

- h- Donner un code PL/SQL amélioré qui prends en considération cette erreur dans la zone des exceptions. Utiliser la clause : « **When TOO_MANY_ROWS Then** ».
- i- Donner un code PL/SQL amélioré qui prends en considération le cas où la requête « **SELECT** » ne retourne aucun résultat. Utiliser la clause : « **When NO_DATA_FOUND Then** » dans la zone des exceptions.
- j- On peut utiliser aussi la clause : « **When others Then** » dans la zone des exceptions pour capter une certaine erreur et ensuite tester son code : **SQLCODE** pour pouvoir personnaliser le message d'erreur. Donner un code PL/SQL qui teste selon le code de l'erreur en utilisant : « **When others Then** ».
- k- La notation **%ROWTYPE** est un type qui permet de déclarer un schéma de relation. Dans ce cas, on n'aura pas besoin de déclarer des variables supplémentaires de type **%TYPE**. Refaire la question « c » en utilisant la notation **%ROWTYPE**
- l- Les curseurs permettent de gérer les requêtes SQL qui retournent plusieurs lignes. Donner un code PL/SQL qui affiche le nom, le salaire et l'adresse de tous les employés en utilisant un curseur.
- m- En utilisant un curseur, refaire la question : « f » pour calculer par exemple le salaire maximum de tous les employés sans utiliser la fonction « MAX » de SQL. (Idem pour calculer le salaire maximum au sein du département : 'Dep2').
- n- Refaire la question précédente en utilisant cette fois ci la boucle : « While » puis la boucle : « FOR ». (Penser à utiliser « **%FOUND** » pour itérer tant qu'un tuple est trouvé (par fetch))

Quelques remarques :

- Absence de : open et close
- Pas besoin de : fetch, ni au début ni en cours de boucle non plus à la condition d'arrêt.
- La variable de contrôle « **EmployEnr** » est déclarée automatiquement du type du schéma de résultat de select.