

Lundi de la SGQA – 22 oct. 2007

Introduction au langage SQL et procédure SQL de SAS

Eric.Venot@jouy.inra.fr
Herve.Lagant@jouy.inra.fr



SGQA

ALIMENTATION
AGRICULTURE
ENVIRONNEMENT

INRA

Préambule (1)

SQL : Structured Query Language

- Créé par IBM en 1970
- Conforme à la norme ANSI et ISO dans les 80's
- Pour communiquer avec les SGBDR

(Système de Gestion des Bases de Données Relationnelles) :

- définir, manipuler et interroger des tables et des vues d'une base de données
- efficace
- facile à apprendre et à utiliser
- complet sur le plan fonctionnel (définit, extrait et manipule les données dans les tables)
- adopté par tous (?) les SGBDR (Oracle, DB2, Microsoft Access, MySQL, PostgreSQL, ...)

Préambule (2)

Instructions SQL

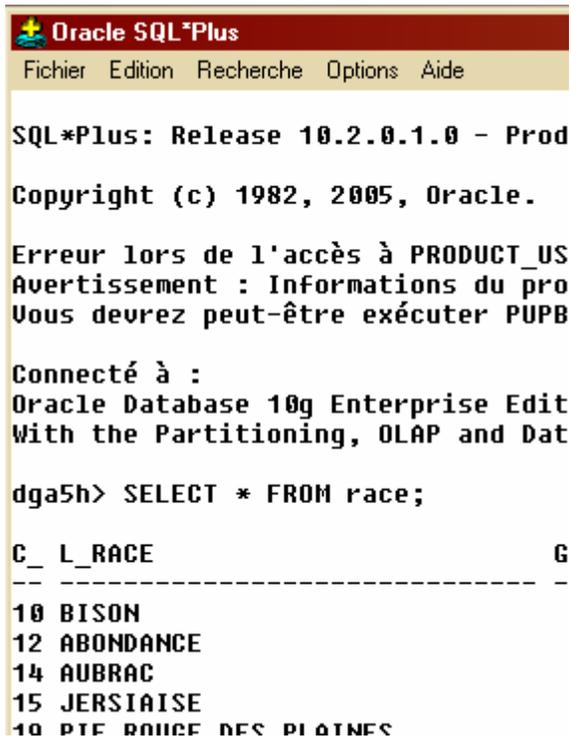
SELECT	Extraction de données de la base
INSERT UPDATE DELETE MERGE	Langage de manipulation des données (LMD) : entrer une nouvelle ligne, modifier des lignes existantes, supprimer des lignes dans la base de données
CREATE ALTER DROP RENAME TRUNCATE	Langage de définition de données (LDD) : définir, modifier et supprimer des structures de données dans les tables
COMMIT ROLLBACK SAVEPOINT	Contrôle des transactions : gère les modifications apportées par les instructions LMD
GRANT REVOKE	Langage de contrôle de données (LCD) : gère les droit d'accès à la base et aux structure qu'elle contient

Préambule (3)

Microsoft Access -> Requêtes

Les SGBDR :

Oracle -> SQL*Plus



```
Oracle SQL*Plus
Fichier Edition Recherche Options Aide

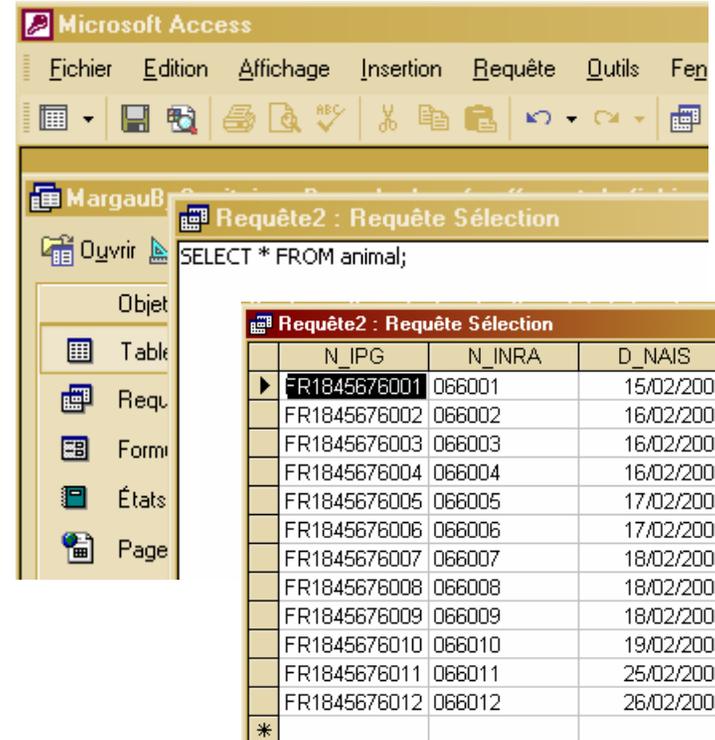
SQL*Plus: Release 10.2.0.1.0 - Prod
Copyright (c) 1982, 2005, Oracle.

Erreur lors de l'accès à PRODUCT_US
Avertissement : Informations du pro
Vous devrez peut-être exécuter PUPB

Connecté à :
Oracle Database 10g Enterprise Edit
With the Partitioning, OLAP and Dat

dga5h> SELECT * FROM race;

C_ L_RACE                                G
-----
10 BISON
12 ABONDANCE
14 AUBRAC
15 JERSIAISE
10 PTE ROUGE DES PLAINES
```



