NGUYEN Duc Hau

ROBERT Quentin

***BDD : Travail 3***

Table des matières

[Introduction : 2](#_Toc498956198)

[Création d’une vue : 3](#_Toc498956199)

[Trigger 3](#_Toc498956200)

[RechercherVolsOuverts 4](#_Toc498956201)

[OuvrirVol 4](#_Toc498956202)

[EffectuerReservation 6](#_Toc498956203)

[ListerVolsPlusCherAVG 7](#_Toc498956204)

[ModifierAdresse 8](#_Toc498956205)

[Conclusion 10](#_Toc498956206)

## Introduction :

Dans ce tp nous allons travailler les vues, les triggers, et les blocs pl sql. Et ce afin de pouvoir mieux gérer les bases de données.

## Création d’une vue :

Permet d’afficher l’intégralité des informations client d’un seul coup (nom, prénom, numéro du passport, adresse, téléphone, nombre de réservation.

La requête est la suivante :

**CREATE** **OR** **REPLACE** **VIEW** FicheClient**(**nom**,** prenom**,** numpassport**,** adresse**,** telephone**,** nbReservation **)**

**AS** **SELECT** **DISTINCT** P**.**NOM**,** P**.**PRENOM**,** Passeport**.**NUMERO**,** '(' **||** T**.**CODE\_PAYS **||** ') ' **||** T**.**CODE\_REGION **||** '-' **||** T**.**NUMERO**,**A**.**APP **||** ' ' **||** A**.**NUMERO **||** ' ' **||** A**.**RUE **||** ' ' **||** A**.**VILLE **,** **count(**RP**.**PERSONNE**)**

**FROM** Personne P**,** Telephone\_Personne TP**,** Telephone T**,**

Adresse\_Personne AP**,** Adresse A**,**

Passeport**,**

Reservation\_Personne RP**,** Reservation R**,** Etat\_Reservation ER

**WHERE** P**.**ID **=** TP**.**PERSONNE **AND** TP**.**TELEPHONE **=** T**.**ID

**AND** P**.**ID **=** AP**.**PERSONNE **AND** AP**.**ADRESSE **=** A**.**ID

**AND** P**.**PASSEPORT **=** Passeport**.**ID

**AND** P**.**ID **=** RP**.**PERSONNE **AND** RP**.**RESERVATION **=** R**.**ID **AND** R**.**ETAT **=** ER**.**ID **AND** ER**.**LIBELLE **LIKE** '%Valide%'

**GROUP** **BY** P**.**NOM**,** P**.**PRENOM**,** Passeport**.**NUMERO**,** T**.**CODE\_PAYS **,** T**.**CODE\_REGION **,** T**.**NUMERO **,** A**.**APP **,** A**.**NUMERO **,** A**.**RUE **,** A**.**VILLE**;**

**/**

Pour afficher le contenu de la vue il suffit de l’interroger comme une table :

**SELECT** **\*** **FROM** FicheClient**;**

Oracle retourne alors des résultats (plus ou moins bien formatés) tel que :

F 95 P.O. Box 398, 8371 Amet Avenue Colchester 4

Mechanique plutôt pratique afin de faciliter l’utilisation de la bdd quand on utilise souvent une requête sur le même model.

## Trigger

Permet de déclencher une action quand il y a modification des données d’une table donnée.

Le code est le suivant :

**CREATE** **OR** **REPLACE** **TRIGGER** AlertETSA

**AFTER** **INSERT** **ON** Reservation **FOR** **EACH** **ROW**

**DECLARE**

destUSA Vol**%ROWTYPE;**

**BEGIN**

**SELECT** V**.\*** **INTO** destUSA

**FROM** Reservation\_Vol RV**,** Vol V**,** Ville\_Desservie VD**,** Ville Vi**,** Pays P

**WHERE** RV**.**RESERVATION **=** **:new.**ID

**AND** RV**.**VOL **=** V**.**ID

**AND** V**.**AEROPORT\_ARRIVE **=** VD**.**AEROPORT

**AND** VD**.**VILLE **=** Vi**.**ID

**AND** Vi**.**PAYS **=** P**.**ID

**AND** P**.**NOM **=** 'USA'**;**

-- no exception == this record is in USA

DBMS\_OUTPUT**.**PUT\_LINE**(**'Il est necessaire au client de posseder l ETSA'**);**

