

Partie 2 : SQL en langage SAS

Le but de cette partie est de réécrire les procédures en langage uniquement. Vous n'êtes pas autorisé à utiliser la procédure sql.

- Restriction : clause where, and, or et opérateurs classiques

PROC SQL; SELECT * FROM CLIENTS WHERE cpcli <> 57500; QUIT;	DATA Exemple_1; SET CLIENTS; WHERE cpcli ne 57500; /*ne : not equal */ RUN;
PROC SQL; SELECT * FROM Clients WHERE villecli= "Saint-Avoid" AND sexe= "H" AND age < 23; QUIT;	DATA Exemple_1; SET CLIENTS; WHERE villecli="Saint-Avoid" AND sexe="H" AND age<23; RUN;

- Restriction: where, upper, lower, between, is null

PROC SQL; SELECT * FROM Clients WHERE Age BETWEEN 18 AND 21; QUIT;	DATA Exemple_1; SET CLIENTS; WHERE Age >= 18 AND Age <= 21; RUN;
PROC SQL; SELECT * FROM Clients WHERE Age BETWEEN 18 AND 21; QUIT;	DATA Exemple_1; SET CLIENTS; WHERE Age >= 18 AND Age <= 21; RUN;

- Restriction: where, in, like

PROC SQL; SELECT * FROM Clients WHERE sexe IN ('H','F'); QUIT;	DATA Exemple_1; SET CLIENTS; WHERE sexe IN ("H", "F"); RUN;
PROC SQL; SELECT * FROM Clients WHERE Nomcli LIKE 'L%'; QUIT;	DATA Exemple_1; SET CLIENTS; WHERE substr(Nomcli, 1, 1) = "L"; RUN;

- Gestion des doublons : Distinct et renommage

<pre>PROC SQL; SELECT DISTINCT Villecli FROM Clients; QUIT;</pre>	<pre>PROC SORT DATA=Clients; BY Villecli; RUN; /*distinct donc on trie les données*/ DATA Exemple_1; SET Clients; BY Villecli; IF FIRST.Villecli THEN OUTPUT; RUN;</pre>
<pre>PROC SQL; SELECT DISTINCT Villecli as ville, sexe FROM clients; QUIT;</pre>	<pre>PROC SORT DATA=Clients; BY Villecli Sexe; RUN; DATA Exemple_1 (KEEP=Ville Sexe); SET Clients; BY Villecli Sexe; IF FIRST.Sexe THEN OUTPUT; RUN;</pre>

-Les jointures : INNER JOIN - LEFT JOIN

<pre>PROC SQL; SELECT * FROM Clients cl INNER JOIN Locations Lo ON cl.codecli=lo.codecli INNER JOIN Films fi ON lo.codefilm =fi.codefilm WHERE cl.cpcli=57800; QUIT;</pre>	<pre>DATA CL; SET Clients; WHERE cpcli=57800; RUN; DATA LO; SET Locations; WHERE codecli IN (SELECT codecli FROM CL); RUN; DATA Exemple_1; MERGE LO(in=inlo) Films(in=infi); BY codefilm; IF inlo AND infi; RUN;</pre>
<pre>PROC SQL; SELECT * FROM Clients cl LEFT JOIN Locations Lo ON cl.codecli=lo.codecli INNER JOIN Films fi ON lo.codefilm =fi.codefilm WHERE fi.codefilm IN (1,5,7); QUIT;</pre>	<pre>DATA FILMS; SET Locations (WHERE=(codefilm IN (1,5,7))); RUN; DATA Exemple_1; MERGE Clients(in=incl) FILMS(in=infi); BY codecli; IF incl; RUN; DATA Exemple_2; MERGE FILMS(in=infi) Films(in=iff); BY codefilm; IF infi AND iff; RUN;</pre>

