

## SAÉ S3.4 associée à la ressource R3.07 SQL dans un langage de programmation

Ce travail est à effectuer en binôme d'étudiants du même groupe ; si l'effectif de votre groupe est impair, votre enseignant de TD choisira entre un monôme ou un trinôme ou un binôme avec un étudiant d'un autre groupe.

À remettre en déposant, au plus tard le vendredi 18 novembre 2022 à 23h59, sur <https://moodle1.u-bordeaux.fr/mod/assign/view.php?id=586237>, un unique fichier archive au format  
S3<groupe>\_<nom 1<sup>er</sup> étudiant>\_<nom 2<sup>nd</sup> étudiant>.<extension>.

L'évaluation tiendra compte de la qualité de tout votre travail, et notamment du code (commentaires, mnémonicité, indentation, style, respect de normes de programmation, simplicité, efficacité voire complexité, etc.).

Vous aurez peut-être à faire une démonstration à votre enseignant de TD.

Vous allez utiliser une base de données sur laquelle vous aviez réalisé une série de requêtes d'interrogation SQL l'an passé en S2 dans la ressource R2.06 Exploitation d'une base de données. Dans le cadre de la SAÉ S3.4, vous devez cette fois-ci réécrire deux de ces requêtes (adaptées et simplifiées), mais en faisant du « mapping » objet-relationnel (*object-relational mapping*, *ORM*) tel que présenté cette année dans la ressource R3.07 SQL dans un langage de programmation.

La base de données « HTS+ », l'école des Hautes Techniques Supérieures (plus), gère des classes (n° et intitulé), des élèves (identifiant, élève parrain ou marraine, classe, nom, année et ville de naissance), des récompenses (élève et année) et des épreuves (élève, abréviation et note). Voici son schéma relationnel en compréhension :

Les schémas des relations (c.-à-d. tables) :

```
T_Classe_CLS ( CLS_Num , CLS_Intit )
T_Eleve_ELV ( ELV_Id , ELV_REF_ELV , ELV_REF_CLS , ELV_Nom , ELV_AnneeNais ,
              ELV_VilleNais )
T_Recompense_RCP ( RCP_REF_ELV , RCP_Annee )
T_Epreuve_EPR ( EPR_REF_ELV , EPR_Abrv , EPR_Note )
```

Les contraintes de clé (primaire) :

CLS\_Num est la clé de T\_Classe\_CLS, contrainte nommée C\_PK\_CLS  
ELV\_Id est la clé de T\_Eleve\_ELV, contrainte nommée C\_PK\_ELV  
( RCP\_REF\_ELV , RCP\_Annee ) est la clé de T\_Recompense\_RCP, contrainte nommée C\_PK\_RCP  
( EPR\_REF\_ELV , EPR\_Abrv ) est la clé de T\_Epreuve\_EPR, contrainte nommée C\_PK\_EPR

Les contraintes d'intégrité référentielles (c.-à-d. l'ensemble des valeurs de la clé étrangère est inclus dans l'ensemble des valeurs de la clé primaire) :

ELV\_REF\_ELV référence ELV\_Id, contrainte nommée C\_FK\_ELV\_ELV  
ELV\_REF\_CLS référence CLS\_Num, contrainte nommée C\_FK\_ELV\_CLS  
RCP\_REF\_ELV référence ELV\_Id, contrainte nommée C\_FK\_RCP\_ELV  
EPR\_REF\_ELV référence ELV\_Id, contrainte nommée C\_FK\_EPR\_ELV

Les contraintes existentielles :

Les seuls attributs facultatifs sont : ELV\_REF\_ELV, ELV\_VilleNais et EPR\_Note

Tous les autres attributs sont donc obligatoires : CLS\_Num, CLS\_Intit, ELV\_Id, ELV\_REF\_CLS, ELV\_Nom, ELV\_AnneeNais, RCP\_REF\_ELV, RCP\_Annee, EPR\_REF\_ELV et EPR\_Abrv contraintes nommées respectivement C\_NOT\_NULL\_CLS\_Num, C\_NOT\_NULL\_CLS\_Intit, C\_NOT\_NULL\_ELV\_Id, C\_NOT\_NULL\_ELV\_REF\_CLS, C\_NOT\_NULL\_ELV\_Nom, C\_NOT\_NULL\_ELV\_AnneeNais, C\_NOT\_NULL\_RCP\_REF\_ELV, C\_NOT\_NULL\_RCP\_Annee, C\_NOT\_NULL\_EPR\_REF\_ELV et C\_NOT\_NULL\_EPR\_Abrv

Les contraintes d'unicité :

CLS\_Intit est unique, contrainte nommée C\_UNI\_CLS\_Intit

Les contraintes de domaine :

L'attribut EPR\_Note, s'il est renseigné, doit être compris entre 0 et 20, contrainte nommée C\_CHK\_EPR\_Note

La base de données est gérée par SQL Server sur le serveur info-dormeur et/ou info-joyeux, et vous pouvez y accéder avec le compte utilisateur de connexion ETD ; vous pouvez aussi récupérer les scripts de création sur <https://moodle1.u-bordeaux.fr/mod/resource/view.php?id=583480>. Il y a 23 classes, 610 élèves, 1745 récompenses et 2908 épreuves.