**EXCEPTION**

-- No data found == not in USA

**WHEN** NO\_DATA\_FOUND **THEN** **NULL;**

**END;**

**/**

Dans le cas présent le trigger s’active sur les inserts uniquement et nous affiche, si la destination est les USA :

Il est necessaire au client de posseder l ETSA

## RechercherVolsOuverts

Exemple d’appel :

**EXECUTE** CLient**.**RechercherVolsOuverts**(**'Macerata'**,** 'Abbotsford'**,** **TO\_DATE(**'01/01/2016'**,** 'DD/mm/YYYY'**),** **TO\_DATE(**'01/01/2019'**,** 'DD/mm/YYYY'**));**

Exemple de retour :

num vol: CA1520

num vol: AF2306

Le code, plutôt simple, est le suivant :

**PROCEDURE** RechercherVolsOuverts**(**ville1 Ville**.**nom**%TYPE,**

ville2 Ville**.**nom**%TYPE,** datemin **DATE,**

datemax **DATE)** **IS**

**BEGIN**

**FOR** tcur **IN** **(SELECT** v**.**numero

**FROM** Vol v**,** Aeroport a1**,** ville\_desservie vd1**,** ville v1**,** aeroport a2**,** ville\_desservie vd2**,** ville v2**,** Statut s

**WHERE** v1**.**nom **=** ville1 **AND** v2**.**nom **=** ville2 **AND** v**.**depart **>=** datemin **AND** v**.**arrivee **<=** datemax **AND** v**.**aeroport\_depart **=** a1**.**id **AND** a1**.**id **=** vd1**.**aeroport **AND** vd1**.**ville **=** v1**.**id **AND** v**.**aeroport\_arrive **=** a2**.**id **AND** a2**.**id **=** vd2**.**aeroport **AND** vd2**.**ville **=** v2**.**id **and** s**.**id **=** v**.**statut **and** s**.**libelle **=** 'Ouvert Reservation'**)** **LOOP**

DBMS\_OUTPUT**.**PUT\_LINE**(**'num vol: ' **||** tcur**.**numero**);**

**END** **LOOP;**

**END** RechercherVolsOuverts**;**

Il s’agit ici d’un simple select from where dont on parcourt les résultats pour les afficher comme on le souhaite, ici simplement par le numéro qui est un moyen simple, et concis, de définir un vol.

## OuvrirVol

Exemple d’appel :

**EXECUTE** OuvrirVol**(**'ifhihfz'**,** **TO\_DATE(**'10/09/2017'**,** 'DD/mm/YYYY'**),** **TO\_DATE(**'25/09/2017'**,** 'DD/mm/YYYY'**),** 'Cressa'**,** 'Macerata'**,** 'Compagnie 1'**);**

Exemple de retour :

PL**/SQL** **procedure** successfully completed**.**

Le retour ne montrant pas la bonne execution du bloc pl sql on peut le vérifier à la main via :

**SELECT** **\***

**FROM** vol**;**

Retournant, entre autre  :

ID NUMERO COMPAGNIE DEPART ARRIVEE AEROPORT\_DEPART

---------- --------------- ---------- --------- --------- ---------------

AEROPORT\_ARRIVE STATUT

--------------- ----------

17 ifhihfz 1 10**-**SEP-17 25**-**SEP-17 1

2 1

Le code est :

**PROCEDURE** OuvrirVol **(**

p\_numero VOL**.**NUMERO**%TYPE,**

p\_date\_depart VOL**.**DEPART**%TYPE,**

p\_date\_arrivee VOL**.**ARRIVEE**%TYPE,**

p\_ville\_depart Ville**.**NOM**%TYPE,**

p\_ville\_arrivee Ville**.**NOM**%TYPE,**

p\_compagnie Compagnie\_Aerienne**.**NOM**%TYPE**

**)** **AS**

v\_id\_compagnie Compagnie\_Aerienne**.**ID**%TYPE;**

v\_aeroport\_depart Aeroport**.**ID**%TYPE;**

v\_aeroport\_arivee Aeroport**.**ID**%TYPE;**

v\_statut\_ouvert STATUT**.**ID**%TYPE;**

depart\_notfound **EXCEPTION;**

arrivee\_notfound **EXCEPTION;**

compagnie\_notfound **EXCEPTION;**

**BEGIN**

-- Check if compagnie exist

**BEGIN**

**SELECT** ID **into** v\_id\_compagnie **FROM** Compagnie\_Aerienne **WHERE** NOM **=** p\_compagnie**;**

**EXCEPTION**

**WHEN** NO\_DATA\_FOUND **THEN**

**raise** compagnie\_notfound**;**

**END;**

-- Check if ville depart exist

**BEGIN**

**SELECT** A**.**ID **INTO** v\_aeroport\_depart

**FROM** Aeroport A**,** Ville\_Desservie VD**,** VILLE V

**WHERE** A**.**ID **=** VD**.**AEROPORT **AND** VD**.**VILLE **=** V**.**ID **AND** V**.**NOM **=** p\_ville\_depart**;**

