    QUIT;
NOTE: PROCEDURE SQL used (Total process time):
      real time           0.02 seconds
      cpu time            0.01 seconds
```

3. Structure de l'énoncé SELECT (suite)

- Clause **ORDER BY** permet de trier la table selon une ou plusieurs des variables présentes dans la table en entrée:

```
CREATE TABLE work.pub_sauveteurs_triee AS
SELECT NOM, SEXE, AGE
FROM CSAS2.PERSONNES
WHERE Age >= 14
ORDER BY Sexe, Age DESC;
```

	Nom	Sexe	Age
1	Mary	F	15
2	Janet	F	15
3	Carol	F	14
4	Judy	F	14
5	Philip	M	16
6	William	M	15
7	Ronald	M	15
8	Alfred	M	14
9	Henry	M	14

- Paramètre DESCENDING <u>après</u> la variable visée

- Plupart des autres fonctionnalités de la procédure **SORT** r
disponibles (NODUPKEY, DUPOUT=, SORTSEQ=…)

4. Calculs et création de variables

- Une des rares procédures capable de créer des variables
- Mot-clé **AS** permet de nommer la nouvelle variable (ou renommer une variable existante)
- Autres attributs de variables peuvent être définis via les mots-clés **FORMAT=**, **LENGTH=**, **LABEL=**, etc.
- Les variables utilisées dans un calcul ou une expression n'ont pas besoin d'être présentes dans les variables du SELECT

4. Calculs et création de variables (suite)

- En présence de données manquantes, le résultat d'une expression est manquant dans les mêmes cas que dans l'étape DATA
- La plupart des fonctions de DATA sont disponibles et opèrent à peu près de la même manière (sauf pour les listes de variables)
- Contrairement au « KEEP » de l'étape DATA, l'ordre des variables dans le SELECT déterminera l'ordre des variables dans la sortie
- Mot-clé CALCULATED dans la requête précédente permet de faire référence à une variable calculée dans la même requête, dans un filtre du WHERE ou dans une variable de tri du ORDER BY

4. Calculs et création de variables (suite)

- Exemple 1 - créer la variable *Ratio_poids_grandeur*

```
SELECT NOM,SEXE,AGE,
   POIDS/GRANDEUR AS Ratio_poids_grandeur
                     FORMAT=6.3
FROM CSAS2.PERSONNES
ORDER BY AGE,
   CALCULATED Ratio_poids_grandeur DESC;
```

- Le résultat est trié par l'âge et ensuite par le ratio en ordre décroissant.

Nom	Sexe	Age	Ratio_poids_grandeur
Thomas	M	11	1.478
Joyce	F	11	0.984
Robert	M	12	1.975
John	M	12	1.686
James	M	12	1.449
Jane	F	12	1.413
Louise	F	12	1.368
Barbara	F	13	1.501
Alice	F	13	1.487
Jeffrey	M	13	1.368
Henry	M	14	1.650
Alfred	M	14	1.630
Carol	F	14	1.607
Judy	F	14	1.400
Ronald	M	15	1.985
Janet	F	15	1.824
Mary	F	15	1.687

4. Calculs et création de variables (suite)

- Exemple 2 - créer les variables *cours_session*

```
SELECT UPCASE(CATX("-", SUBSTR(Cours,1,4), Session)) AS cours_session1,
       *,
       LOWCASE(CATX("-", Session, SUBSTR(Cours,1,4))) AS cours_session2
FROM CSAS2.courssessions;
```

4. Calculs et création de variables (suite)

- Exemple 2 - créer les variables *cours_session*

cours_session1	Cours	Session	Animation	Jour	Heure	cours_session2
HOCK-2022E	Hockey	2022E	Patrice	Lundi	17:00	2022e-hock
BADM-2022E	Badminton	2022E	Hans	Mardi	18:00	2022e-badm
RUGB-2022E	Rugby	2022E	Louis-René	Mercredi	17:30	2022e-rugb
ESCA-2022E	Escalade	2022E	Alain	Jeudi	12:00	2022e-esca
PILA-2022E	Pilates	2022E	Christine	Jeudi	19:00	2022e-pila
HOCK-2022H	Hockey	2022H	Patrice	Samedi	10:00	2022h-hock
BADM-2022H	Badminton	2022H	Jean	Mardi	18:00	2022h-badm
BADM-2022H	Badminton	2022H	Alhassana	Vendredi	19:30	2022h-badm
ESCA-2022H	Escalade	2022H	Hans	Dimanche	14:00	2022h-esca

5. DISTINCT… pour éliminer les doublons

- Insérer le paramètre « DISTINCT » juste après le mot-clé SELECT permet d'éliminer toutes les lignes qui sont complètement dupliquées (toutes les données de toutes les colonnes sont identiques)

- Exemple: pour obtenir la liste de tous les animateurs à une session:

```
SELECT DISTINCT animation
FROM csas2.COURSSESSIONS
WHERE Session EQ '2022H';

SELECT DISTINCT Cours
FROM csas2.COURSSESSIONS
WHERE Session EQ '2022H';
```

Animation
Alhassana
Hans
Jean
Patrice

Cours
Badminton
Escalade
Hockey

Même si un animateur donne plusieurs cours, il n'apparaîtra qu'une seule fois.