Microsoft Access

MargauB

Requête2 : Requête Sélection

```
SELECT * FROM animal;
```

	N_IPG	N_INRA	D_NAIS	
▶	FR1845676001	066001	15/02/2006	
	FR1845676002	066002	16/02/2006	
	FR1845676003	066003	16/02/2006	
	FR1845676004	066004	16/02/2006	
	FR1845676005	066005	17/02/2006	
	FR1845676006	066006	17/02/2006	
	FR1845676007	066007	18/02/2006	
	FR1845676008	066008	18/02/2006	
	FR1845676009	066009	18/02/2006	
	FR1845676010	066010	19/02/2006	
	FR1845676011	066011	25/02/2006	
	FR1845676012	066012	26/02/2006	
	*			

DB2 -> QMS

Allez voir sur DGA1...

Préambule (4)

La terminologie :

SAS Datastep	SQL (SGBDR)
Dataset	Table
Variable	Colonne
Observation	Ligne
Merge	Jointure

L'instruction « SELECT »

```
SELECT * | { [DISTINCT] colonne |expression [AS alias], ...}  
FROM table1 [AS alias]  
      {[LEFT|RIGHT] [INNER|OUTER] JOIN table2  
      [ON colonne1=colonne2]...}  
WHERE expression  
ORDER BY {colonne, expression} [ASC |DESC]  
GROUP BY colonnel [, colonne2, ...]  
HAVING expression  
  
;
```

Exemple de données

Fichier de performances PERF

anim	d_pesee	poids
1	01/01/2000	20
1	01/02/2000	65
1	01/03/2000	110
2	01/01/2000	25
2	01/02/2000	85
3	01/01/2000	30
3	01/02/2000	45
4	01/01/2000	40
4	01/02/2000	58
4	01/03/2000	76
5	01/01/2000	50
5	01/02/2000	71

Pedigree PED

NUNATI	SEXE	NUPERE	NUMERE
FR1	F	A	D
FR2	F	A	E
FR3	M	B	F
FR4	F	B	G
FR6	F	C	I

Le « SELECT » pas à pas...



Comment voir le contenu de la table PERF ?

```
PROC SQL;  
    SELECT *  
    FROM PERF;  
QUIT;
```



Comment voir le numéro et le poids des animaux ?

```
PROC SQL;  
    SELECT anim, poids  
    FROM PERF;  
QUIT;
```

Le « SELECT » pas à pas...



Comment afficher la liste des animaux pesés ?

```
PROC SQL;  
    SELECT DISTINCT anim  
    FROM PERF;  
QUIT;
```

Le « SELECT » pas à pas...

Les tris :



Comment trier les animaux par ordre croissant et leurs pesées par ordre décroissant ?

```
PROC SQL;  
    SELECT *  
    FROM PERF  
    ORDER BY anim, poids DESC ;  
QUIT;
```

Le « SELECT » pas à pas...

Les opérateurs :

* / + -

Les chaînes de caractères littérales :

= chaîne de caractère ou nombre ou date inclus dans une liste **SELECT**
placés entre apostrophes

la chaîne définie apparaît sur chaque ligne renvoyée

```
SELECT anim || ' pèse ' || put(poids, 5.1)
```

```
FROM ...
```

```
FR1234567890 pèse 123.0
```

```
FR2345678901 pèse 234.5
```

```
...
```

Le « SELECT » pas à pas...

