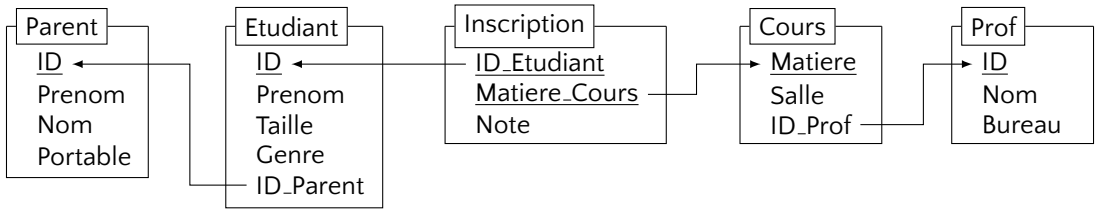
Bases de données et SQL (L1 SoQ et L3 MIASHS)

CM 4 : Requêtes de groupage et d'agrégation

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



Les contenus des tables sont suivants.

Etudiant

ID	Prenom	Taille	Genre	ID_Parent
1	Jean	178	H	1
3	Jean	162	H	3
4	Marie	159	H	1
5	Paul	161	F	2
6	Luc	161	H	NULL
8	Marion	159	F	4

Parent

ID	Prenom	Nom	Portable
1	Bruno	Dubois	06.14.21.56.34
2	Constance	Dupont	06.41.21.32.14
3	Adèle	Martin	06.84.81.96.12

Cours

Matiere	Salle	ID_Prof
SQL	B2.461	11
HTML	A2.061	46
IA	A1.423	11

Prof

ID	Nom	Bureau
11	Sławek	D.42
46	Fabien	C.21
57	Marc	D.42

Inscription

ID_Etudiant	Matiere_Cours	Note
1	SQL	12.0
1	HTML	12.0
1	IA	NULL
3	SQL	15.0
4	SQL	16.0
4	HTML	17.0
4	IA	12.0
6	SQL	11.0
6	IA	NULL
8	HTML	16.0
8	IA	NULL

Q1 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.Genre, Etudiant.ID_Parent
  FROM Etudiant LEFT OUTER JOIN Parent ON (Etudiant.ID_Parent = Parent.ID);
```

EtudiantNom

ID	Prenom	Nom	Taille	Genre	ID_Parent
1	Jean	Dubois	178	H	1
3	Jean	Martin	162	H	3
4	Marie	Dubois	159	H	1
5	Paul	Dupont	161	F	2
6	Luc	NULL	161	H	NULL
8	Marion	NULL	159	F	4

Q2 Afficher la taille moyenne pour tout sexe d'étudiant

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

Genre	AVG(Taille)
F	160.0
H	165.0

Q3 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;
```

Genre	AVG(Taille)	MIN(Taille)	MAX(Taille)
F	160.0	159	161
H	165.0	159	178

Q4 Afficher la moyenne des notes de tout étudiant

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

ID_Etudiant	AVG(Note)
1	12.0
3	15.0
4	15.0
6	11.0
8	16.0

Q5 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;
```

ID_Etudiant	Moyenne
8	16.0
3	15.0
4	15.0
1	12.0
6	11.0

Q6 Pour tout étudiant afficher sa moyenne ainsi que le nombre de cours auquels 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;
```

ID_Etudiant	AVG(Note)	Inscriptions	Notes
1	12.0	3	2
3	15.0	1	1
4	15.0	3	3
6	11.0	2	1
8	16.0	2	1

Q7 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);
```

ID_Etudiant	AVG(Note)
3	15.0
4	15.0

Q8 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;
```

Prenom	Nom	Moyenne
Marion	NULL	16.0
Jean	Martin	15.0
Marie	Dubois	15.0
Jean	Dubois	12.0
Luc	NULL	11.0

Q9 Calculer la moyenne de tout cours

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

Matiere_Cours	AVG(Note)
HTML	15.0
IA	12.0
SQL	13.5

Q10 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;
```

Matiere_Cours	Genre	AVG(Note)
HTML	F	16.0
HTML	H	14.5
IA	F	NULL
IA	H	12.0
SQL	H	13.5

Q11 Pour tout professeur compter le nombre des cours qu'il assure.

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

Nom	Cours Enseignes
Fabien	1
Marc	0
Sławek	2

Q12 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) AS "Etudiants Encadres"
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) AS "Etudiants Encadres"
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;
```

Nom	Etudiants Encadres
Fabien	3
Marc	0
Sławek	5