**EXCEPTION**

**WHEN** NO\_DATA\_FOUND **THEN**

**raise** depart\_notfound**;**

**END;**

-- Check if ville arrivee exist

**BEGIN**

**SELECT** A**.**ID **INTO** v\_aeroport\_arivee

**FROM** Aeroport A**,** Ville\_Desservie VD**,** VILLE V

**WHERE** A**.**ID **=** VD**.**AEROPORT **AND** VD**.**VILLE **=** V**.**ID **AND** V**.**NOM **=** p\_ville\_arrivee**;**

**EXCEPTION**

**WHEN** NO\_DATA\_FOUND **THEN**

**raise** arrivee\_notfound**;**

**END;**

**SELECT** ID **INTO** v\_statut\_ouvert **FROM** STATUT **WHERE** LIBELLE **LIKE** '%Ouvert%'**;**

dbms\_output**.**put\_line**(**'ID : ' **||** v\_statut\_ouvert**);**

**INSERT** **INTO** VOL**(**NUMERO**,** COMPAGNIE**,** DEPART**,** ARRIVEE**,** AEROPORT\_DEPART**,** AEROPORT\_ARRIVE**,** STATUT**)**

**VALUES** **(**p\_numero**,** v\_id\_compagnie**,** **TO\_DATE(**p\_date\_depart**,** 'DD/mm/YYYY'**),** **TO\_DATE(**p\_date\_arrivee**,** 'DD/mm/YYYY'**),** v\_aeroport\_depart**,** v\_aeroport\_arivee**,** v\_statut\_ouvert**);**

**EXCEPTION**

**WHEN** depart\_notfound **THEN**

dbms\_output**.**put\_line**(**'Votre ville de depart n existe pas ou ne possede pas d aeroport'**);**

**WHEN** arrivee\_notfound **THEN**

dbms\_output**.**put\_line**(**'Votre ville d arrivee n existe pas ou ne possede pas d aeroport'**);**

**WHEN** compagnie\_notfound **THEN**

dbms\_output**.**put\_line**(**'Compagnie non trouvee'**);**

**END** OuvrirVol**;**

Le code est un peu long mais il se dégage deux parties, une vérification des entrés utilisateurs, afin de trouver les id correspondants aux villes, ect… et une 2eme partie on on réalise l’insert en lui-même. On utilise le système d’exception pour signaler un problème à l’utilisation, plutôt qu’un affichage qui passe facilement inaperçu.

## EffectuerReservation

Exemple d’appel :

**EXECUTE** Client**.**EffectuerReservation**(**1**,** 'zefzef'**,** 'Business'**);**

Exemple de retour :

PL**/SQL** **procedure** successfully completed**.**

De même le retour ne confirme pas le bon fonctionnement, on vérifie par une requête de type:

**SELECT** pl**.**vol**,**pl**.**classe**,**pl**.**place**,**pl**.**prix**,** Disponible

**FROM** Infos\_Place p**,** Place pl**,** Classe c**,** Vol v

**WHERE** v**.**id **=** 8 **AND** v**.**id **=** pl**.**vol **AND** pl**.**classe **=** c**.**id **AND** pl**.**Place **=** p**.**id **AND** p**.**Disponible **=** 'F'**;**

Qui retourne ici :

VOL CLASSE PLACE PRIX D

---------- ---------- ---------- ---------- -

8 1 2 125 F

Le code du bloc pl sql est :

