1. **Préparation de la fonction FUNCTION getDepData**

Créer des types de données personnalisés (**t\_dep\_table** et **t\_dep\_row**) qui seront utilisés dans la fonction **getDepData**

1. **Création du type t\_dep\_row** :

Script 1 :

**create or replace TYPE t\_dep\_row AS OBJECT(**

**NOM\_DEPARTEMENT);**

Ce script crée un type d'objet **t\_effectif\_dep** qui représente une ligne de données dans la table des DEPARTEMENTS. Chaque objet de ce type contient les attributs suivants :

**NOM\_DEPARTEMENT : type varchar2**

1. **Création du type t\_dep\_table** :

Script 2 :

**create or replace TYPE t\_dep\_table AS TABLE OF t\_dep\_row;**

Ce script crée un type de table **t\_dep\_table** qui est une table de type **t\_dep\_row**

**Cela permet de définir un type de variable qui peut contenir plusieurs lignes de données ; Chaque ligne étant représentée par le type t\_dep\_row**

**Cette table sera utilisée pour stocker les résultats de la fonction getDepData**

**En résumé, ces deux scripts définissent des types de données personnalisés qui seront utilisés dans la fonction getDepData pour manipuler et renvoyer des ensembles de données structurées.**

1. **Création de la fonction FUNCTION getDepData**

**Code :**

**CREATE OR REPLACE FUNCTION getDepData (p\_current\_user in varchar2) RETURN t\_dep\_table**

**AS**

**v\_result t\_dep\_table;**

**v\_dr number;**

**v\_current\_user varchar2(256):=trim(p\_current\_user);**

**BEGIN**

**begin**

**select dr\_id into v\_dr from gpi\_users where users\_gpi=v\_current\_user;**

**dbms\_output.put\_line('dir id=,' ||v\_dr)**

**exception when no\_data\_found then**

**Rollback;**

**end;**

**v\_result := t\_dep\_table();**

**if(v\_dr=1) then**

**for rec in (SELECT NOM\_DEPARTEMENT**

**FROM DEPARTEMENTS) loop**

**v\_result.extend; v\_result(v\_result.count):=t\_effectif\_row(rec.NOM\_DEPARTEMENT);**

**end loop ;**

**RETURN v\_result;**

**end if;**

**if(v\_dr !=1) then**

**for rec in (SELECT NOM\_DEPARTEMENT**

**FROM DEPARTEMENTS where dr\_id=(select dr\_id from gpi\_users where users\_gpi=v\_current\_user) ) loop**

**v\_result.extend; v\_result(v\_result.count):=t\_effectif\_row(rec.NOM\_DEPARTEMENT);**

**end loop ;**

**RETURN v\_result;**

**end if;**

**END getDepData;**

**create or replace FUNCTION getDepData(p\_current\_user in varchar2) RETURN t\_dep\_table**

**AS**

**v\_result t\_dep\_table;**

**v\_dr number;**

**v\_current\_user varchar2(256):=trim(p\_current\_user);**

**BEGIN**

**begin**

**select dr\_id into v\_dr from gpi\_users where users\_gpi=v\_current\_user;**

**dbms\_output.put\_line('dir id=,' ||v\_dr);**

**exception when no\_data\_found then**

**Rollback;**

**end;**

**v\_result := t\_dep\_table();**

**if(v\_dr=1) then**

**for rec in (SELECT**

**NOM\_DEPARTEMENT**

**FROM DEPARTEMENTS) loop**

**v\_result.extend;**

**v\_result(v\_result.count):=t\_dep\_row(rec.NOM\_DEPARTEMENT);**

**end loop ;**

**RETURN v\_result;**

**end if;**

**if(v\_dr !=1) then**

**for rec in (SELECT**

**NOM\_DEPARTEMENT**

**FROM DEPARTEMENTS where dr\_id=(select dr\_id from gpi\_users where users\_gpi=v\_current\_user) ) loop**

**v\_result.extend;**

**v\_result(v\_result.count):=t\_dep\_row(rec.NOM\_DEPARTEMENT);**

**end loop ;**

**RETURN v\_result;**

**end if;**

**END getDepData;**

**Explication :**

**En résumé, cette fonction récupère l'identifiant de direction d'un utilisateur, puis sélectionne les données de DEPARTEMENTS en fonction de cet identifiant, avant de les renvoyer sous forme d'un ensemble de données structurées de type t\_dep\_table.**

1. **Signature de la fonction:**
   * **La fonction s'appelle getDepData**
   * **Elle prend en paramètre un nom d'utilisateur actuel (p\_current\_user) de type varchar2.**
   * **Elle renvoie un ensemble de données de type t\_dep\_table**
2. **Déclaration des variables locales:**
   * **v\_result : Une variable de type t\_dep\_table pour stocker les résultats de la fonction.**
   * **v\_dr : Une variable de type number pour stocker l'identifiant de direction (dr\_id).**
   * **v\_current\_user : Une variable de type varchar2(256) pour stocker le nom d'utilisateur actuel, après suppression des espaces blancs.**
3. **Bloc exécutable:**
   * **Le bloc BEGIN marque le début de la section exécutable de la fonction.**
4. **Bloc de traitement de l'exception:**
   * **Le bloc begin imbriqué est utilisé pour traiter l'exception no\_data\_found qui peut se produire lors de la sélection de l'identifiant de direction.**
   * **Si aucune donnée n'est trouvée, la fonction effectue un rollback.**
5. **Initialisation de v\_result:**
   * **La variable v\_result est initialisée en tant qu'ensemble de données vide de type t\_dep\_table**
6. **Traitement des données en fonction de v\_dr:**
   * **Si v\_dr est égal à 1, la fonction sélectionne toutes les lignes de la table DEPARTEMENTS et les ajoute à v\_result**
   * **Sinon, la fonction sélectionne les lignes de la table DEPARTEMENTS où l'identifiant de direction correspond à celui de l'utilisateur actuel, puis les ajoute à v\_result.**
7. **Retour des résultats:**
   * **Les résultats sont renvoyés à la fin de chaque bloc de condition.**
8. **Fin de la fonction:**
   * **END getDepData; marque la fin de la définition de la fonction.**

select CODE\_DR c1, DR\_ID c2 from table(getDirData(:APP\_USER))