PROC SQL

Programmation & Logiciels Statistiques

Cours 10

Salim Lardjane - Université de Bretagne-Sud

La proc SQL

- La proc sql met à la disposition de l'utilisateur de SAS le langage SQL de requête et de manipulation de données
- Elle permet de gérer des bases de données sous SAS sans passer par une étape **data** ou une étape proc autre que la proc **sql**
- Les principaux mots-clefs de SQL sont : select, from, on, where, group by, having, order by

La proc SQL

- Select est une instruction qui est requise par toute requête SQL
- Les autres mots-clefs correspondent à des **clauses** de cette instruction
- La seule **clause obligatoire** est from qui spécifie l'origine des données
- Les clauses sont toujours ordonnées dans l'ordre ci-dessus et une clause n'apparaît **au plus qu'une seule fois** dans une instruction select

La proc SQL
sélection de colonnes

```
proc sql;
      select distinct rating from movies;
quit;
```

- Par défaut, les résultats de la requête sont affichés. Pour qu'ils ne le soient pas, utiliser l'option noprint dans proc sql
- distinct est une option qui permet de supprimer les lignes redondantes du résultat obtenu

La proc SQL

Tri des données

```
proc sql;
      select *
      from movies
      order by category;
quit;
```

La proc SQL

Tri des données

- select * signifie qu'on sélectionne **toutes** les colonnes (variables) de la table *movies*
- order by doit toujours être après from
- Les données sont triées selon la variable *category*

La proc SQL

Sélection d'observations

```
proc sql;
      select title, category
      from movies
      where category contains 'Action';
quit;
```

La proc SQL
Sélection d'observations

- On utilise la virgule pour séparer les noms des colonnes sélectionnées
- contains est utilisé uniquement pour les variables caractère
- On aurait pu utiliser également

```
where upcase(category) like '%ACTION%';
```

La proc SQL

Sélection d'observations

```
proc sql;
      select title, category, rating
      from movies
      where category =* 'drana';
quit;
```

La proc SQL

Sélection d'observations

- Le where doit toujours être après from
- L'opérateur =* signifie 'phonétiquement proche'
- Le résultat est qu'on retient les observations dont la catégorie est proche phonétiquement de 'drana'

La proc SQL

Création de variables

```
proc sql;
     select title, rating,
     case rating
          when 'G' then 'general' else
          'other'
     end as level
     from movies;
quit;
```

La proc SQL

Création de variables

- L'ordre des clauses (instructions) est important
- La clause case … end as doit être comprise entre select et from
- Une virgule est requise après le dernier nom de colonne spécifié
- Le when … then … else … est utilisé pour créer une nouvelle variable (colonne)
- Le as sert à l'appeler *level*

La proc SQL

Création de tables

```
proc sql;
      create table action as
      select title, category
      from movies
      where category contains 'Action';
quit;
```

La proc SQL

Création de tables

- Le create table qui précède l'instruction select sert à créer une table SAS appelée *action* dans le répertoir **Work**
- Par défaut, select oriente les résultats de la requête vers la fenêtre d'output. create table permet de les sauvegarder dans une table SAS sans qu'il y ait d'affichage
- create view permet de sauvegarder le résultat de la requête sous le forme d'une **vue** SAS
- La table ou la vue obtenue peut être utilisée dans les instructions SQL ou les étapes SAS subséquentes

La proc SQL

Création de vues

- Une **vue** SAS est un **lien** vers un ensemble de données
- Elle ne contient qu'une requête qui sera exécutée lors de chaque accès à la vue
- Cette requête permet de récupérer l'information voulue à partir d'une ou plusieurs sources de données

La proc SQL

Création de vues

- Une **vue** occupe une **place réduite** en termes de stockage (seule la requête est sauvegardée)
- Elle permet d'avoir accès à des données perpétuellement **à jour** (sans intervention de l'utilisateur)
- L'accès à une **table** est plus **rapide**
- Une table permet de **figer dans le temps** un état des données

La proc SQL

Création de vues

```
proc sql;
      create view g_movies as
      select title, category, rating
      from movies
      where rating = 'G'
      order by title;

      select * from g_movies;
quit;
```

La proc SQL

Création de vues

- Première partie : obtention d'une vue SAS
- Deuxième partie : affichage des sorties
- Le point-virgule permet de distinguer les deux blocs d'instructions dans la proc sql
- Lorsqu'une vue est créée, c'est la requête elle-même qui est sauvegardée en tant que vue SAS, **sans accès aux données**

La proc SQL

Fusion cartésienne

```
proc sql;
      select *
      from dir.customers, movies;
quit;
```

La proc SQL

Fusion cartésienne

- On parlera indifféremment de **fusion** ou de **jointure** (**merge**) de tables ou bases de données
- SQL ne requiert pas le tri préalable des données selon la variable-clef, contrairement à l'instruction merge de l'étape **data**
- La virgule est utilisée pour distinguer les tableaux de données dans la clause from

La proc SQL

Fusion cartésienne

- Sans clause where, **toutes les combinaisons possibles des lignes des deux tables sont obtenues, toutes les colonnes sont retenues**
- Pour un meilleur affichage, sélectionner l'option Create HTML dans le menu Tools/Options/Preferences…/Results

La proc SQL

Fusion interne

```
proc sql;
      select *
      from movies, actors
      where movies.title = actors.title;
quit;
```

La proc SQL

Fusion interne

- La clause where permet de spécifier la variable-clef (title)
- On obtient **toutes les lignes pour lesquelles le nom du film est identique et uniquement ces lignes**
- La variable-clef peut avoir un nom différent dans chacune des tables