- Requêtes et calculs arithmétiques

<pre>PROC SQL; SELECT * FROM produits WHERE (CodeCateg=1 AND PrixUnit > 50) OR (CodeCateg=3 AND PrixUnit < 90); QUIT;</pre>	<pre>DATA PRODUITS; SET produits; WHERE (CodeCateg=1 AND PrixUnit > 50) OR (CodeCateg=3 AND PrixUnit < 90); RUN;</pre>
<pre>PROC SQL; SELECT RefProd, PrixUnit *(UnitesStock-5) as soustraction FROM produits WHERE UnitesStock >= 5; QUIT;</pre>	<pre>DATA PRODUITS; SET produits; IF UnitesStock >= 5 THEN soustraction = PrixUnit * (UnitesStock-5); DROP UnitesStock PrixUnit; RENAME soustraction = PrixTotal; RUN;</pre>

- Traitement conditionnel: CASE WHEN

<pre> PROC SQL; SELECT Refprod, Nomprod, PrixUnit, CASE WHEN PrixUnit<=50 THEN "Bas prix" WHEN 50 < PrixUnit <=90 THEN "Prix moyens" ELSE "Produits de luxe" END AS Gamme FROM Produits ; QUIT; </pre>	<pre> DATA PRODUITS; SET Produits; IF PrixUnit <= 50 THEN Gamme = "Bas prix"; ELSE IF PrixUnit <= 90 THEN Gamme = "Prix moyens"; ELSE Gamme = "Produits de luxe"; DROP PrixUnit; RUN; </pre>
<pre> PROC SQL; SELECT Refprod, UnitesStock, UnitesCom, NiveauReap, CASE WHEN UnitesCom > 40 THEN "Déjà commandé" WHEN UnitesCom < NiveauReap THEN "A Commander " WHEN UnitesCom = 0 THEN "N'est plus en stock " ELSE " Disponible " END AS Informations FROM Produits ; QUIT; </pre>	<pre> DATA PRODUITS; SET Produits; IF UnitesCom > 40 THEN Informations = "Déjà commandé"; ELSE IF UnitesCom < NiveauReap THEN Informations = "A Commander"; ELSE IF UnitesCom = 0 THEN Informations = "N'est plus en stock"; ELSE Informations = "Disponible"; DROP UnitesCom NiveauReap; RUN; </pre>

- Agrégation

<pre> PROC SQL; SELECT NoFour, COUNT(*) AS Nombre_de_produits, ROUND(AVG(PrixUnit),.001) AS PrixMoyen, MIN(PrixUnit) AS PrixMinimum, MAX(PrixUnit) AS PrixMaximum FROM produits GROUP BY NoFour ORDER BY NoFour; QUIT; </pre>	<pre> data produits_summary; set produits; by NoFour; if first.NoFour then do; Nombre_de_produits = 1; PrixTotal = PrixUnit; PrixMinimum = PrixUnit; PrixMaximum = PrixUnit; end; else do; Nombre_de_produits + 1; PrixTotal + PrixUnit; if PrixUnit < PrixMinimum then PrixMinimum = PrixUnit; if PrixUnit > PrixMaximum then PrixMaximum = PrixUnit; end; if last.NoFour then do; PrixMoyen = PrixTotal / Nombre_de_produits; output; end; drop PrixUnit PrixTotal; run; proc sort data=produits_summary; by NoFour; run; proc print data=produits_summary; var NoFour Nombre_de_produits PrixMoyen PrixMinimum PrixMaximum; run; </pre>
<pre> PROC SQL; SELECT NoFour, COUNT(*) AS Nombre_de_produits, ROUND(AVG(PrixUnit)) AS PrixMoyen, MIN(PrixUnit) AS PrixMinimum, MAX(PrixUnit) AS PrixMaximum FROM produits GROUP BY NoFour HAVING Nombre_de_produits < 2 ORDER BY NoFour; QUIT; </pre>	<pre> PROC SORT DATA=produits; BY NoFour; RUN; PROC MEANS DATA=produits NOPRINT; BY NoFour; VAR PrixUnit; OUTPUT OUT=stats N(COUNT(*))=Nombre_de_produits MEAN= PrixMoyen MIN= PrixMinimum MAX= PrixMaximum; RUN; DATA stats; </pre>

	<pre>SET stats; IF Nombre_de_produits < 2; DROP _TYPE__FREQ_; RUN; PROC SORT DATA=stats; BY NoFour; RUN;</pre>
--	--