**PROCEDURE** EffectuerReservation**(**

ppersonne Personne**.**ID**%TYPE,**

pvol Vol**.**NUMERO**%TYPE,**

pclasse Classe**.**LIBELLE**%TYPE**

**)** **IS**

idPlace Place**%ROWTYPE;**

pasDePlace **EXCEPTION;**

**BEGIN**

**SELECT**

pl**.**vol**,**

pl**.**classe**,**

pl**.**place**,**

pl**.**prix

**INTO** idPlace

**FROM** Infos\_Place p**,** Place pl**,** Classe c**,** Vol v

**WHERE** c**.**libelle **=** pclasse **AND** v**.**numero **=** pvol **AND** v**.**id **=** pl**.**vol **AND** pl**.**classe **=** c**.**id **AND** pl**.**Place **=** p**.**id **AND** p**.**Disponible **=** 'T'**;**

-- exception if no place available

**IF** **(**idPlace**.**vol **IS** **NULL)**

**THEN**

**RAISE** pasDePlace**;**

**END** **IF;**

-- Réserver vol

**UPDATE** Infos\_Place

**SET** Disponible **=** 'F'

**WHERE** idPlace**.**Place **=** Infos\_Place**.**id**;**

**INSERT** **INTO** Reservation **VALUES** **((Select** **max(**id**)**+1 **from** Reservation**),** 'r5'**,** **TO\_DATE(**'1/1/2017'**,** 'DD/mm/YYYY'**),** 1**);**

**INSERT** **INTO** Reservation\_Vol **VALUES** **(**idPlace**.**vol**,** **(Select** **max(**id**)** **from** Reservation**));**

**INSERT** **INTO** Reservation\_Personne **VALUES** **(**ppersonne**,** **(Select** **max(**id**)** **from** Reservation**));**

**EXCEPTION**

**WHEN** pasDePlace **THEN**

dbms\_output**.**put\_line**(**'Pas de place sur ce vol'**);**

**END** EffectuerReservation**;**

On commence par chercher si on peut trouver une place sur le vol avec les informations données, si non une exception est levée, sinon on réalise les insert dans les tables concerné, ainsi qu’un update pour signaler la place comme prise.

## ListerVolsPlusCherAVG

Exemple d’appel :

**EXECUTE** ListerVolsPlusChersAVG**(**'Business'**);**

Exemple de retour :

Prix 500 zfehgf 3 01**-**JAN-17 01**-**JAN-17 4 3 1

Prix 1000 eekuyn 3 01**-**JAN-17 01**-**JAN-17 4 3 1

Prix 666 tottoo 3 01**-**JAN-17 01**-**JAN-17 4 3 1

Prix 999 escale 2 01**-**JAN-17 01**-**JAN-17 6 4 1

Prix 550 Paris NY 2 01**-**JAN-17 01**-**JAN-17 5 4 1

Le code est :

**CREATE** **OR** **REPLACE** **FUNCTION** avgVol**(**p\_classe Classe**.**LIBELLE**%TYPE)**

**RETURN** **PLS\_INTEGER** **IS**

moyenne **PLS\_INTEGER;**

**BEGIN**

**SELECT** **AVG(**P**.**PRIX**)** **INTO** moyenne

**FROM** Place P**,** Vol V**,** Statut S**,** Classe C

**WHERE** P**.**VOL **=** V**.**ID **AND** V**.**STATUT **=** S**.**ID **AND** S**.**LIBELLE **LIKE** '%Ouvert%'

**AND** P**.**CLASSE **=** C**.**ID **AND** C**.**LIBELLE **=** p\_classe**;**

**RETURN** moyenne**;**

**END;**

**/**

**CREATE** **OR** **REPLACE** **PROCEDURE** ListerVolsPlusChersAVG **(**

p\_classe Classe**.**LIBELLE**%TYPE**

**)** **AS**

moyenne **PLS\_INTEGER;**

**BEGIN**

moyenne **:=** avgVol**(**p\_classe**);**

**FOR** rec **IN** **(**

**SELECT** V**.\*,** P**.**Prix

**FROM** Vol V**,** PLACE P**,** STATUT S**,** CLASSE C

**WHERE** V**.**ID **=** P**.**VOL **AND** P**.**CLASSE **=** C**.**ID **AND** C**.**LIBELLE **=** p\_classe

**AND** V**.**STATUT **=** S**.**ID **AND** S**.**LIBELLE **LIKE** '%Ouvert%'

**AND** P**.**Prix **>** moyenne

**)** **LOOP**

dbms\_output**.**put\_line**(**'Prix ' **||**rec**.**Prix **||**' '**||**rec**.**NUMERO **||** ' ' **||** rec**.**COMPAGNIE **||** ' ' **||** rec**.**DEPART **||** ' ' **||** rec**.**ARRIVEE

**||** ' ' **||** rec**.**AEROPORT\_DEPART **||** ' ' **||** rec**.**AEROPORT\_ARRIVE **||** ' ' **||** rec**.**STATUT**);**

**END** **LOOP;**

**END;**

**/**

Ici le code est un peu plus compliqué, nous avons utilisé une fonction pour calculer la moyenne du prix des vols. Après l’appel de cette fonction on parcourt les vols et on vérifie si leur prix est en dessous ou non de la moyenne. Si il est au-dessus on l’affiche. Ici on affiche le prix, les villes, et les dates car on ne sait pas forcement quel vol va ressortir et cela permet de l’identifier plus facilement.