Même si un cours se donne plusieurs fois, il n'apparaîtra qu'une seule fois.

6. Regroupements et balayage vertical

- Permet de faire des calculs portant sur plusieurs rangées de données: sommes, moyennes, min, max, compter, …
- Il faut énumérer dans le SELECT les variables qui forment les sous-groupes
- Il faut spécifier la fonction à calculer et spécifier sur quelle variable doit se faire les regroupements / les calculs.
- Il faut répéter dans une nouvelle instruction GROUP BY les variables qui forment les sous-groupes

```
SELECT sexe, MIN(age) AS Age_minimal,
             MAX(age) AS Age_maximal,
             MEAN(age) AS Moyenne_age
FROM csas2.personnes
GROUP BY sexe;
```

Sexe	Age_minimal	Age_maximal	Moyenne_age
F	11	15	13.22222
M	11	16	13.4

6. Regroupements et balayage vertical

- Pour les variables de regroupements, on peut encore faire référence à un champ calculé à l'aide du mot-clé « CALCULATED » suivi du nom de la variable créée
  - pour appliquer un tri dans le ORDER BY
    (comme pour les variables calculées vues précédemment)
  - pour appliquer un filtre dans une nouvelle section **HAVING** immédiatement après la section GROUP BY

6. Regroupements et balayage vertical

- Pour les variables de regroupements, on peut encore faire référence à un champ calculé à l'aide du mot-clé « CALCULATED » suivi du nom de la variable créée
  - Par exemple: Trouver quels groupes de l'été 2022 n'ayant personne âgé de plus de 15 ans – pour les assurances…

```
SELECT crs.Cours, crs.Session, crs.Animation,
   MAX(pers.Age) AS Age_maximal
FROM csas2.courssessions crs
   INNER JOIN csas2.inscriptions inscr
      ON crs.Cours EQ inscr.Cours
         AND crs.Session EQ inscr.Session
   INNER JOIN csas2.personnes pers
      ON inscr.Personne EQ pers.Nom
WHERE crs.Session EQ '2022E'
GROUP BY crs.Cours, crs.Session, crs.Animation
HAVING Calculated Age_maximal < 16;
```

6. Regroupements et balayage vertical

- Par exemple: Trouver quels groupes de l'été 2022 n'ayant personne âgé de plus de 15 ans – pour les assurances…

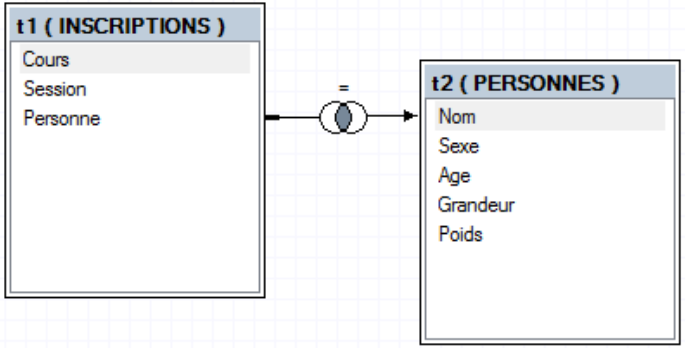
Cours	Session	Animation	Age_maximal
~~Badminton~~	~~2022E~~	~~Hans~~	~~16~~
~~Escalade~~	~~2022E~~	~~Alain~~	~~16~~
Hockey	2022E	Patrice	15
Pilates	2022E	Christine	15
Rugby	2022E	Louis-René	15

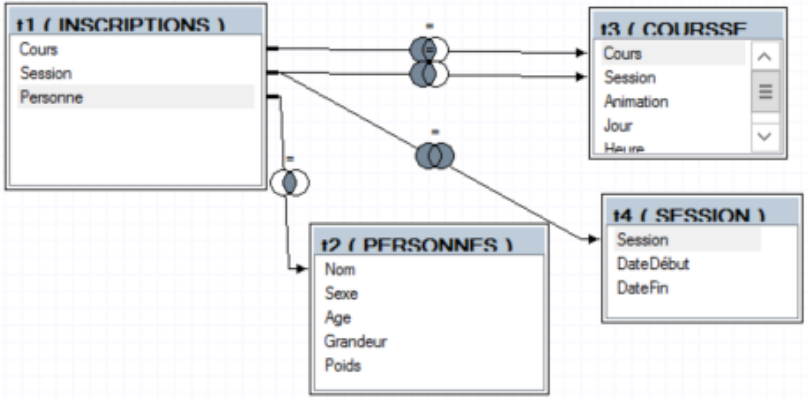
Cours	Session	Animation	Age_maximal
Hockey	2022E	Patrice	15
Pilates	2022E	Christine	15
Rugby	2022E	Louis-René	15

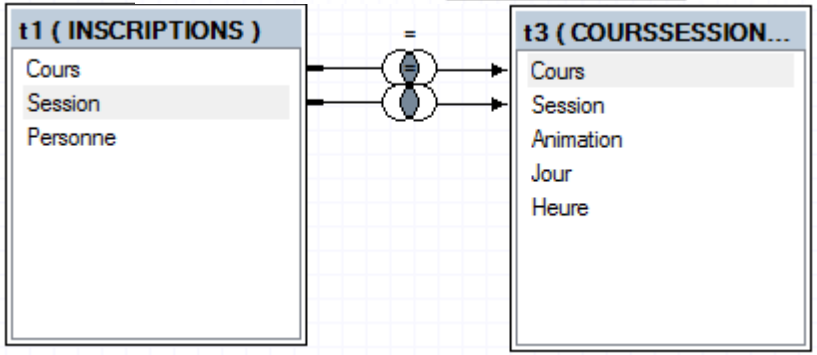