Les alias (... AS ...):

- renommer les colonnes
- utiliser un nom alternatif aux tables

Les attributs :

- FORMAT=
- INFORMAT=
- LABEL=
- LENGTH=

```
SELECT p.anim, d_pesee, p.poids FORMAT=6.2,  
       (p.d_pesee-p.d_nais) AS age_pesee LABEL="Age à la pesée"  
FROM perf AS p ;
```

Le « SELECT » pas à pas...

Les conditions de comparaison :

=	(eg)	Égal à
>	(gt)	Supérieur à
>=	(ge)	Supérieur ou égal à
<	(lt)	Inférieur à
<=	(le)	Inférieur ou égal à
<>	(ne)	Différent de
BETWEEN ... AND ...		Compris entre ... et ... (bornes comprises)
IN (... , ...)		Correspond à une valeur de la liste
LIKE		Ressemblance partielle de chaîne de caractère
IS MISSING		Correspond à une valeur manquante *
IS NULL		Correspond à une valeur NULL *

* dans SAS, proc SQL traite les valeurs manquantes comme les "blancs" ou les "0", et les considère comme des valeurs NULL

Le « SELECT » pas à pas...

Les conditions logiques :

AND	Renvoie TRUE si les 2 conditions sont vraies
OR	Renvoie TRUE si l'une des conditions est vraie
NOT	Renvoie TRUE si la condition qui suit l'opérateur est fausse

Le « SELECT » pas à pas...



Comment sélectionner les pesées supérieures à 100 kg de l'animal "1" ?

```
PROC SQL;  
    SELECT *  
    FROM perf  
    WHERE anim='1' AND poids>100 ;  
QUIT;
```



Comment sélectionner les animaux dont le numéro commence par FR ?

```
PROC SQL;  
    SELECT *  
    FROM ped  
    WHERE nunati LIKE 'FR%' ;  
QUIT;
```

Le « SELECT » pas à pas...



Comment sélectionner les pesées de tous les animaux sauf 1 et 2 ?

```
PROC SQL;  
    SELECT *  
    FROM PERF  
    WHERE anim NOT IN ('1', '2') ;  
QUIT;
```



Comment sélectionner les pesées des animaux dont le poids est compris entre 100 et 200 kg ?

```
PROC SQL;  
    SELECT *  
    FROM PERF  
    WHERE poids BETWEEN 100 AND 200 ;  
QUIT;
```

Le « SELECT » pas à pas...

Les fonctions SQL monolignes :

Toutes les fonctions de SAS ! (sauf lag)

Générales	Numériques	Alpha-numériques	Date
CASE	ROUND	LOWER / UPPER	YEAR
	TRUNC	SUBSTR	MONTH
	MOD	CONCAT	
		TRIM	
		REPLACE	
		LPAD / RPAD	
		INSTR	
...

Le « SELECT » pas à pas...



Comment sélectionner les pesées, arrondies au kg, effectuées au mois de juin et juillet ?

```
PROC SQL;  
    SELECT anim, d_pesee, MONTH(d_pesee) AS mois,  
           ROUND(poids, 1) AS poids  
    FROM PERF  
   WHERE MONTH(d_pesee) IN (6, 7) ;  
QUIT;
```

Le « SELECT » pas à pas...

Le cas « CASE » :

```
CASE WHEN expr_comp_1 THEN expr_retournée_1
      [WHEN expr_comp_2 THEN expr_retournée_2
      WHEN expr_comp_n THEN expr_retournée_n
      ELSE expr_autre]
END
```



Comment corriger le poids des animaux selon le père ?

```
PROC SQL;
    SELECT anim, pere,
           CASE WHEN pere='P1' THEN 1.10*poids
                WHEN pere='P2' THEN 1.20*poids
                ELSE poids END AS poids_corr
    FROM PERF ;
QUIT;
```

Le « CREATE TABLE »



Comment obtenir la liste des animaux pesés dans une nouvelle table SAS ?

```
PROC SQL;  
  
    CREATE TABLE perf_bis AS  
  
    SELECT DISTINCT anim  
  
    FROM PERF ;  
  
QUIT;
```

Le « INSERT INTO »



Comment ajouter une pesée supplémentaire ?

```
PROC SQL;  
    INSERT INTO PERF  
        VALUES ( '4' , '01/02/2000' , 100 ) ;  
QUIT;
```

OU

```
PROC SQL;  
    INSERT INTO PERF  
    SET  anim='4' ,  
        d_pesee= '01/02/2000' ,  
        poids=100 ;  
QUIT;
```

remarque : même syntaxe pour l'instruction « UPDATE », mais



Le « INSERT INTO »