La proc SQL

Fusion interne

```
proc sql;
     select m.title, m.rating,
                a.actor_leading
     from movies m, actors a
     where movies.title = actors.title;
quit;
```

La proc SQL

Fusion interne

- Il est possible de définir des abréviations pour les noms de tables
- Celles-ci peuvent être utilisées dans l'instruction select et les clauses where
- Elles doivent être **déclarées** dans la clause from

La proc SQL

Fusion interne (3 tableaux ou plus)

```
proc sql;
      select     c.cust_no, c.name,
                 m.title, m.rating,
                 m.category,
                 a.actor_leading
      from dir.customers c,
                 dir.movies2 m,
                 actors a
      where c.cust_no = m.cust_no
            and m.title = a.title;
quit;
```

La proc SQL

Fusion interne (3 tableaux ou plus)

- On utilise and dans la clause where pour spécifier plusieurs conditions
- On obtient **les lignes vérifiant toutes les conditions et uniquement celles-là**
- On peut fusionner jusqu'à 32 tables à l'aide d'une seule requête SQL

La proc SQL

Fusion interne

```
proc sql;
      select m.title, rating, actor_leading
      from movies m
      inner join actors a
      on m.title = a.title;
quit;
```

La proc SQL

Fusion interne

- On utilise on de la même façon que where pour les fusions internes
- On peut également utiliser on pour les fusions **externes**

La proc SQL

Fusion externe gauche

```
proc sql;
     select movies.title, actor_leading,
               rating
     from movies
          left join
               actors
          on movies.title = actors.title;
quit;
```

La proc SQL

Fusion externe gauche

- On obtient **toutes les lignes vérifiant la condition spécifiée dans la clause on ET toutes les lignes de la table à GAUCHE (movies) qui ne correspondent à aucune ligne de la table à droite**
- Les lignes de la table à gauche sont donc « protégées » et donc présentes dans la table obtenue qu'elles correspondent ou non à des lignes de la table à droite
- Il est nécessaire de spécifier un nom de table pour la variable-clef dans select

La proc SQL

Fusion externe droite

```
proc sql;
     select actors.title, actor_leading,
               rating
     from movies
          right join
               actors
          on movies.title = actors.title;
quit;
```

La proc SQL

Fusion externe droite

- On obtient **toutes les lignes vérifiant la condition spécifiée dans la clause on ET toutes les lignes de la table à DROITE (actors) qui ne correspondent à aucune ligne de la table à droite**
- Les lignes de la table à droite sont donc « protégées » et donc présentes dans la table obtenue qu'elles correspondent ou non à des lignes de la table à gauche

La proc SQL

Fusion externe (concaténation)

```
proc sql;
      select * from dir.customers
      outer union
      select * from movies;
quit;
```

La proc SQL

Fusion externe (concaténation)

- On obtient une **concaténation** des deux tables, de façon analogue à ce qu'on aurait obtenu avec un set dans une étape data
- On obtient toutes les lignes de la première table **puis** toutes les lignes de la deuxième table

La proc SQL

Création de variables

```
proc sql;
      select title, length, category, year,
                  rating,
                2011-year as age
      from movies;
quit;
```

La proc SQL

Création de variables

- Il est possible de créer de nouvelles variables au sein d'une instruction select
- Le nom de la nouvelle variable est spécifié par as
- Bien noter l'ordre requis par la clause as

La proc SQL

Statistiques descriptives avec SQL

```
proc sql;
     select *,
          count(title) as notitle,
          max(year) as most_recent,
          min(year) as earliest,
          sum(length) as total_length,
          nmiss(rating) as nomissing
     from movies
     group by rating;
quit;
```

La proc SQL

Statistiques descriptives avec SQL

- Les fonctions statistiques de base sont disponibles
- Toutes ces fonctions peuvent être appliquées par groupes (clause group by)

La proc SQL

Insertion d'observations

```
proc sql noprint;
      insert into customers
            values(1 'Peng');
      insert into customers
            set cust_no = 2, name = 'Sasha';
quit;
```

La proc SQL

Insertion d'observations

- Il y a deux façons d'insérer des observations dans une table
- Le type des données doit être le même
- Première façon : values()
- les nouvelles valeurs doivent être séparées par des espaces
- Deuxième façon : set nom_de_colonne = nouvelle_valeur
- Les différentes affectations doivent être séparées par des virgules

La proc SQL

Supprimer des lignes

```
proc sql;
      delete
      from movies
      where length <= 100;
quit;
```

La proc SQL

Supprimer des colonnes

```
proc sql;
      create table new
            (drop = rating) as
      select *
      from movies;
quit;
```

La proc SQL

Mise à jour des observations

```
proc sql noprint;
      update customers
            set Name = 'Liu 2'
            where cust_no = 1;
quit;
```

La proc SQL

Mise à jour des observations

- La clause complète de mise à jour est update…set…where
- Si plusieurs lignes vérifient la condition, elles sont toutes mises à jour avec les données spécifiées dans la clause set

Références

- *SAS SQL*, P. Liu & A. Vedrashko, Haas School of Business, University Of Berkeley (disponible sur Internet)
- *Manipulating Data with PROC SQL*, K. P. Lafler, Software Intelligence Corporation (disponible sur Internet)
- *Exploring the World of PROC SQL Joins*, K. P. Lafler, Software Intelligence Corporation (disponible sur Internet)

Pour aller plus loin…

- *SAS 9.2 SQL Procedure: User's Guide*, SAS Documentation (disponible sur Internet)
- *Undocumented and Hard-to-find PROC SQL Features*, K. P. Lafler, Software Intelligence Corporation (disponible sur Internet)