7. Requêtes sur plusieurs tables

- La plupart du temps, l'information se trouve dans différentes tables. (ressemble à l'utilisation de l'énoncé MERGE dans une étape DATA)
- On doit alors inclure toutes les tables souhaitées dans l'instruction FROM
- On doit aussi spécifier quel est le lien entre chacune des tables à l'aide des instructions de jointures (JOIN)
- On n'a pas besoin de trier les tables (contrairement au MERGE de l'étape DATA)
  → *mais peut influencer la performance / le temps de traitement*

7. Requêtes sur plusieurs tables







7. Requêtes sur plusieurs tables

- On doit aussi spécifier quel est le lien entre chacune des tables à l'aide des instructions de jointures (JOIN).
- Ce lien doit identifier de manière **unique** la ligne de la 2e table est liée à la ligne de la table principale.
- Ces éléments identifiant de façon unique un enregistrement dans une table se nomme **les clés**.
- Parfois, ce sont plusieurs variables qui identifient de façon unique un enregistrement. On parle alors de « clés composées ».

7. Requêtes sur plusieurs tables

- Exemple: Faire la liste des inscriptions pour la feuille de présences d'un cours

```
SELECT inscr.*,
     pers.sexe, pers.age,
     crs.Session, crs.Animation, crs.Jour, crs.Heure
FROM csas2.INSCRIPTIONS inscr
     INNER JOIN csas2.personnes pers
          ON inscr.Personne = pers.Nom
     INNER JOIN csas2.courssessions crs
          ON inscr.Cours = crs.Cours
               AND inscr.Session = crs.Session
WHERE inscr.cours EQ 'Badminton' AND inscr.Session EQ '2022E'
ORDER BY inscr.Personne;
```

7. Requêtes sur plusieurs tables

- Exemple: Faire la liste des inscriptions pour la feuille de présences d'un cours

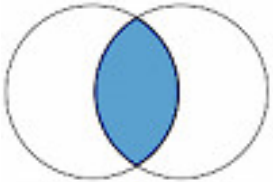
Cours	Session	Personne	Sexe	Age	Session	Animation	Jour	Heure
Badminton	2022E	Alice	F	13	2022E	Hans	Mardi	18:00
Badminton	2022E	Carol	F	14	2022E	Hans	Mardi	18:00
Badminton	2022E	John	M	12	2022E	Hans	Mardi	18:00
Badminton	2022E	Joyce	F	11	2022E	Hans	Mardi	18:00
Badminton	2022E	Louise	F	12	2022E	Hans	Mardi	18:00
Badminton	2022E	Philip	M	16	2022E	Hans	Mardi	18:00
Badminton	2022E	Ronald	M	15	2022E	Hans	Mardi	18:00
Badminton	2022E	Ronald	M	15	2022E	Hans	Mardi	18:00

7. Requêtes sur plusieurs tables

- Il existe plusieurs types de jointures entre 2 tables:

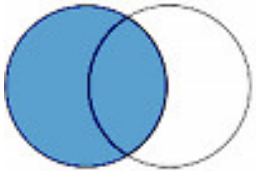
  - Les INNER JOIN (jointures internes) – la requête fournit tous les enregistrements pour lesquels la clé (variables de jointure) existe dans les 2 tables.

