Étape Data (Merge) ou SAS SQL (JOIN)??

Présenté Par Amin Guerss, PMP

ANALYSTE SENIOR SOLUTIONS DE CREDIT à la BANQUE LAURENTIENNE

Email: guerss.amin@banquelaurentienne.ca

LinkedIn: http://www.linkedin.com/pub/amin-guerss-pmp/b/2b2/266

Étape Data (Merge) ou SAS SQL (JOIN)

- Quelle technique de match-fusion est la mieux adaptée? L'étape DATA ou SAS SQL?
- Traditionnellement, la Match-fusioner des fichiers en SAS était via les Étapes data.
- SAS permet l’usage des requêtes (SQL).

Préparation des données

- Les fichiers de données utilisés doivent au moins avoir une clé similaire.
- Si les tables ne contiennent pas les mêmes noms de variables, l'étape data ne peut pas être utilisée. (possibilité d'utiliser proc SQL)
- S'assurer qu'il n'y a pas de doublons
- Une étape de tri des données (proc sort) est nécessaire avant l'utilisation de l'étape data merge.

Proc data merge

- Il existe 7 façons pour réaliser un match-fusion. Pour autant de fichier données possible.
- Ne pas oublier de trier les données en préalable en utilisant la même clé!!!

```
Exhibit 2: Default Match-Merge
------------------------------
DATA OUT;
    MERGE ONE TWO;
    BY ID;
RUN;
------------------------------
```

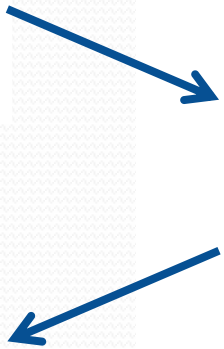


Exhibit 1: Two Input Files

FILE ONE		FILE TWO		
ID	NAME	ID	AGE	SEX
A01	SUE	A01	58	F
A02	TOM	A02	20	M
A05	KAY	A04	47	F
A10	JIM	A10	11	M

FILE OUT

ID	NAME	AGE	SEX
A01	SUE	58	F
A02	TOM	20	M
A04		47	F
A05	KAY	.	
A10	JIM	11	M

Les sept combinaisons possible du merge

```
Exhibit 3: All Match-Merge Sub-sets
-------------------------------------------
OPTIONS MERGENOBY=WARN MSLEVEL=I;
DATA ONEs TWOs inBOTH
       NOmatch1 NOmatch2 allRECS NOmatch;
 MERGE ONE(IN=In1) TWO(IN=In2);
 BY ID;
 IF In1=1 then output ONEs;
 IF In2=1 then output TWOs;
 IF (In1=1 and In2=1) then output inBOTH;
 IF (In1=0 and In2=1) then output NOmatch1;
 IF (In1=1 and In2=0) then output NOmatch2;
 IF (In1=1 OR In2=1)  then output allRECS;
 IF (In1+In2)=1       then output NOmatch;
RUN;
-------------------------------------------
```

(3d) FILE NOmatch1 (In1=0 and In2=1)

ID	NAME	AGE	SEX
A04		47	F

(3e) FILE NOmatch2 (In1=1 & In2=0)

ID	NAME	AGE	SEX
A05	KAY	.	

(3f) FILE allRECS (In1=1 OR In2=1)

ID	NAME	AGE	SEX
A01	SUE	58	F
A02	TOM	20	M
A04		47	F
A05	KAY	.	
A10	JIM	11	M

(3g) FILE NOmatch(In1+In2)

ID	NAME	AGE	SEX
A04		47	F
A05	KAY	.	

