# Marion CRAIPEAU p1502716

08/02/2016

# CIRQUE :

**1/** Créer et alimenter cette table avec un jeu de données.

CREATE TABLE PERSONNEL (

NOM VARCHAR2(15),

ROLE VARCHAR2(20)

);

INSERT INTO PERSONNEL values ('Kevin', ‘Clown’);

INSERT INTO PERSONNEL values ('Jeremie', ‘Dompteur de Lion’);

INSERT INTO PERSONNEL values ('Martin', 'Jongleur');

INSERT INTO PERSONNEL values ('Marion', 'Jongleur');

INSERT INTO PERSONNEL values ('Guillaume', 'Dompteur de Jeremie');

**2/** Ecrire un bloc PL/SQL anonyme permettant de :

• compter le nombre de n-uplets dans la table PERSONNEL et stocker le résultat dans une variable ;

• compter le nombre d'employés dont le rôle est « Jongleur » dans la table PERSONNEL et stocker le résultat

dans une deuxième variable ;

• calculer la proportion (en pourcentage), stocker le résultat dans une troisième variable et afficher le résultat

à l'écran.

set serveroutput on

DECLARE

v\_nb\_jongleur NUMBER(2);

v\_nb\_employe NUMBER(2);

v\_pourcentage\_jongleur NUMBER(2);

BEGIN

select count(\*) into v\_nb\_employe from PERSONNEL;

DBMS\_OUTPUT.PUT\_LINE('Nombre d''employés : ' || v\_nb\_employe);

select count(\*) into v\_nb\_jongleur from PERSONNEL where upper(role) = upper('jongleur');

DBMS\_OUTPUT.PUT\_LINE('Nombre de Jongleurs : ' || v\_nb\_jongleur);

v\_pourcentage\_jongleur := (100\* v\_nb\_jongleur / v\_nb\_employe);

DBMS\_OUTPUT.PUT\_LINE('Pourcentage de Jongleurs dans la troupe : ' || v\_pourcentage\_jongleur);

END;

/

**3.** Inclure dans le programme précédent un traitement d'exception permettant de détecter si la table PERSONNEL

est vide, c'est-à-dire, que le nombre total de n-uplets dans PERSONNEL est égal à zéro. Dans ce cas, déclencher

une erreur fatale. Tester en effaçant tout le contenu de la table PERSONNEL.

set serveroutput on

DECLARE

v\_nbPersonnes NUMBER(3);

v\_nbJongleurs NUMBER(3);

v\_pourcentage NUMBER(5,2);

e\_tableVide EXCEPTION;

BEGIN

select count(\*) into v\_nbPersonnes from PERSONNEL;

DBMS\_OUTPUT.PUT\_LINE('Nombre d'employés : ' || v\_nbPersonnes);

select count(\*) into v\_nb\_jongleur from PERSONNEL where role = 'Jongleur';

DBMS\_OUTPUT.PUT\_LINE('Nombre de Jongleurs : ' || v\_nbJongleurs);

IF (v\_nbPersonnes is null) THEN

RAISE e\_tableVide;

END IF;

select (v\_nbJongleurs\*100/v\_nbPersonnes) INTO v\_pourcentage FROM PERSONNEL;

DBMS\_OUTPUT.PUT\_LINE('Pourcentage de jongleurs : ' || v\_pourcentage);

EXCEPTION

WHEN e\_tableVide THEN

RAISE\_APPLICATION\_ERROR(-20001, 'La table Personnel est vide');

WHEN OTHERS THEN

DBMS\_OUTPUT.PUT\_LINE(SQLERRM);

END;

/

# CATALOGUE

Ecrire un programme PL/SQL (bloc PL/SQL anonyme) permettant d'afficher votre catalogue système (liste des

tables et des vues de votre compte) disponible grâce à la vue système TAB

set serveroutput on

DECLARE

CURSOR c\_tab is

select \* from TAB;

v\_tab c\_tab%ROWTYPE;

BEGIN

OPEN c\_tab;

LOOP

FETCH c\_tab INTO v\_tab;

EXIT WHEN c\_tab%NOTFOUND;

DBMS\_OUTPUT.PUT\_LINE('L''objet ' ||v\_tab.TNAME|| ' est de type ' || v\_tab.tabtype||'.');

END LOOP;

CLOSE c\_tab;

END;

/

# LES GROS NAVIONS

1/ Création de table et insertion des jeux de données.