    → Si un enregistrement n'a pas de clé, ou s'il n'a pas de clé correspondante dans l'autre table, il ne fera pas partie des résultats.

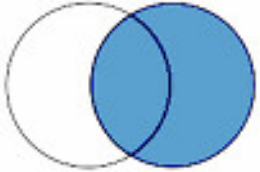


7. Requêtes sur plusieurs tables

- Il existe plusieurs types de jointures:
  - Les LEFT JOIN (jointures à gauche) – la requête fournit **tous les enregistrements de la 1ère table** et complète les informations de la 2e table pour les enregistrements ayant une clé correspondante.



  - Les RIGHT JOIN (jointure à droite) – la requête fournit **tous les enregistrements de la 2e table** et complète les informations de la 1ère table pour les enregistrements ayant une clé correspondante.

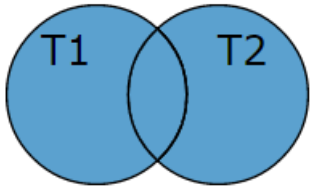


7. Requêtes sur plusieurs tables

- Il existe plusieurs types de jointures:

  - Les FULL JOIN (jointure totale) – la requête fournit **<u>tous les enregistrements de chacune des tables</u>**.

    → Pour les enregistrements ayant une clé correspondante, toutes les colonnes sont alimentées. Pour les enregistrements n'ayant pas de clé correspondante dans l'autre table, les champs de cette autre table sont restent vides.

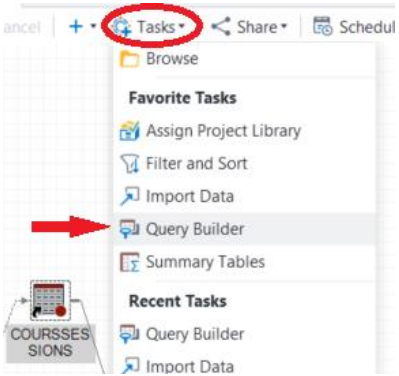


8. Utilisation de l’assistant dans Enterprise Guide

Offre une interface « point and click » pour générer et exécuter des requêtes SQL à l’aide de PROC SQL.

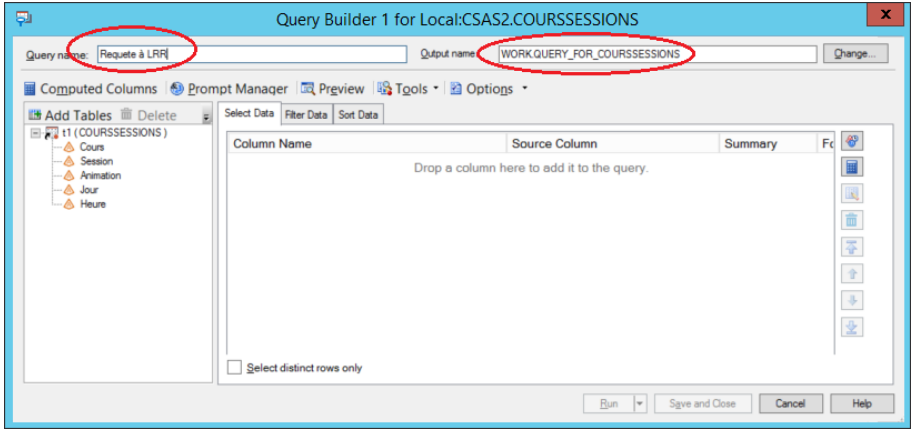
- Dans un flux de processus, on doit « déposer » une table dans l’espace de travail.