Exhibit 1: Two Input Files

FILE ONE		FILE TWO		
ID	NAME	ID	AGE	SEX
A01	SUE	A01	58	F
A02	TOM	A02	20	M
A05	KAY	A04	47	F
A10	JIM	A10	11	M

(3a) FILE ONEs (In1=1)

ID	NAME	AGE	SEX
A01	SUE	58	F
A02	TOM	20	M
A05	KAY	.	
A10	JIM	11	M

(3b) FILE TWOs (In2=1)

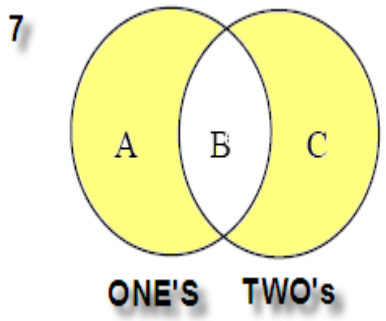
ID	NAME	AGE	SEX
A01	SUE	58	F
A02	TOM	20	M
A04		47	F
A10	JIM	11	M

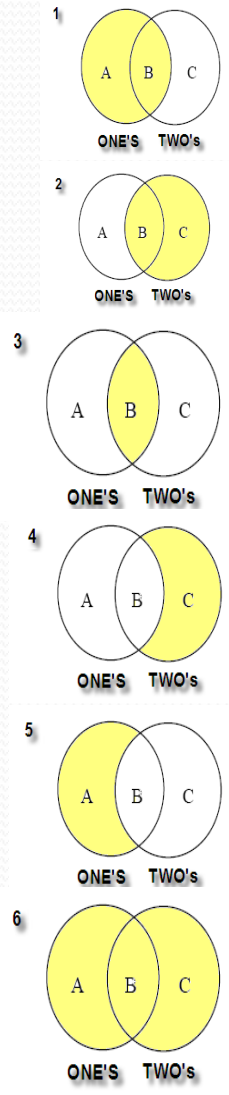
(3c) inBOTH(In1=1 & In2=1)

ID	NAME	AGE	SEX
A01	SUE	58	F
A02	TOM	20	M
A10	JIM	11	M

Diagramme de Venn

```
Exhibit 3: All Match-Merge Sub-sets
-----------------------------------------
OPTIONS MERGENOBY=WARN MSLEVEL=I;
DATA ONEs TWOs inBOTH
       NOmatch1 NOmatch2 allRECS NOmatch;
 MERGE ONE(IN=In1) TWO(IN=In2);
 BY ID;
1 IF In1=1 then output ONEs;
2 IF In2=1 then output TWOs;
3 IF (In1=1 and In2=1) then output inBOTH;
4 IF (In1=0 and In2=1) then output NOmatch1;
5 IF (In1=1 and In2=0) then output NOmatch2;
6 IF (In1=1 OR In2=1)  then output allRECS;
7 IF (In1+In2)=1       then output NOmatch;
RUN;
-----------------------------------------
```





Proc SQL

Définition:

- SQL «Structured Query Language» est un langage informatique normalisé servant à exploiter des bases de données relationnelles. La partie *langage de manipulation des données* de SQL permet de rechercher, d'ajouter, de modifier ou de supprimer des données dans les bases de données relationnelles.

- Un programmeur SAS peut utiliser des instructions SQL pour manipuler des ensembles de données dans un environnement SAS via PROC SQL.

- Il existe quatre types de jointure dite s d'horizontales:
  1. l'intene «INNER JOIN»,
  2. à gauche «LEFT JOIN»,
  3. à droite «RIGHT JOIN»
  4. bilatérale «FULL OUTER JOIN»

  Toutes ces jointures sont des produits cartésiens faites sur des variables clés spécifiées.

Proc SQL : INNER JOIN

```
Exhibit 14: An Inner Join
-------------------------------------
PROC SQL;
  CREATE TABLE QinBOTH AS
    SELECT *
    FROM ONE inner join TWO
    ON ONE.ID=TWO.ID
 ;
QUIT;
-------------------------------------
```

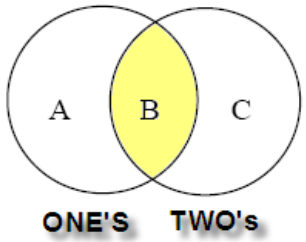
Exhibit 1: Two Input Files

FILE ONE		FILE TWO		
ID	NAME	ID	AGE	SEX
A01	SUE	A01	58	F
A02	TOM	A02	20	M
A05	KAY	A04	47	F
A10	JIM	A10	11	M

FILE QinBOTH

ID	NAME	AGE	SEX
A01	SUE	58	F
A02	TOM	20	M
A10	JIM	11	M

3



Proc SQL : LEFT JOIN

```
Exhibit 16: An SQL Left Outer Join
-----------------------------------
PROC SQL;
  CREATE TABLE qONEs AS
    SELECT *
    FROM ONE left join TWO
    ON ONE.ID=TWO.ID
 ;
QUIT;
-----------------------------------
```