Comment ajouter les pesées du fichier performance d'avril 2007 ?

```
PROC SQL;  
  
    INSERT INTO PERF  
  
        SELECT *  
  
        FROM PERF_200704;  
  
QUIT;
```

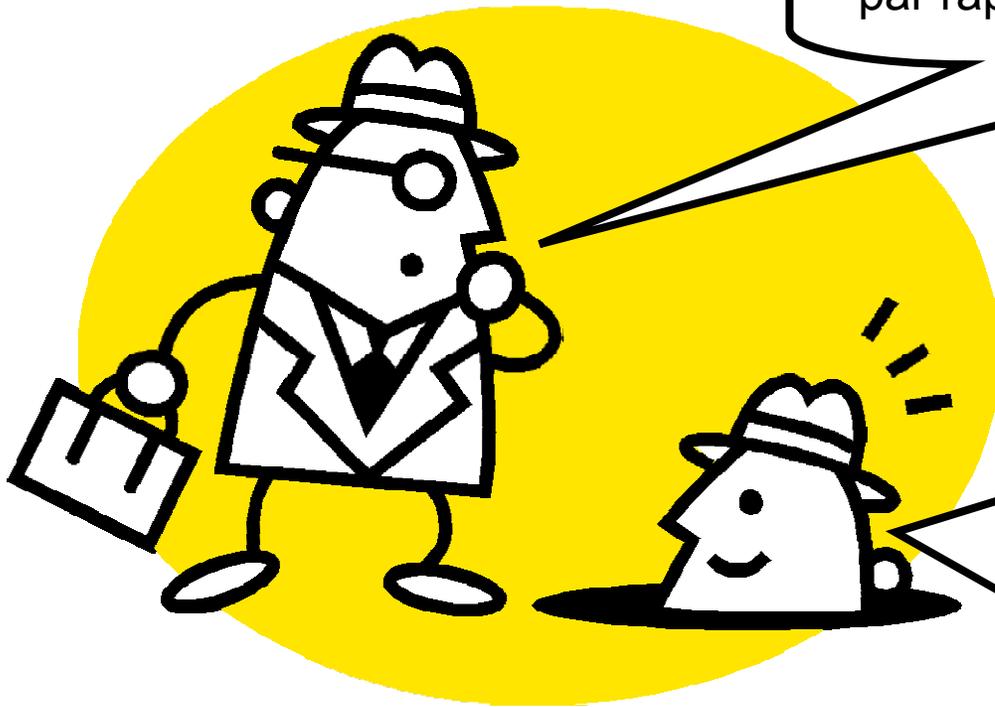
Le « DELETE »



Comment supprimer les pesées de l'année 2007 ?

```
PROC SQL;  
  
    DELETE FROM PERF  
  
    WHERE YEAR(d_pesee) = 2007 ;  
  
QUIT;
```





Mais quel intérêt dans tout ça
par rapport au langage SAS « classique » ?

On va creuser un peu le sujet...

- les fonctions d'aggrégations,
- les instructions imbriquées,
 - les jointures,
 - les macro-variables,

...



Les fonctions d'agrégation pour examiner les données

Fonction	Définition
COUNT, FREQ, N	Nombre de valeurs non manquantes
NMISS	Nombre de valeurs manquantes
AVG, MEAN	Moyenne
STDERR	Erreur standard à la moyenne
MIN, MAX	Minimum, Maximum
RANGE	Etendue des valeurs
SUM	Somme des valeurs
VAR, STD	Variance, Ecart-Type
...	



Les fonctions d'agrégation pour examiner les données



On veut calculer le nombre de pesées, le nombre d'animaux différents, les poids minimum et maximum dans le fichier PERF

```
PROC SQL;  
  
    SELECT  
        COUNT(*) AS nbpesees,  
        COUNT(DISTINCT anim) AS nbanim,  
        MIN(poids) AS mini, MAX(poids) AS maxi  
    FROM PERF;  
  
QUIT;
```

output:

<code>nbpesees</code>	<code>nbanim</code>	<code>mini</code>	<code>maxi</code>
12	5	20	110



Les fonctions d'agrégation pour examiner les données



On veut calculer le nombre de pesées, les poids minimum et maximum pour chaque animal

```
PROC SQL;  
  
    SELECT anim,  
           COUNT(*) AS nbpesees,  
           MIN(poids) AS mini, MAX(poids) AS maxi  
    FROM PERF  
    GROUP BY anim ;  
  
QUIT;
```



Les fonctions d'agrégation pour examiner les données



On veut calculer le nombre de pesées, les poids minimum et maximum pour chaque animal

QUE POUR LES ANIMAUX AVEC PLUS DE 2 PESEES

```
PROC SQL;  
  
    SELECT anim,  
           COUNT(*) AS nbpesees,  
           MIN(poids) AS mini, MAX(poids) AS maxi  
    FROM PERF  
   GROUP BY anim  
  
   HAVING nbpesees>2;  
  
QUIT;
```

Les instructions imbriquées

Les sous requêtes :



Comment sélectionner les pesées supérieures à la moyenne générale des pesées ?