- Cliquer sur la table et cliquer sur « Task » → « Query builder »



8. Utilisation de l’assistant dans Enterprise Guide

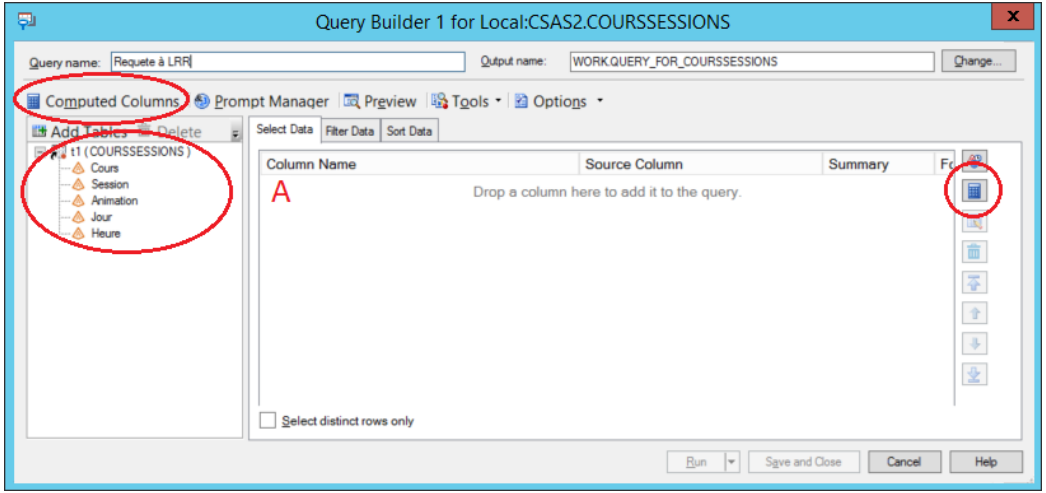
- Une fois dans l’assistant, vous pouvez:



- Changer le nom de la tâche dans le flux de processus
- Changer l’emplacement et le nom de la table (Change)

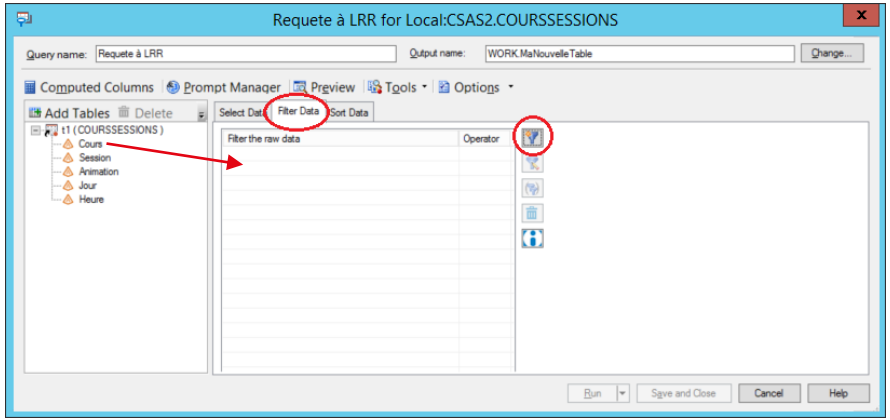
8. Utilisation de l’assistant dans Enterprise Guide

- Une fois dans l’assistant, vous pouvez:
  - Faire glisser des variables dans la zone « Select Data » (A)
  - Cliquer sur « Computed Columns » pour créer des variables calculées
  - Cliquer sur une variable sélectionnée, une fois et une 2e fois pour renommer cette variable



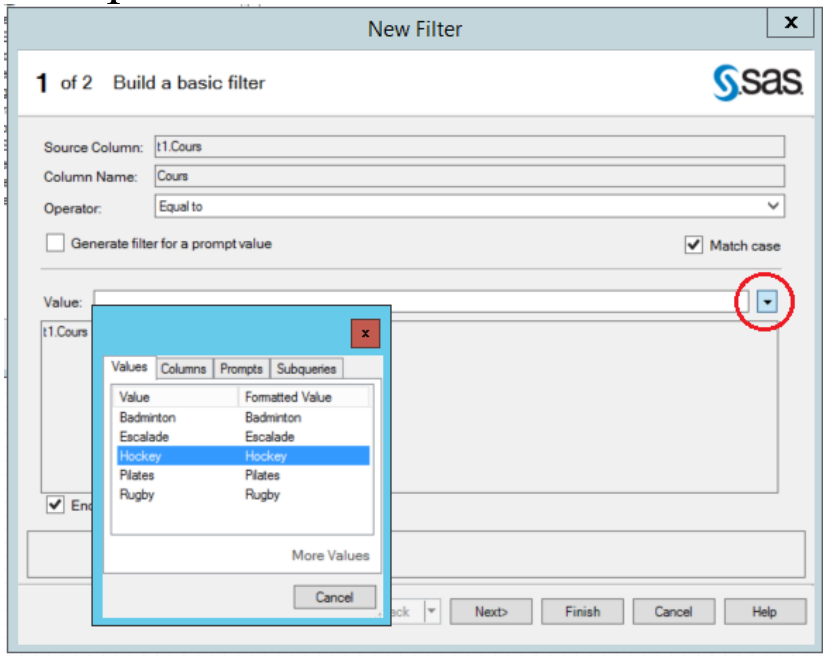
8. Utilisation de l'assistant dans Enterprise Guide

- Une fois dans l'assistant, vous pouvez:
  - Filtrer les données selon un ou plusieurs critères, à l'aide de l'assistant (faire glisser une variable pour y appliquer un filtre)



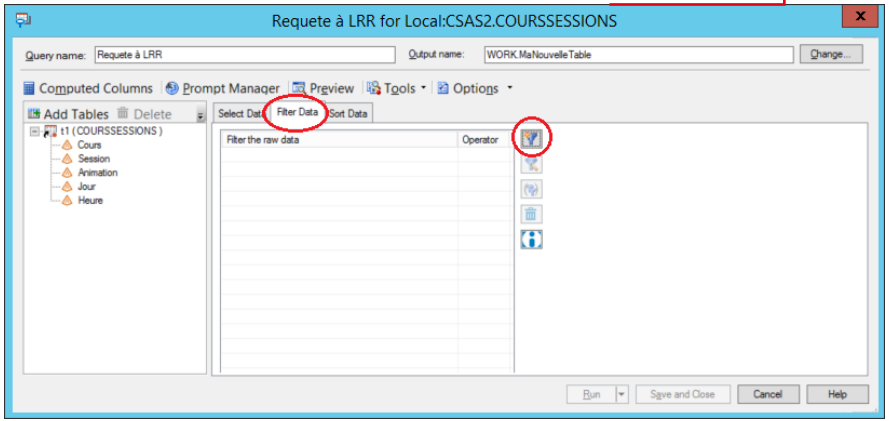
