

# APPLICATION 2

## UTILISATION DES ÉTAPES DATA

0

## UTILISATION DES ÉTAPES DATA - PARTIE 1

Créer une librairie nommée app2

Stocker la table CARS contenue dans l'onglet CARS du fichier  
Excel transmis

0

# SOLUTION

```
Libname app2 'C:\...\SAS FILES';  
PROC IMPORT  
  OUT = app2.CARS  
  DATAFILE = 'C:\...\Cars et report.xls'  
  DBMS = XLS REPLACE;  
  SHEET = 'CARS';  
  GETNAMES = YES;  
RUN;
```

0

## UTILISATION DES ÉTAPES DATA - PARTIE 2

Créer 2 tables

- La première nommée CARCAR comprendra les variables alphanumériques de la table CARS
- La seconde nommée CARNUM comprendra les variables numériques de la table CARS

Créer une table qui contient les modèles dont le prix de vente conseillé (MSRP) est supérieur à 50 000 \$

0

# SOLUTION

```
/*Création de CARCAR dans la librairie WORK*/  
DATA CARCAR;  
    SET app2.CARS(KEEP=Make Model Type Origin DriveTrain);  
RUN;  
  
/*Création de CARNUM dans la librairie WORK*/  
DATA CARNUM;  
    SET app2.CARS(DROP=Make Model Type Origin DriveTrain);  
RUN;  
  
/*Création de sup_50 dans la librairie WORK qui contiendra les modèles dont le  
prix de vente conseillé est supérieur à 50 000$*/  
DATA sup_50;  
    SET app2.CARS(WHERE=(MSRP > 50000));  
RUN;
```

0

## UTILISATION DES ÉTAPES DATA - PARTIE 2

Créer 3 tables qui contiennent les modèles européens,  
américains et asiatiques

0

# SOLUTION

```
/*Création de 3 tables dans la librairie app2 qui contiendront les modèles  
selon les 3 origines géographiques*/
```

```
DATA app2.Asia app2.Europe app2.US;  
    SET app2.CARS;  
    if Origin = 'Asia' then output app2.Asia;  
    if Origin = 'Europe' then output app2.Europe;  
    if Origin = 'USA' then output app2.US;  
RUN;
```

0

## UTILISATION DES ÉTAPES DATA - PARTIE 3

Modifier la variable DriveTrain afin qu'elle corresponde aux exigences de sa traduction

- Ses modalités deviendront Avant, Arrière et Toutes
- Son label doit être modifié afin de pouvoir lire dans la table "Transmission de la voiture" en lieu et place de "DriveTrain"

0

# SOLUTION

```
/*Création d'une table CARS_TRANS dans la librairie app2 qui contiendra les informations traduites de la variable DriveTrain*/
```

```
DATA app2.Cars_trans;  
  SET app2.CARS;  
  Format trans $25.;  
  Label trans = 'Transmission de la voiture';  
  if DriveTrain = 'Rear' then trans = 'Arrière';  
  if DriveTrain = 'Front' then trans = 'Avant';  
  if DriveTrain = 'ALL' then trans = 'Toutes';  
  Drop DriveTrain;  
RUN;
```

0

## UTILISATION DES ÉTAPES DATA - PARTIE 3

Convertissez la variable MPG\_City en variable alphanumérique  
Puis tenter de reconverter cette nouvelle variable en variable  
numérique

0

# SOLUTION

```
/*Création de city_code et city_num*/  
DATA app2.Cars_trans;  
    SET app2.Cars_trans;  
    Format City_code $2.;  
    City_code = put(MPG_City,$2.);  
    City_num = input(City_code,8.);  
RUN;
```

0

## UTILISATION DES ÉTAPES DATA - PARTIE 4

Importer puis concaténer les tables contenues dans les onglets  
report\_fr, report\_de et report\_uk du fichier excel transmis

Identifier la provenance géographique des table en ajoutant une  
variable Country qui aura 3 modalités :

- FR
- DE
- UK

0

# SOLUTION

```
/*Import des 3 reports FR, DE et UK*/  
PROC IMPORT  
    OUT = app2.Report_FR  
    DATAFILE = 'C:\...\Cars et report.xls'  
    DBMS = XLS REPLACE;  
    SHEET = 'Report_FR inc '  
    GETNAMES = YES;  
RUN;  
  
PROC IMPORT  
    OUT = app2.Report_DE  
    DATAFILE = 'C:\...\Cars et report.xls'  
    DBMS = XLS REPLACE;  
    SHEET = 'Report_DE inc';  
    GETNAMES = YES;  
RUN;
```

0

# SOLUTION

```
PROC IMPORT  
    OUT = app2.Report_UK  
    DATAFILE = 'C:\...\Cars et report.xls'  
    DBMS = XLS REPLACE;  
    SHEET = 'Report_UK inc';  
    GETNAMES = YES;  
RUN;  
  
/*Création de report_mond qui est la concaténation des 3 reporting avec une  
nouvelle Variable Country qui identifiera de quel bureau provient  
l'information*/  
DATA app2.report_mond;  
    SET app2.Report_FR (in=fr) app2.Report_DE (in=de) app2.Report_UK;  
    if fr = 1 then Country = 'FR';  
    else if de = 1 then Country = 'DE';  
    else Country = 'UK';  
RUN;
```

0

# UTILISATION DES ÉTAPES DATA - PARTIE 5

Nous souhaitons maintenant conserver l'unicité des individus  
Fusionner les rapports avec la table qui liste tous les modèles afin  
d'ajouter les résultats des ventes et des stocks des  
concessionnaires français, allemands et anglais dans 6 colonnes  
distinctes