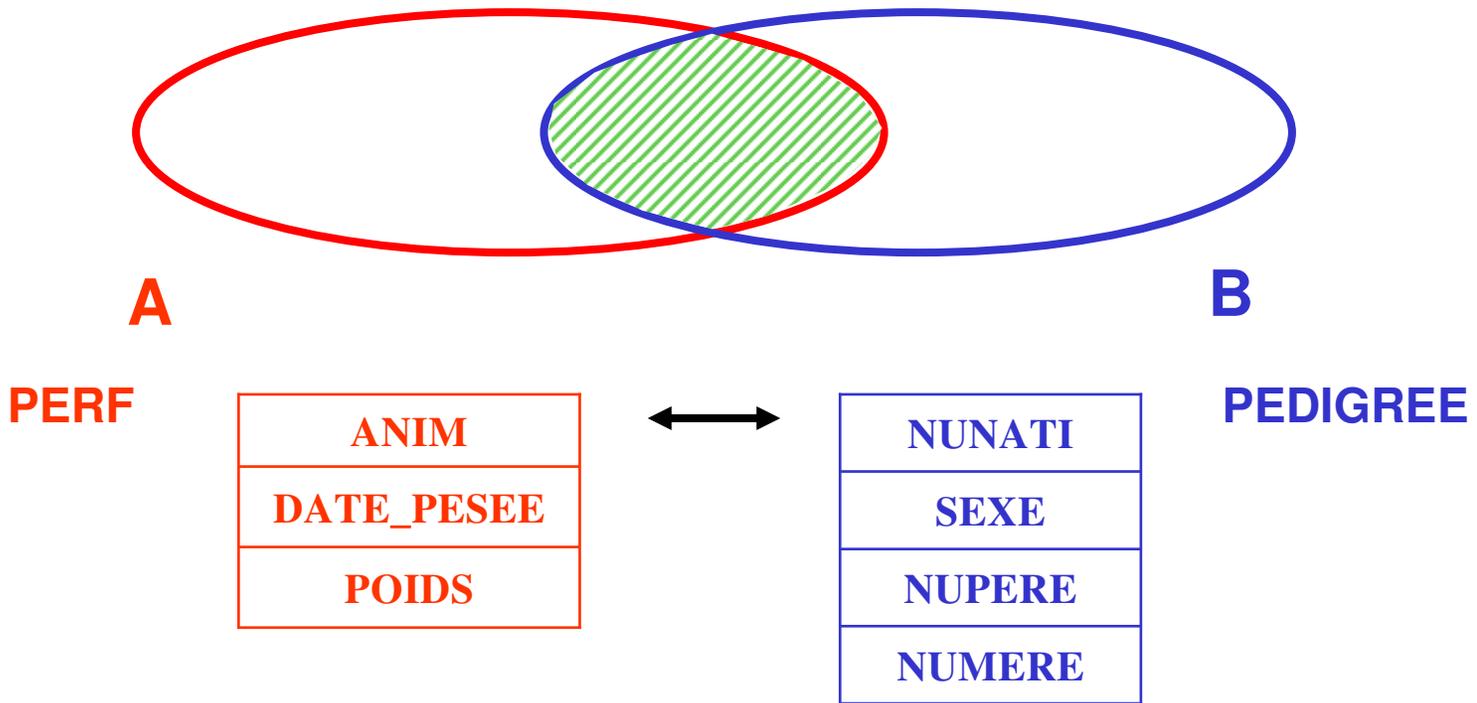
```
PROC SQL;  
  SELECT anim, d_pesee, poids  
  FROM PERF  
  WHERE poids > (SELECT MEAN(poids) FROM perf) ;  
QUIT;
```



Comment sélectionner les pesées des animaux avec généalogie ?