8. Utilisation de l'assistant dans Enterprise Guide

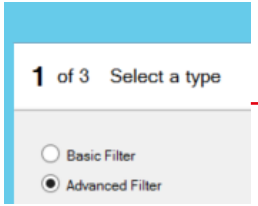
- Une fois dans l'assistant, vous pouvez:
  - Filtrer les données selon un ou plusieurs critères, à l'aide de l'assistant (faire glisser une variable pour y appliquer un filtre)

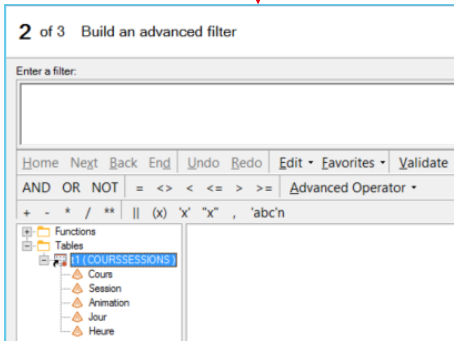


8. Utilisation de l'assistant dans Enterprise Guide

- Une fois dans l'assistant, vous pouvez:
  - Filtrer les données en écrivant une expression de filtre avancé

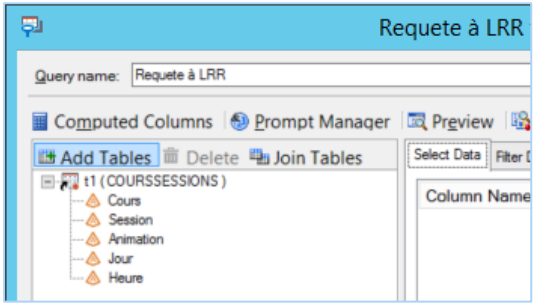


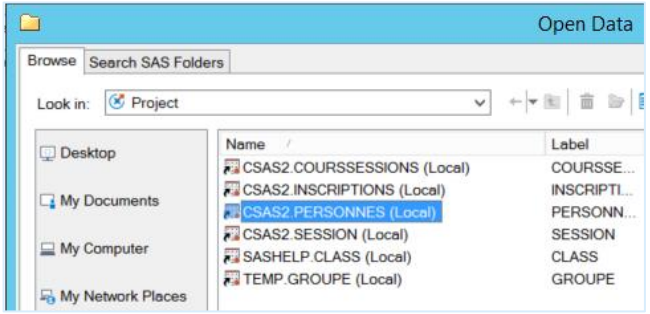




8. Utilisation de l’assistant dans Enterprise Guide

- Une fois dans l’assistant, vous pouvez:
  - Ajouter une ou plusieurs autres tables et en définir les liens

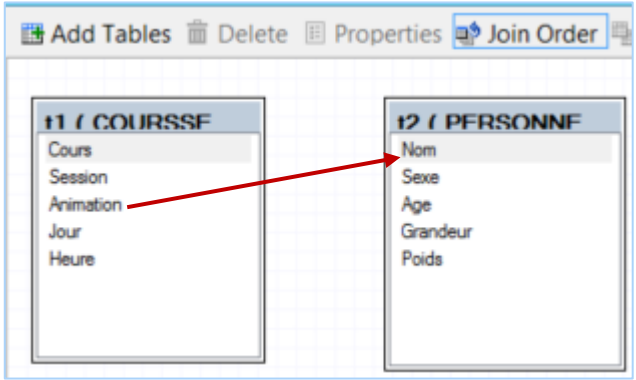


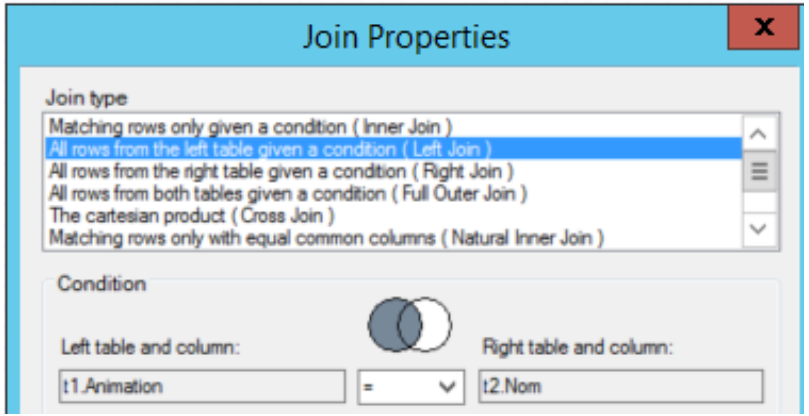


8. Utilisation de l'assistant dans Enterprise Guide

- Une fois dans l'assistant, vous pouvez:
  - Ajouter une ou plusieurs autres tables et en définir les liens

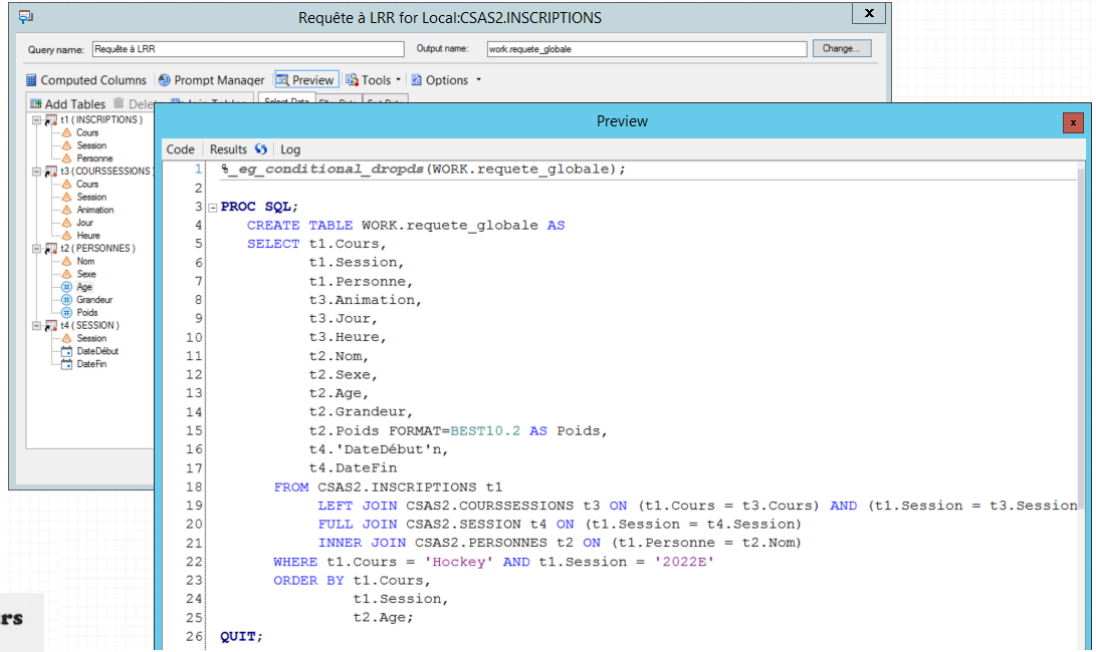