## ModifierAdresse

Exemple d’appel :

**EXECUTE** Client**.**ModifierAdresse **(**1**,**1**,**12**,** 'K'**,**'Ap #595-7594 Quam Rd.'**,**'Strona'**,**'Q8V9H1'**);**

Exemple de retour :

PL**/SQL** **procedure** successfully completed**.**

Pour en vérifier le bon fonctionnement ici :

**Select** **\*** **from** adresse **where** id**=**1**;**

Qui affiche, avec tout le côté artistique d’oracle:

---------- ---------- -

RUE

--------------------------------------------------------------------------------

VILLE

--------------------------------------------------------------------------------

CODE\_P PAYS PROVINCE

------ ---------- ----------

1 12 K

Ap #595**-**7594 Quam Rd**.**

Strona

Q8V9H1 3 24

Le code de cette modification d’adresse est :

**PROCEDURE** ModifierAdresse **(**

client Personne**.**ID**%TYPE,**

id\_adresse Adresse**.**ID**%TYPE,**

p\_numero Adresse**.**NUMERO**%TYPE,**

p\_app Adresse**.**APP**%TYPE,**

p\_rue Adresse**.**RUE**%TYPE,**

p\_ville Adresse**.**VILLE**%TYPE,**

p\_codepostal Adresse**.**CODE\_POSTAL**%TYPE**

**)** **IS**

nbClient **INTEGER;**

ppays Adresse**.**PAYS**%TYPE;**

pprovince Adresse**.**PROVINCE**%TYPE;**

**BEGIN**

-- check if adress binded by 2 or more persons

-- if adress binded by 2 or more, create new one and point to the new

-- if not, modify directly the adress

-- Pour le présent on fait le minimum

**Select** **count(\*)** **into** nbClient **from** Adresse A**,** Adresse\_Personne AP **where** AP**.**Personne **=** client **AND** AP**.**Adresse **=** id\_adresse**;**

**if** **(**nbClient**=**1**)** **then**

**UPDATE** Adresse

**SET** NUMERO **=** **NVL(**p\_numero**,** NUMERO**),**

APP **=** **NVL(**p\_app**,** APP**),**

RUE **=** **NVL(**p\_rue**,** RUE**),**

VILLE **=** **NVL(**p\_ville**,** VILLE**),**

CODE\_POSTAL **=** **NVL(**p\_codepostal**,** CODE\_POSTAL**)**

**WHERE** ID **=** id\_adresse**;**

**else**

**Select** pays**,** province **into** ppays**,** pprovince **from** Adresse A **where** id\_adresse **=** A**.**id**;**

**INSERT** **INTO** Adresse**(**NUMERO**,** APP**,** RUE**,** VILLE**,** CODE\_POSTAL**,** PAYS**,** PROVINCE**)**

**VALUES** **(**p\_numero**,** p\_app**,** p\_rue**,** p\_ville**,** p\_codepostal**,** ppays**,** pprovince**);**

**UPDATE** Adresse\_Personne AP **SET** Adresse **=** **(select** **max(**id**)** **from** adresse**)** **WHERE** AP**.**adresse **=** id\_adresse **and** AP**.**Personne **=** client**;**

**end** **if;**

**END** ModifierAdresse**;**

Ici deux cas à distinguer, si l’adresse correspond à une seule personne ou à plusieurs. Si c’est une seule personne il suffit de faire un update… Sinon il faut ajouter une nouvelle ligne dans Adresse puis update Adresse\_Personne. Ce qui permet de ne pas modifier l’adresse de tout les clients.

## Conclusion

Travail qui permet d’approfondir la gestion des bases de données avec trigger, vues, et procédures ! On note que les procédures sont un peu (beaucoup) lourdes à debug car les erreurs d’oracle sont... assez étranges.