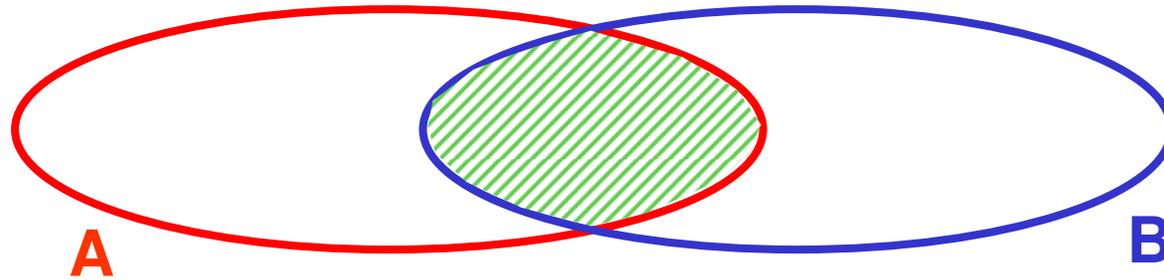
```
PROC SQL;  
  SELECT anim, d_pesee, poids  
  FROM PERF  
  WHERE anim IN (SELECT SUBSTR(nunati,3,1) FROM perf);  
QUIT;
```

Les jointures



On souhaite ajouter le numéro de la mère dans la table perf pour les femelles

Les jointures



4 étapes:

1- le renommage de la clé

2- les tris

3- la fusion
+ élimination des variables
en trop

4- sélection des femelles

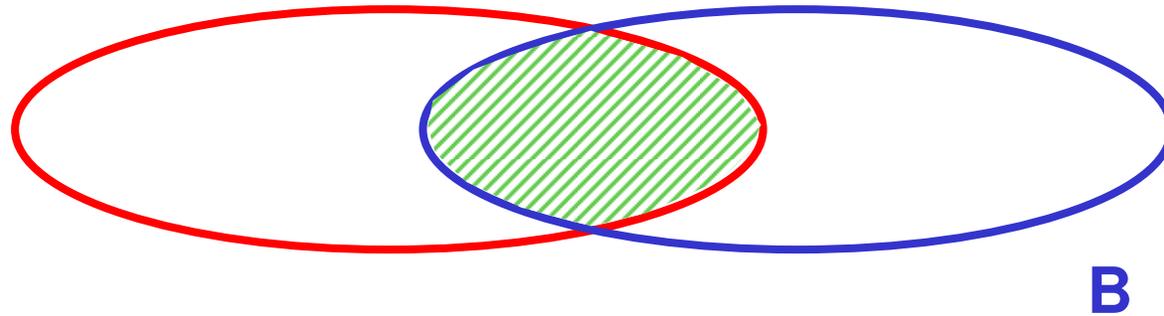
```
DATA PED1 (DROP=NUNATI); SET PED;  
    ANIM=SUBSTR(NUNATI, 3, 1);
```

```
PROC SORT DATA=PERF; BY ANIM;  
PROC SORT DATA=PED1; BY ANIM;
```

```
DATA SORTIE (DROP=SEXE NUPERE);  
    MERGE PERF (IN=A) PED1 (IN=B);  
    BY ANIM;  
    IF A AND B;
```

```
    IF SEXE='F';
```

Les jointures



```
PROC SQL;
```

```
    CREATE TABLE SORTIE AS  
    SELECT T1.*, T2.numere  
    FROM PERF T1 INNER JOIN PED T2  
    ON T1.anim = SUBSTR(T2.nunati, 3, 1)  
    WHERE T2.sexe = 'F';
```

```
QUIT;
```

Concrètement...

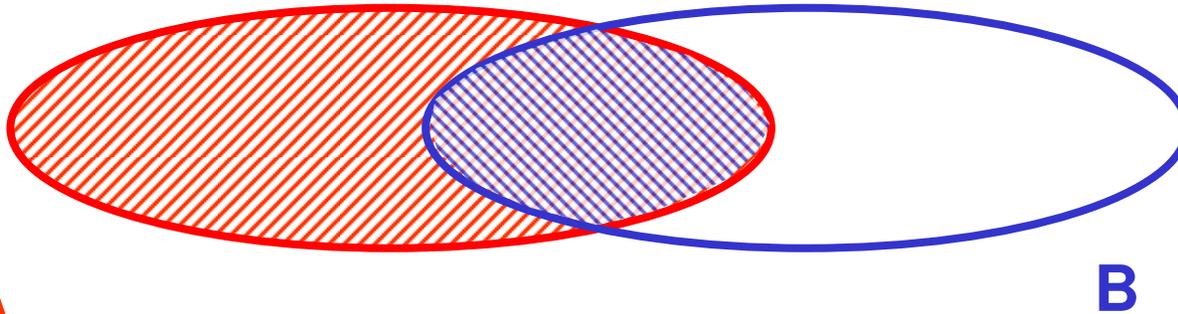
PERF → SORTIE

anim	d_pesee	poids	NUMERE
1	01/01/2000	20	A
1	01/02/2000	65	A
1	01/03/2000	110	A
2	01/01/2000	25	A
2	01/02/2000	85	A
3	01/01/2000	30	A
3	01/02/2000	45	A
4	01/01/2000	40	B
4	01/02/2000	58	B
4	01/03/2000	76	B
5	01/01/2000	50	B
5	01/02/2000	71	B