    Faire glisser une variable sur la clé de la 2e table pour les lier et choisir le type de jointure





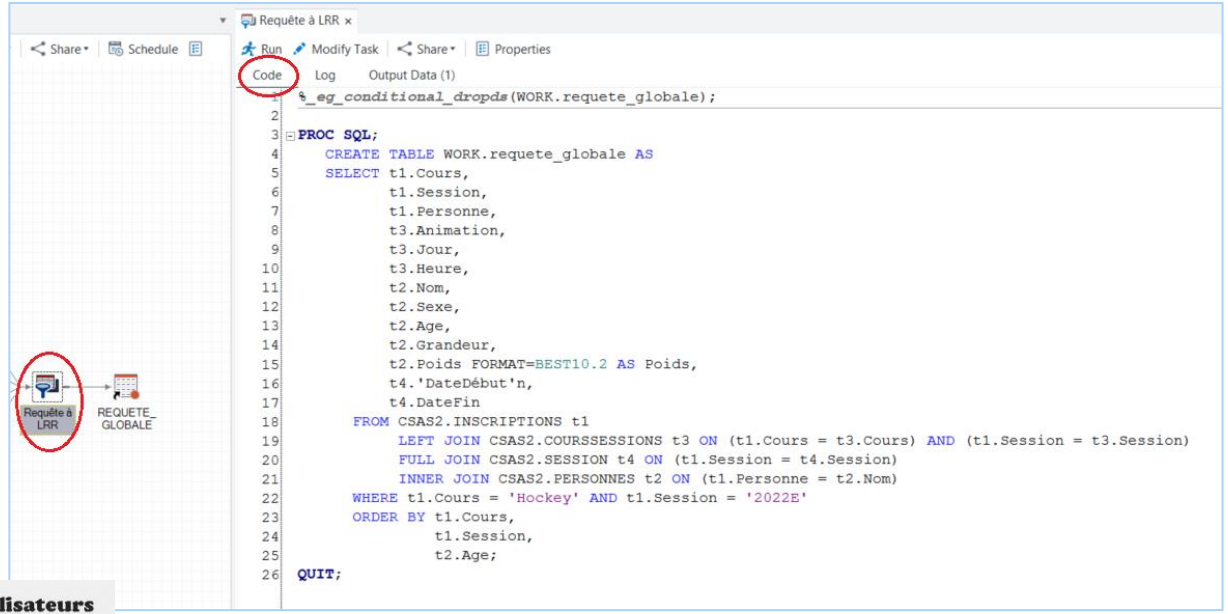
8. Utilisation de l’assistant dans Enterprise Guide

- Pour voir le code généré par l’assistant
  - À l’intérieur de l’assistant, cliquer sur le bouton « Preview »



8. Utilisation de l’assistant dans Enterprise Guide

- Pour voir le code généré par l’assistant
  - À partir du flux de processus, double-cliquer sur la tâche et cliquer sur « **Code** »




```
%_eg_conditional_dropds(WORK.requete_globale);

PROC SQL;
   CREATE TABLE WORK.requete_globale AS
   SELECT t1.Cours,
          t1.Session,
          t1.Personne,
          t3.Animation,
          t3.Jour,
          t3.Heure,
          t2.Nom,
          t2.Sexe,
          t2.Age,
          t2.Grandeur,
          t2.Poids FORMAT=BEST10.2 AS Poids,
          t4.'DateDébut'n,
          t4.DateFin
      FROM CSAS2.INSCRIPTIONS t1