Exhibit 1: Two Input Files

FILE ONE

ID	NAME
A01	SUE
A02	TOM
A05	KAY
A10	JIM

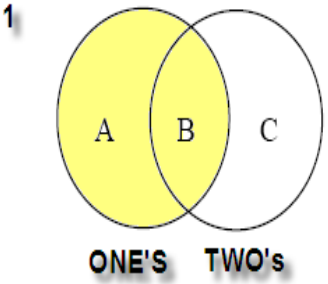
FILE TWO

ID	AGE	SEX
A01	58	F
A02	20	M
A04	47	F
A10	11	M

FILE qONEs

ID	NAME	AGE	SEX
A01	SUE	58	F
A02	TOM	20	M
A05	KAY	.	
A10	JIM	11	M

1



Proc SQL : Right JOIN

```
Exhibit 16: An SQL Left Outer Join
-----------------------------------
PROC SQL;
  CREATE TABLE qONEs AS
    SELECT *
    FROM ONE left join TWO
    ON ONE.ID=TWO.ID
 ;
QUIT;
-----------------------------------
```

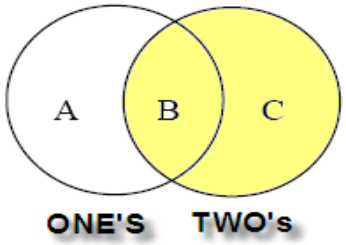
Exhibit 1: Two Input Files

FILE ONE		FILE TWO		
ID	NAME	ID	AGE	SEX
A01	SUE	A01	58	F
A02	TOM	A02	20	M
A05	KAY	A04	47	F
A10	JIM	A10	11	M

FILE qTWOs

ID	NAME	AGE	SEX
A01	SUE	58	F
A02	TOM	20	M
A04		47	F
A10	JIM	11	M

2



Proc SQL : FULL OUTER JOIN

Exhibit 17: A Full Outer Join (Surprise)

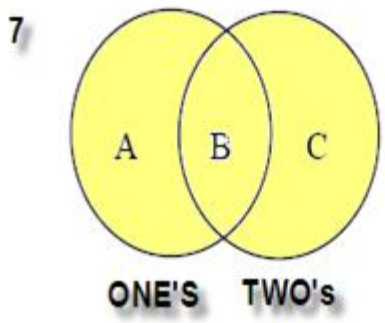
```
PROC SQL;
  CREATE TABLE ALL AS
    SELECT *
    FROM ONE full join TWO
    ON ONE.ID=TWO.ID
 .
 QUIT;
```