Pedigree PED

NUNATI	SEXE	NUPERE	NUMERE
FR1	F	A	D
FR2	F	A	E
FR3	M	B	F
FR4	F	B	G
FR6	F	C	I

Les jointures



A

B

```
PROC SQL;
```

```
CREATE TABLE SORTIE AS  
SELECT T1.*, T2.numere  
FROM PERF T1 LEFT JOIN PED T2  
ON T1.anim = SUBSTR(T2.nunati, 3, 1)  
WHERE T2.sexe = 'F';
```

```
QUIT;
```

Concrètement...

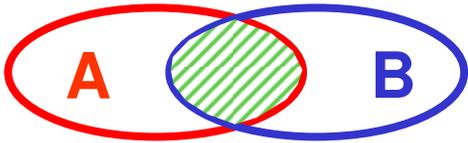
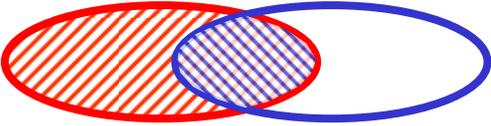
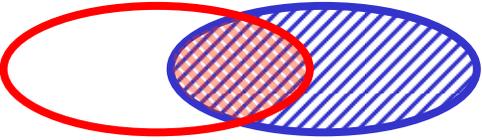
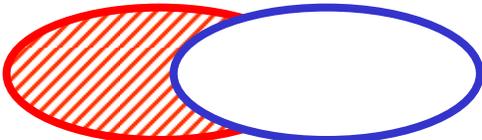
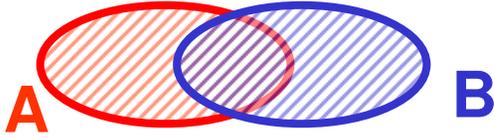
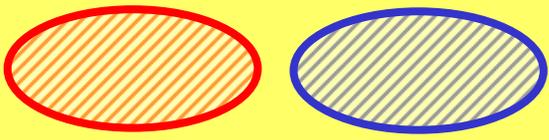
PERF → SORTIE

anim	d_pesee	poids	NUMERE
1	01/01/2000	20	A
1	01/02/2000	65	A
1	01/03/2000	110	A
2	01/01/2000	25	A
2	01/02/2000	85	A
3	01/01/2000	30	.
3	01/02/2000	15	.
4	01/01/2000	40	B
4	01/02/2000	58	B
4	01/03/2000	76	B
5	01/01/2000	50	.
5	01/02/2000	71	.

Pedigree PED

NUNATI	SEXE	NUPERE	NUMERE
FR1	F	A	D
FR2	F	A	E
FR3	M	B	F
FR4	F	B	G
FR6	F	C	I

Les jointures

	DATA MERGE	PROC SQL
	IF A AND B	INNER JOIN
	IF A	LEFT JOIN
	IF B	RIGHT JOIN
	IF A AND NOT B	LEFT JOIN WHERE T2.NUNATI IS NULL
	IF NOT A AND B	RIGHT JOIN WHERE T1.ANIM IS NULL
	IF A OR B	OUTER JOIN
	DATA SET	UNION

Un exemple



On veut obtenir les pesées des filles du taureau C supérieures à la moyenne de tous les animaux

3 étapes:

- 1- On sélectionne les pesées supérieures à la moyenne générale
- 2- On sélectionne la liste des filles du taureau C
- 3- On fusionne les 2 tables.

```
PROC SQL;  
    CREATE TABLE PES AS  
    SELECT anim, poids, d_pesee  
    FROM PERF  
    WHERE poids > (SELECT MEAN(poids) FROM PERF);  
  
    CREATE TABLE LISTE AS  
    SELECT SUBSTR(nunati,3,1) AS anim  
    FROM PED  
    WHERE nupere='C' AND sexe='F';  
  
    CREATE TABLE FINAL AS  
    SELECT T1.*  
    FROM PES T1 INNER JOIN LISTE T2  
    ON T1.anim = T2.anim;  
  
QUIT;
```

Le même exemple en 1 étape!



