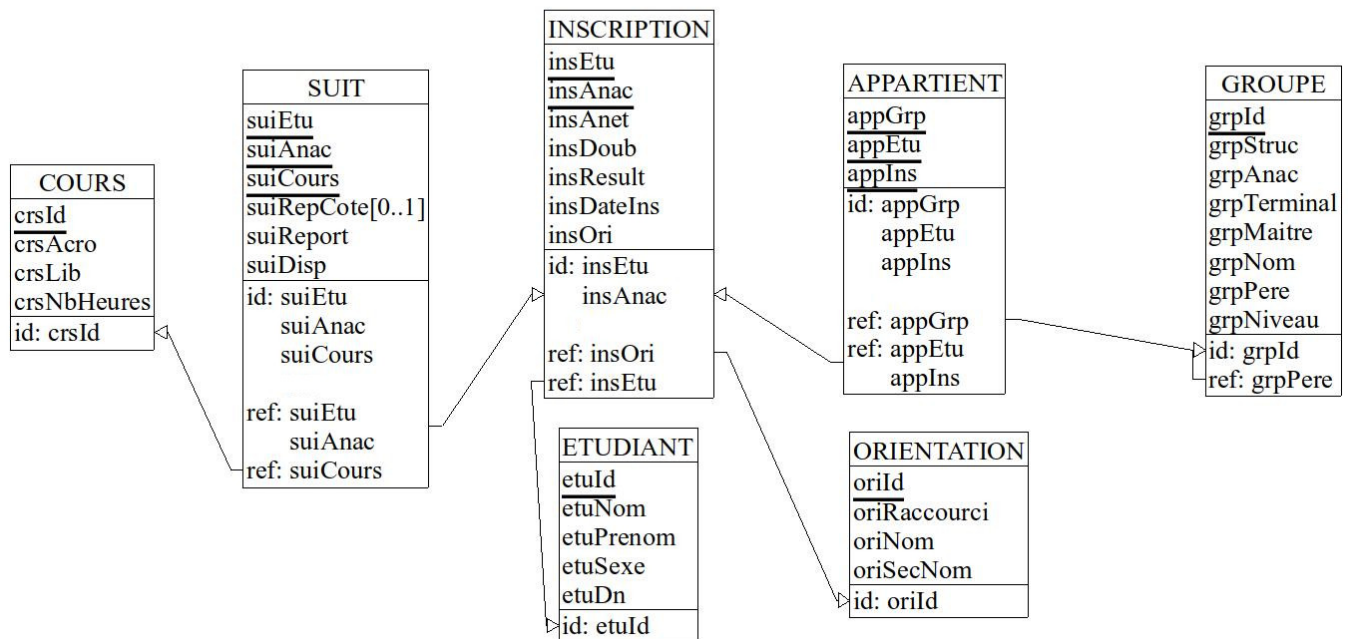


Labo 9 — Sous-requêtes (séance 2) — Solutions

Schéma

Voici le schéma conceptuel de base de données pour les inscriptions d'étudiants à une haute-école.

HauteEcole/1



Etudiant

- **etuId** : Numéro de l'étudiant – Entier, unique.
- **etuNom** : Nom de l'étudiant – Chaîne de 50 caractères.
- **etuPrenom** : Prénom de l'étudiant – Chaîne de 50 caractères.
- **etuSexe** : Sexe de l'étudiant – Un caractère : 'M' ou 'F'.
- **etuDn** : Date de naissance de l'étudiant – Date.

Inscription

- **insEtu** : Numéro de l'étudiant ayant pris l'inscription – Entier.
- **insAnac** : Année académique de l'inscription – Entier (2003 pour 2003-2004).
- **insAnet** : Année d'étude de l'inscription – Entier.
- **insDoub** : L'étudiant est-il doubleur dans cette orientation et cette année d'étude ? — Entier (1 ou 0 pour vrai ou faux).
- **insResult** : Code du résultat de l'année – Entier (0 réussi, 1 raté, -1 pour absence de valeur)
- **insDateIns** : Date d'inscription – Chaîne de 30 caractères au format JJ/MM/AAAA.
- **insOri** : Numéro de l'orientation dans laquelle l'inscription est faite – Entier.

Suit

- **suiEtu** : Numéro d'étudiant – Entier.
- **suiAnac** : Année académique de l'inscription – Entier.
- **suiCours** : Numéro du cours – Entier.
- **suiRepCote** : Dans le cas où suiReport est vrai, ceci est la note reportée pour le cours sinon NULL – Entier.
- **suiReport** : L'étudiant bénéficie-t-il d'un report pour ce cours ? – Entier, 1 ou 0 pour vrai ou faux.
- **suiDisp** : L'étudiant bénéficie-t-il d'une dispenses pour ce cours ? – Entier, 1 ou 0 pour vrai ou faux.

Cours

- **crsId** : Numéro du cours – Entier, unique.
- **crsAcro** : Acronyme du cours – Chaîne de 6 caractères, unique.
- **crsLib** : Intitule du cours – Chaîne de 200 caractères.
- **crsNbHeures** : Nombre d'heures du cursus de cette année – Entier.