Exhibit 1: Two Input Files

FILE ONE		FILE TWO		
ID	NAME	ID	AGE	SEX
A01	SUE	A01	58	F
A02	TOM	A02	20	M
A05	KAY	A04	47	F
A10	JIM	A10	11	M

FILE ALL

ID	NAME	AGE	SEX
A01	SUE	58	F
A02	TOM	20	M
		47	F
A05	KAY	.	
A10	JIM	11	M



Étape Merge Vs proc SQL

```
Exhibit 3: All Match-Merge Sub-sets
-------------------------------------------
OPTIONS MERGENOBY=WARN MSLEVEL=I;
DATA ONEs TWOs inBOTH
       NOmatch1 NOmatch2 allRECS NOmatch;
 MERGE ONE(IN=In1) TWO(IN=In2);
 BY ID;
 IF In1=1 then output ONEs;
 IF In2=1 then output TWOs;
 IF (In1=1 and In2=1) then output inBOTH;
 IF (In1=0 and In2=1) then output NOmatch1;
 IF (In1=1 and In2=0) then output NOmatch2;
 IF (In1=1 OR In2=1)  then output allRECS;
 IF (In1+In2)=1       then output NOmatch;
RUN;
-------------------------------------------
```

```
Exhibit 16: An SQL Left Outer Join
-----------------------------------
PROC SQL;
  CREATE TABLE qONEs AS
    SELECT *
    FROM ONE left join TWO
    ON ONE.ID=TWO.ID
 ;
QUIT;
-----------------------------------
```


```
Exhibit 16: An SQL Left Outer Join
-----------------------------------
PROC SQL;
  CREATE TABLE qONEs AS
    SELECT *
    FROM ONE left join TWO
    ON ONE.ID=TWO.ID
 ;
QUIT;
-----------------------------------
```


```
Exhibit 14: An Inner Join
-----------------------------------
PROC SQL;
  CREATE TABLE QinBOTH AS
    SELECT *
    FROM ONE inner join TWO
    ON ONE.ID=TWO.ID
 ;
QUIT;
-----------------------------------
```

```
Exhibit 17: A Full Outer Join (Surprise)
-----------------------------------
PROC SQL;
  CREATE TABLE ALL AS
    SELECT *
    FROM ONE full join TWO
    ON ONE.ID=TWO.ID
;
 QUIT;
-----------------------------------
```

Conclusion

- les Étapes «data merge» et le proc SQL sont différentes.
- Le match-fusion a été conçu pour manipuler des ensembles de données triées.
- SQL a été conçu pour créer et manipuler des ensembles de données à partir de bases de données relationnelles.
- Les étapes data permettent de combiner plusieurs fichiers à la fois et de générer 7 résultats possibles.
- Les jointures permettent la même souplesse, mais ne produisent qu'un résultat à la fois.

Conclusion

Proc data Merge Vs PROC SQL	
Caractéristique d'une fusion (Proc Merge)	**Caractéristique d'une jointure (Proc SQL)**
Ne concerne que le logiciel SAS et n'est pas réutilisable dans d'autres bases de données.	Le code généré réutilisable dans autres bases de données relationnelles.
Les données doivent d'abord être triées par valeur.	Les données n'ont pas besoin d'être triées à l'avance
Nécessite le même nom de variable	La correspondance des noms de variable n'est pas requise
Les résultats ne sont pas automatiquement affichés	Les résultats sont automatiquement affichés à moins que l'option NOPRINT soit spécifiée.
Plusieurs étapes de tris de données sont nécessaires	
Possibilité d'avoir plusieurs fichiers de sortis	
Recommandations	
Usage fréquent	Usage adhoc
Adaptée pour le tenue des données «data lineage» et validation des données	Plus performante et moins fastidieuse
	Réutilisable dans d'autre environnement

Références

- http://support.sas.com/resources/papers/proceedings12/251-2012.pdf
- http://www2.sas.com/proceedings/sugi25/25/cc/25p109.pdf
- http://www2.sas.com/proceedings/sugi30/249-30.pdf
- http://www.nesug.org/Proceedings/nesug11/cc/cc31.pdf
- Foley, Malachy J. "Advanced Match-Merging: Techniques, Tricks and Traps" Proceedings of the Twenty-Second Annual SAS Users Group International Conference (1997) pp 199-206.
- Foley, Malachy J. "Match-Merging: 20 Some Traps and How to Avoid Them" Proceedings of the Twenty-Third Annual SAS Users Group International Conference (1998) pp 277-286.
- SAS Institute, Inc., SAS Language: Reference, Version 6, First Edition (Carry, NC: SAS Institute Inc., 1990)
- SAS Institute, Inc., SAS Procedures Guide, Version 8, First Edition (Carry, NC: SAS Institute Inc., 1999)

Question

Présenté Par Amin Guerss, PMP

ANALYSTE SENIOR SOLUTIONS DE CREDIT à la BANQUE LAURENTIENNE

Email: guerss.amin@banquelaurentienne.ca

LinkedIn: http://www.linkedin.com/pub/amin-guerss-pmp/b/2b2/266