CREATE TABLE AVION(

AvNum NUMBER constraint PK\_AVION Primary Key,

AvNom VARCHAR2(15) not null,

Capacite NUMBER not null,

Localisation VARCHAR2(20) not null);

CREATE TABLE PILOTE(

PlNum NUMBER constraint PK\_PILOTE Primary Key,

PlNom VARCHAR2(15) not null,

PlPrenom VARCHAR2(15) not null,

Ville VARCHAR2(20) not null,

Salaire NUMBER not null

);

CREATE TABLE VOL(

VolNum NUMBER constraint PK\_VOL Primary Key,

PlNum NUMBER not null ,

AvNum NUMBER not null,

VilleDep VARCHAR2(20) not null,

VilleArr VARCHAR2(20) not null,

HeureDep TIMESTAMP not null,

HeureArr TIMESTAMP not null,

Constraint fk\_PILOTE\_VOL Foreign Key (PlNum) references PILOTE(PlNum),

Constraint fk\_AVION\_VOL Foreign Key (AvNum) references AVION(AvNum));

INSERT INTO AVION values (1, 'A300s', 300, 'Lyon');

INSERT INTO AVION values (2, 'A310', 400, 'Marseille');

INSERT INTO AVION values (3, 'A250', 200, 'Paris');

INSERT INTO AVION values (4, 'A300', 340, 'Nates');

INSERT INTO AVION values (5, 'A500', 300, 'Lilles');

INSERT INTO AVION values (6, 'A205', 200, 'Bourges');

INSERT INTO AVION values (7, 'A320', 380, 'Lyon');

INSERT INTO AVION values (8, 'A310', 380, 'Paris');

INSERT INTO PILOTE values (1, 'Debars', 'Jean', 'Lyon', 15000);

INSERT INTO PILOTE values (2, 'Bollot', 'Martin', 'Marseille', 20000);

INSERT INTO PILOTE values (3, 'Guillot', 'Jeremie', 'Lilles', 22000);

INSERT INTO PILOTE values (4, 'Coissard', 'Kevin', 'Nantes', 10000);

INSERT INTO PILOTE values (5, 'Guillaume', 'Meneroud', 'Paris', 20000);

INSERT INTO VOL values (7, 3, 8, 'Bordeaux', 'Bourg-en-Bresse', TO\_TIMESTAMP('05:50', 'HH24:MI'), TO\_TIMESTAMP('07:05', 'HH24:MI') );

Insert into VOL values (6, 4, 2, 'La Rochelle', 'Strasbourg', TO\_TIMESTAMP('08:00', 'HH24:MI'), TO\_TIMESTAMP('09:05', 'HH24:MI') );

2/ Dans un bloc PL/SQL anonyme, déclarer un curseur permettant de lire les données suivantes : numéro de

vol, numéro d'avion, heure de départ et heure d'arrivée des vols pour lesquels l'avion utilisé est le n° 1, 2, 4 ou 8.

Pour chaque vol lu par le curseur, calculer le temps de vol, le réduire dans la proportion voulue selon l'avion

utilisé, puis mettre à jour l'attribut HEUREARR de ce vol dans la table VOL.

set serveroutput on

DECLARE

CURSOR c\_vol is

select volnum, avnum, TO\_CHAR(heuredep, 'HH24:MI') as Depart, TO\_CHAR(heurearr, 'HH24:MI') as arrive, heurearr - heuredep as duree from VOL where avnum in (1,2,4,8);

v\_vol c\_vol%ROWTYPE;

BEGIN

OPEN c\_vol;

LOOP

FETCH c\_vol INTO v\_vol;

EXIT WHEN c\_vol%NOTFOUND;

UPDATE VOL SET HeureArr=HeureDep+v\_vol.duree\*0.9

WHERE AvNum=1 or AvNum=4;

UPDATE VOL SET HeureArr=HeureDep+v\_vol.duree\*0.85

WHERE AvNum=2 or AvNum=8;

END LOOP;

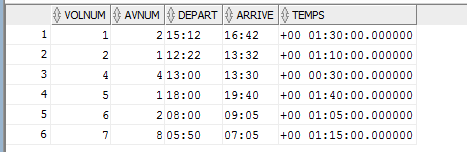
CLOSE c\_vol;

END;

/

3/ TEST :

Liste des vols des avions 1,2, 4 et 8 avant changement de durée de vol :



Liste des vols des avions 1,2, 4 et 8 après changement de durée de vol :