Orientation

- **orId** : Numéro de l'orientation – Entier, unique.
- **oriRaccourci** : Acronyme de l'orientation – Chaîne de 4 caractères, unique.
- **oriNom** : Libellé de l'orientation – Chaîne de 50 caractères.
- **oriSecNom** : Libellé de la section chapeautant l'orientation – Chaîne de 50 caractères.

Appartient

- **appGrp** : Numéro de groupe – Entier.
- **appEtu** : Numéro de l'étudiant – Entier.
- **appIns** : Année académique de l'inscription – Entier.

Groupe

- **grpId** : Numéro du groupe – Entier, unique.
- **grpStruc** : Nom de la structure à laquelle appartient le groupe – Chaîne de 6 caractères.
- **grpAnac** : Année académique d'utilisation du groupe – Entier, 2003 pour 2003-2004.
- **grpTerminal** : Le groupe est-il terminal ? – Entier, 1 ou 0 pour vrai ou faux
- **grpMaitre** : Le groupe est-il racine de l'arbre de la structure ? – Entier, 1 ou 0 pour vrai ou faux.
- **grpNom** : Nom du groupe dans cette structure – Chaîne de 6 caractères.
- **grpPere** : Numéro du groupe père dans la structure – Entier, NULL si ce groupe est un groupe maître.
- **grpNiveau** : Niveau du groupe dans la structure – Entier.

Exercices

Répondez aux questions suivantes par une requête SQL.

1. Affichez le numéro, le nom et le prénom des étudiants qui n'ont jamais doublé.

```
SELECT etuId, etuNom, etuPrenom
FROM ETUDIANT
WHERE etuId IN (
    SELECT insEtu
```

```
FROM INSCRIPTION
GROUP BY insEtu
HAVING SUM(insDoub) = 0
);
```

ou mieux

```
SELECT etuId, etuNom, etuPrenom
FROM ETUDIANT JOIN INSCRIPTION ON insEtu = etuId
GROUP BY etuId, etuNom, etuPrenom
HAVING SUM(insDoub) = 0 ) ;
```

2. Affichez toutes les informations des cours qu'aucun étudiant n'a suivi durant l'année académique 2020-2021.

```
SELECT *
FROM COURS
WHERE crsId NOT IN (
    SELECT DISTINCT suiCours
    FROM SUI
    suiAnac = '2020'
);
```

3. Affichez les informations des groupes de l'année académique où il y a eu le plus de dispenses.

```
SELECT *
FROM GROUPE
WHERE grpAnac IN (
    SELECT suiAnac
    FROM SUI
    GROUP BY suiAnac
    HAVING SUM(suiDisp) >= ALL(
        SELECT SUM(suiDisp)
        FROM SUI
        GROUP BY suiAnac
    )
);
```

4. Affichez la liste ordonnée des acronymes des cours ayant plus d'heures que la moyenne des cours.

```
SELECT crsAcro
FROM COURS
WHERE crsNbHeures > (
    SELECT AVG(crsNbHeures)
    FROM COURS
);
```

```
)  
ORDER BY crsAcro;
```

5. Donnez la liste des étudiants, ordonnée sur le nom et le prénom, qui sont dans les plus petits groupes de l'année académique 2020-2021.

```
SELECT etuId, etuNom, etuPrenom  
FROM ETUDIANT  
JOIN INSCRIPTION ON insEtu = etuId  
JOIN APPARTIENT ON insEtu = appEtu AND insAnac = appIns  
WHERE appIns = '2020'  
GROUP BY appGrp  
HAVING COUNT(*) <= ANY(  
    SELECT COUNT(*)  
    FROM APPARTIENT  
    WHERE appIns = '2020'  
    GROUP BY appGrp  
)  
ORDER BY etuNom, etuPrenom
```

6. Donnez la liste des acronymes des cours ordonnée sur le nombre de reports entre 2010 et 2020.

```
SELECT crsAcro, COUNT(suiRepCote) -- ou SUM(suiReport)  
FROM COURS JOIN SUI ON crsId = suiCours  
WHERE suiAnac BETWEEN 2010 AND 2020  
GROUP BY crsAcro  
ORDER BY COUNT(suiRepCote);
```

7. Donnez la liste des groupes ayant plus de filles que de garçons.

```
SELECT groupe.appGrp  
FROM APPARTIENT groupe  
JOIN INSCRIPTION ON insEtu = groupe.appEtu AND insAnac = groupe.appIns  
JOIN ETUDIANT ON etuId = insEtu  
WHERE etuSexe = 'F'  
GROUP BY groupe.appGrp  
HAVING COUNT(*) > (  
    SELECT COUNT(*)  
    FROM APPARTIENT  
    JOIN INSCRIPTION ON insEtu = appEtu AND insAnac = appIns  
    JOIN ETUDIANT ON etuId = insEtu  
    WHERE etuSexe = 'M' AND groupe.appGrp = appGrp  
)
```

8. Donnez la liste des nombres d'inscriptions ordonnée par année académique et par orientation.

```
SELECT insAnac, insOri, count(*) nbreEtudiants  
FROM INSCRIPTION  
GROUP BY insAnac, insOri  
ORDER BY insAnac, insOri
```