On considère les deux requêtes d'interrogation SQL suivantes :

- Afficher pour chaque élève de n° inférieur à 42 et pour chaque épreuve de contrôle continu de n° 1 à 3, la moyenne des notes et le nombre de notes, ainsi que certaines lignes supplémentaires de cumuls (par élève et épreuve, par élève, et pour tous les élèves).

```
SELECT EPR_REF_ELV , EPR_Abrv , AVG(EPR_Note) Moy_Epr_Note , COUNT(*) Nb_Epr_Elv
FROM T_Epreuve_EPR
-- sans JOIN T_Eleve_ELV ON ELV_Id = EPR_REF_ELV
WHERE EPR_REF_ELV < 42 AND EPR_Abrv BETWEEN 'CC1' AND 'CC3'
GROUP BY ROLLUP( EPR_REF_ELV , EPR_Abrv )
ORDER BY EPR_REF_ELV DESC , EPR_Abrv DESC
```

	EPR_REF_ELV	EPR_Abrv	Moy_Epr_Note	Nb_Epr_Elv
1	35	CC1	NULL	1
2	35	NULL	NULL	1
3	31	CC3	13.000000	1
4	31	NULL	13.000000	1
5	30	CC3	12.000000	1
6	30	CC1	12.500000	1
7	30	NULL	12.250000	2
8	29	CC3	7.500000	1
9	29	NULL	7.500000	1
10	28	CC3	NULL	1
11	28	CC2	17.500000	1
12	28	NULL	17.500000	2
13	26	CC1	12.000000	1
14	26	NULL	12.000000	1
15	20	CC2	16.000000	1
16	20	CC1	8.000000	1
17	20	NULL	12.000000	2
18	18	CC3	4.500000	1
19	18	NULL	4.500000	1
20	15	CC2	18.500000	1
21	15	NULL	18.500000	1
22	10	CC3	9.000000	1
23	10	NULL	9.000000	1
24	NULL	NULL	11.863636	13

dont voici le résultat :

- Afficher l'identifiant et le nom de l'élève, un n° de ligne (unique et séquentiel) et le rang pour l'ordre des identifiants des élèves, l'année de récompense et le numéro d'ordre de la récompense pour chaque élève. Les élèves sont dans la classe Hadron et ont un identifiant au moins égal à 837. Le résultat est trié par identifiant d'élève et année de récompense.

```
SELECT ELV_Id ,
       ELV_Nom ,
       ROW_NUMBER() OVER(ORDER BY ELV_Id DESC) Num_Ligne,
       RANK() OVER(ORDER BY ELV_Id DESC) Rg_Id ,
       RCP_Annee ,
       RANK() OVER(PARTITION BY ELV_Id ORDER BY RCP_Annee) Rg_Annee_par_Id
FROM T_Classe_CLS
JOIN T_Eleve_ELV ON CLS_Num = ELV_REF_CLS
JOIN T_Recompense_RCP ON ELV_Id = RCP_REF_ELV
WHERE CLS_Intit = 'Hadron' AND ELV_Id >= 837
ORDER BY ELV_Id DESC , RCP_Annee
```

	ELV_Id	ELV_Nom	Num_Ligne	Rg_Id	RCP_Annee	Rg_Annee_par_Id
1	988	Sergent	1	1	2004	1
2	988	Sergent	2	1	2009	2
3	968	Hutt	3	3	2005	1
4	968	Hutt	4	3	2006	2
5	968	Hutt	5	3	2010	3
6	968	Hutt	6	3	2011	4
7	955	Temis	7	7	2005	1
8	955	Temis	8	7	2006	2
9	955	Temis	9	7	2010	3
10	955	Temis	10	7	2011	4
11	950	Hyatt	11	11	2004	1
12	950	Hyatt	12	11	2005	2
13	950	Hyatt	13	11	2008	3
14	950	Hyatt	14	11	2009	4
15	950	Hyatt	15	11	2010	5
16	889	Travers	16	16	2006	1
17	889	Travers	17	16	2011	2
18	852	Halbert	18	18	2006	1
19	852	Halbert	19	18	2009	2
20	848	Travers	20	20	2004	1
21	848	Travers	21	20	2010	2

dont voici le résultat :

Les résultats des deux requêtes sont triés de façon cohérente avec tous les calculs (cumuls et fonctions de fenêtrage) à effectuer, de sorte à avoir une complexité raisonnable et des performances correctes ainsi que des algorithmes nettement simplifiés. De plus, même si un système de gestion de base de données relationnel manipule de très gros volumes de données stockées en mémoire auxiliaire, on pourra ici supposer que vos traitements pourront tous être exécutés en mémoire principale.

Le livrable doit contenir toute votre application, et en particulier tout le code réalisé pour effectuer ces deux requêtes d'interrogation en faisant du « mapping » objet-relationnel. Pour accéder aux données, si vous devez utiliser une requête d'interrogation SQL, seules les clauses SELECT, FROM, JOIN, WHERE et ORDER BY sont autorisées tandis que les clauses GROUP BY et HAVING (et donc CUBE, GROUPING SETS et ROLLUP) et fonctions de fenêtrage (ROW\_NUMBER, RANK et donc OVER()) sont interdites.