0

## SOLUTION

```
/*Modification et tri du rapport français*/  
DATA report_fr;  
    SET app2.Report_FR;  
    rename  
        Constructeur = make  
        Modele = Model  
        nb_vente = sales_fr  
        Stock = stock_fr  
;  
    label  
        Constructeur = 'Make FR'  
        Modele = 'Model FR'  
        nb_vente = 'Sales FR'  
        Stock = 'Stock FR'  
;  
RUN;  
/*Tri de la table en vue de la fusion*/  
PROC SORT;by make model;RUN;
```

0



# SOLUTION

```
/*Modification et tri du rapport allemand*/
DATA report_de;
  SET app2.Report_DE;
  rename
    Baumeister = make
    Modell = Model
    Verkauf = sales_de
    Stock = stock_de
;
  label
    Baumeister = 'Make DE'
    Modell = 'Model DE'
    Verkauf = 'Sales DE'
    Stock = 'Stock DE'
;
RUN;
/*Tri de la table en vue de la fusion*/
PROC SORT;by make model;RUN;
```

0

# SOLUTION

```
/*Modification et tri du rapport anglais*/
DATA report_UK;
  SET app2.Report_UK;
  rename
    sales = sales_uk
    Stock = stock_uk
;
  label
    Make = 'Make UK'
    Modell = 'Model UK'
    Verkauf = 'Sales UK'
    Stock = 'Stock UK'
;
RUN;
/*Tri de la table en vue de la fusion*/
PROC SORT;by make model;RUN;
/*Tri de la MASTER table en vue de la fusion*/
PROC SORT data=app2.Cars;by make model;RUN;
```

0

# SOLUTION

```
/*FUSION des rapports avec la master table CARS*/  
DATA app2.Report_Monde_Final;  
    MERGE app2.CARS report_FR report_DE report_UK;  
    by make model;  
RUN;
```

0

## UTILISATION DES ÉTAPES DATA - PARTIE 6

Créer 2 nouvelles variables

- sales\_euro\_cont qui sera la somme des ventes françaises et allemandes
- sales\_world qui sera la somme des ventes dans les trois pays

ATTENTION

Identifier par 'Données incomplètes' les modèles qui n'existe pas dans tous les pays

Qu'êtes-vous obligés de faire pour faire apparaître cette modalité dans les variables?

0

# REMARQUES PRÉLIMINAIRES

Sous SAS, les variables ne peuvent être numériques et alphanumériques

Il faudra penser à mettre la somme en format alphanumérique afin que la modalité 'Données incomplètes' puissent être implémentée dans les variables sales\_euro\_cont et sales\_world

Dans une conditionnalité comme if then else, les instructions sont uniques après les mots-clé then et else

Si plusieurs instructions sont nécessaires il faudra penser à programmer une boucle du type DO END

0

## SOLUTION

```
/*Calcul des variables et identifications des valeurs manquantes*/  
DATA app2.Report_Monde_Final;  
  SET app2.Report_Monde_Final;  
  if (sales_fr ne . and sales_de ne . and sales_uk ne .)  
  then do;  
    sales_euro_cont = put((sales_fr + sales_de), $20.);  
    sales_world = put((sales_fr + sales_de + sales_uk), $20.);  
  end;  
  else do;  
    sales_euro_cont = 'Données incomplètes';  
    sales_world = 'Données incomplètes';  
  end;  
RUN;
```

0

# UTILISATION DES ÉTAPES DATA - PARTIE 7

Créer à l'aide des instructions input et cards, 3 nouveaux modèles

VAR	Modèle 1	Modèle 2	Modèle 3
Make	Mercedes-Benz	Mercury	Scion
Model	CL550 2dr	Mountaineer Road 2	xZ 4dr hatch core
Type	Sedan	SUV	Sedan
Origin	Europe	USA	Asia
DriveTrain	Rear	Front	Front
MSRP	104 820\$	39 995\$	10 000\$
Invoice	93 324\$	32 317\$	8 500\$
EngineSize	5	4	1.5
Cylinders	8	6	4
Horsepower	302	210	108
MPG_City	16	16	32
MPG_Highway	24	21	38
Weight	4085	4374	2340
Wheelbase	114	114	93
Lenght	196	190	154

## SOLUTION

```
/*Création de 3 nouveaux modèles*/
```

```
DATA app2.NEW_CARS ;
```

```
format Make Origin $13.;
```

```
format Model $40.;
```

```
format Type $6.;
```

```
format DriveTrain $5.;
```

```
format MSRP Invoice DOLLAR21.2;
```

```
input Make $ Model $ Type $ Origin $ DriveTrain $ MSRP Invoice EngineSize  
Cylinders Horsepower MPG_City MPG_Highway Weight Wheelbase Length;
```

```
Cards;
```

```
Mercedes CL550_2dr Sedan Europe Rear 104820.00 93324.00 5 8 302 16 24 4085  
114 196
```

```
Mercury Mountaineer_Road_2 SUV USA Front 39995.00 32317.00 4 6 210 16 21 4374  
114 190
```

```
Scion xZ_4dr_hatch_core Sedan Asia Front 10000.00 8500.00 1.5 4 108 32 38  
2340 93 154
```

```
;
```

```
RUN;
```

# UTILISATION DES ÉTAPES DATA - PARTIE 7

Ajouter les trois nouveaux modèles aux anciens dans une table nommée Resume\_Cars\_New

0

## SOLUTION

```
/*Concaténation des fichiers*/  
DATA app2.Resume_NEW;  
    SET app2.CARS app2.NEW_CARS;  
RUN;
```

0