

Mapping objet-relationnel : Réalisation de codes avec mêmes résultats requête SQL

Cahier des charges

Réaliser du code qui donne les mêmes résultats de 2 requêtes d'interrogation données en SQL

Faire du mapping objet-relationnel

Utilisation du langage **LINQ** sur C#

Accéder aux données, si utilisation de requêtes d'interrogation

Seules clauses autorisées :

SELECT, FROM, JOIN, WHERE et ORDER BY

Clauses non autorisées :

GROUP BY, HAVING , fonctions de fenêtrage, etc

Affichage des résultats au choix

Résultat de la mission

```
private void HandleSecondQuery()
{
    //récupération du résultat de la requête
    var students = from c in HTS_SQL.T_Classe_CLS
                   join e in HTS_SQL.T_Eleve_ELV on c.CLS_Num equals e.ELV_REF_CLS
                   join r in HTS_SQL.T_Recompense_RCP on e.ELV_Id equals r.RCP_REF_ELV
                   where c.CLS_Intit.Equals("Hadron") && e.ELV_Id >= 837
                   orderby e.ELV_Id descending, r.RCP_Annee ascending
                   select new
                   {
                       e.ELV_Id,
                       e.ELV_Nom,
                       r.RCP_Annee
                   };

    //on convertit student en liste
    students.ToList();
    int numLigne = 1;
    int Rg_id = 1;
    int Rg_Annee_Par_id = 1;
    //on crée une liste d'élève vide
    List<decimal> idElvs = new List<decimal>();
    //on parcourt chaque étudiant
    foreach (var s in students)
    {
        //chaque élève ne rentre qu'une fois dans la condition
        if (!idElvs.Contains(s.ELV_Id))
        {
            //on l'ajoute dans la liste (pour actualiser la verification)
            idElvs.Add(s.ELV_Id);
            //on donne un rang à l'élève (qui sera identique pour chaque occurrence de cet élève
            Rg_id = numLigne;
            //on donne un rang à l'année ( c'est à dire que chaque ligne de l'élève aura un rang
            Rg_Annee_Par_id = 1;
        }
    }
}
```

Requete1

Requete2

Méthodes

Utilisation papier pour les algorithmes

Faire des recherches sur des clauses (par exemple, ROLLUP) pour mieux comprendre leur fonctionnement

Compétences acquises

Savoir comprendre un résultat d'une requête d'interrogation en SQL et de faire son algorithme sur C# avec LINQ pour retrouver les mêmes résultats

Progression sur l'utilisation de LINQ pour les requêtes SQL

Technologie
(langages et application)