           LEFT JOIN CSAS2.COURSSESSIONS t3 ON (t1.Cours = t3.Cours) AND (t1.Session = t3.Session)
           FULL JOIN CSAS2.SESSION t4 ON (t1.Session = t4.Session)
           INNER JOIN CSAS2.PERSONNES t2 ON (t1.Personne = t2.Nom)
      WHERE t1.Cours = 'Hockey' AND t1.Session = '2022E'
      ORDER BY t1.Cours,
               t1.Session,
               t2.Age;
QUIT;
```

9. Conclusion

- La procédure SQL est un outil puissant et flexible
- Complément essentiel (et non un "remplacement") à l'étape DATA
- Beaucoup de profondeur dans les jointures, la création des variables Macro, la gestion des visées, etc.
- Puissant outil pour s'interfacer avec les bases de données relationnelles (DB2, Oracle, Sybase, Sql Server, MySql, etc.)

Avez-vous des questions

Club des utilisateurs SAS de Québec

UNIVERSITÉ LAVAL

Merci pour votre écoute!

Et un merci spécial à Jean Hardy pour avoir monté la base de cette présentation

Louis-René Rheault – Université Laval
Louis-Rene.Rheault@vrex.ulaval.ca

Club des utilisateurs SAS de Québec

UNIVERSITÉ LAVAL