On veut obtenir les pesées des filles du taureau C supérieures à la moyenne de tous les animaux

1- On sélectionne les pesées supérieures à la moyenne générale

2- On sélectionne la liste des filles du taureau C

3- On fusionne les 2 tables (sans tri, ni renommage).

```
PROC SQL;
  SELECT T1.*
  FROM (SELECT anim, poids, d_pesee
        FROM PERF
        WHERE poids > (SELECT MEAN(poids)
                      FROM PERF) T1
        INNER JOIN PED T2
        ON T1.anim = T2.SUBSTR(nunati, 3, 1)
        WHERE T2.nupere = 'C' AND T2.sexe = 'F' ;
  QUIT;
```

Les macro-variables

```
PROC SQL;
  SELECT MEAN(poids), STD(poids) INTO :moy, :et
  FROM PERF WHERE poids > 100;

  TITLE "Attention: il y a des valeurs à vérifier" ;
  SELECT * FROM PERF
  WHERE    ( poids > &moy + 3*&et )
          OR ( poids < &moy - 3*&et );

QUIT;
```

=> Macrovariables créées (&moy, &et...) utilisables ensuite partout dans le programme: Titre, bornes des graphiques...

Toujours plus sur les macro-variables

```
PROC SQL NOPRINT;
```

```
SELECT (DISTINCT ANIM) INTO :NUM1 - :NUM99
```

```
FROM PERF;
```

=> 1 macro-variable par animal

```
SELECT (DISTINCT ANIM) INTO: NUMERO
```

```
SEPARATED BY ','
```

```
FROM PERF;
```

=> 1 macro-variable pour tous les animaux

```
QUIT;
```



Les informations cachées

Grâce à Proc SQL :

- ▶ on peut obtenir la description des tables SAS (structure, contenu...)

```
PROC SQL;  
  DESCRIBE TABLE PERF;  
QUIT;
```

- ▶ on peut accéder aux tables du dictionnaire qui regroupe toutes les informations sur les bibliothèques SAS, les tables SAS, les options du système et les fichiers externes associés à la session.

```
PROC SQL;  
  SELECT LIBNAME, MEMNAME, CRDATE, MODATE, NVAR, NOBS  
  FROM DICTIONARY.TABLES  
  WHERE LIBNAME="WORK" AND MEMNAME="PERF" ;  
QUIT;
```

On se résume...

- ▶ langage structuré, simple et très répandu
- ▶ sélection, renommage, labellisation et choix de l'ordre des colonnes simples
- ▶ merge entre tables sans renommage, ni tri préalable
- ▶ simplification de la création des macro-variables
- ▶ utilisation des informations cachées sur les tables SAS (dictionnaire)

Quelques liens utiles...

SAS Guide to the SQL procedure (SAS Institute INC.)

La référence en français dans le texte (très complet avec exemples) :

<http://wwwsas.stat.ucl.ac.be/sasdiscute/doc/tutorials/base/sql.html>

Cours général sur SAS, dont SQL (en français) :

<http://cedric.cnam.fr/~saporta/Cours%20SAS%20v2.pdf>

Introductions à SAS PROC SQL (en anglais) :

<http://www2.sas.com/proceedings/sugi26/p150-26.pdf>

<http://www2.sas.com/proceedings/sugi27/p070-27.pdf>

Les dix meilleures raisons d'utiliser PROC SQL (en anglais) :

<http://www2.sas.com/proceedings/sugi29/042-29.pdf>

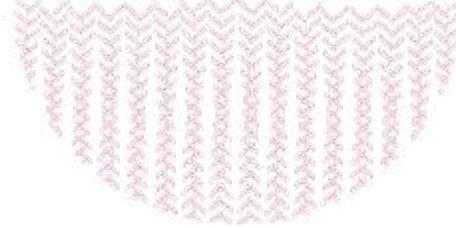
Match en direct entre DATA STEP et PROC SQL (en anglais) (score 8-3 pour SQL!) :

http://www.pauldickman.com/teaching/sas/proc_sql_slides_20041026.pdf

Détails sur toutes les informations disponibles dans le dictionnaire des tables :

<http://www2.sas.com/proceedings/sugi24/Posters/p254-24.pdf>

et bien d'autres exemples sur le Net ...



Et comme tout bon utilisateur de SQL...
à la fin d'un travail, on vous...

QUIT ;