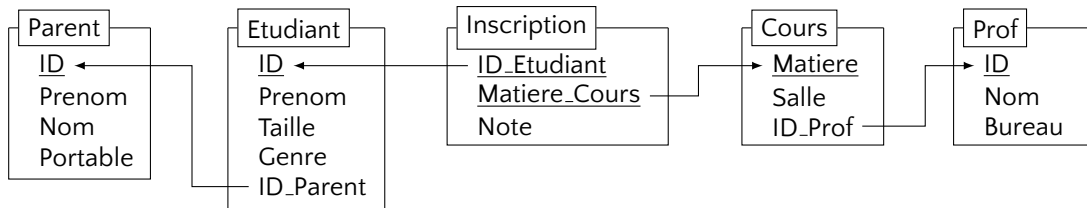


SQL 2018 (L1 SoQ)

CM 4 : Requêtes de groupage et d'agrégation

La base de données fac.db est disponible sur la page du cours¹. Son schéma est présenté ci-dessous.



Requête 1. Créer une vue qui complète les informations sur les étudiants dans la table Etudiant avec le nom de famille (pris de la table Parent).

```
CREATE VIEW EtudiantNom AS
SELECT Etudiant.ID, Etudiant.Prenom, Parent.Nom, Etudiant.Taille, Etudiant.ID_Parent
FROM Etudiant LEFT OUTER JOIN Parent ON (Etudiant.ID_Parent = Parent.ID);
```

Requête 2. Afficher la taille moyenne pour tout sexe d'étudiant

```
SELECT Genre, AVG(Taille)
FROM Etudiant
GROUP BY Genre;
```

Requête 3. Afficher la taille moyenne, la taille minimale ainsi que la taille maximale pour tout sexe d'étudiant

```
SELECT Genre, AVG(Taille), MIN(Taille), MAX(Taille)
FROM Etudiant
GROUP BY Genre;
```

Requête 4. Afficher la moyenne des notes de tout étudiant

```
SELECT ID_Etudiant, AVG(Note)
FROM Inscription
GROUP BY ID_Etudiant;
```

Requête 5. Idem + ordonner les résultats par la moyenne (l'ordre décroissant)

```
SELECT ID_Etudiant, AVG(Note) AS Moyenne
FROM Inscription
GROUP BY ID_Etudiant
ORDER BY Moyenne DESC;
```

Requête 6. Pour tout étudiant afficher sa moyenne ainsi que le nombre de cours auxquels il est inscrits et le nombre des notes qui sont renseignés.

```
SELECT ID_Etudiant, AVG(Note), COUNT(*) AS Inscriptions, COUNT(Note) AS Notes
FROM Inscription
GROUP BY ID_Etudiant;
```

¹ <http://researchers.lille.inria.fr/~staworko/sql18.html>

Requête 7. Afficher les moyennes des étudiants dont toutes les notes sont renseignés.

```
SELECT ID_Etudiant, AVG(Note)
FROM Inscription
GROUP BY ID_Etudiant
HAVING COUNT(*) = COUNT(Note);
```

Requête 8. Pour tout étudiant afficher son prénom, son nom de famille (s'il est renseigné) et sa moyenne

```
SELECT Etudiant.Prenom, Parent.Nom, AVG(Note) AS Moyenne
FROM Etudiant LEFT OUTER JOIN Parent ON (Etudiant.ID_Parent = Parent.ID)
JOIN Inscription ON (Etudiant.ID = Inscription.ID_Etudiant)
GROUP BY Etudiant.Prenom, Parent.Nom;
```

ou plus simplement en utilisant la vue EtudiantNom

```
SELECT EtudiantNom.Prenom, EtudiantNom.Nom, AVG(Note) AS Moyenne
FROM EtudiantNom
JOIN Inscription ON (EtudiantNom.ID = Inscription.ID_Etudiant)
GROUP BY EtudiantNom.Prenom, EtudiantNom.Nom
ORDER BY Moyenne DESC;
```

Requête 9. Calculer la moyenne de tout cours

```
SELECT Matiere_Cours, AVG(Note)
FROM Inscription
GROUP BY Matiere_Cours;
```

Requête 10. Calculer la statistique qui compare les moyennes des étudiants et des étudiantes pour tous les cours.

```
SELECT Matiere_Cours, Genre, AVG(Note)
FROM Inscription
JOIN Etudiant ON (Etudiant.ID = Inscription.ID_Etudiant)
GROUP BY Matiere_Cours, Genre
ORDER BY Matiere_Cours, Genre;
```

Requête 11. Pour tout professeur compter le nombre des cours qu'il assure.

```
SELECT Prof.Nom, COUNT(Cours.Matiere)
FROM PROF
LEFT OUTER JOIN Cours ON (Prof.ID = Cours.ID_Prof)
GROUP BY Nom;
```

Requête 12. Pour tout professeur calculer le nombre d'étudiant qu'il encadre lors de ces cours (si un professeur encadre le même étudiant aux plusieurs cours, cet étudiant doit être compté uniquement une fois).

```
SELECT Prof.NOM, COUNT(S.ID_Etudiant)
FROM Prof
LEFT OUTER JOIN (
    SELECT DISTINCT Cours.ID_Prof, Inscription.ID_Etudiant
    FROM Cours
    JOIN Inscription ON (Cours.Matiere = Inscription.Matiere_Cours)
) AS S ON (Prof.ID = S.ID_Prof)
GROUP BY Prof.Nom;
```

ou sans sous-expressions

```
SELECT Prof.NOM, COUNT(DISTINCT Inscription.ID_Etudiant)
FROM Prof
LEFT OUTER JOIN Cours ON (Prof.ID = Cours.ID_Prof)
LEFT OUTER JOIN Inscription ON (Cours.Matiere = Inscription.Matiere_Cours)
GROUP BY Prof.Nom;
```