
TD n°13 : Requêtes en langage SQL



3 heures

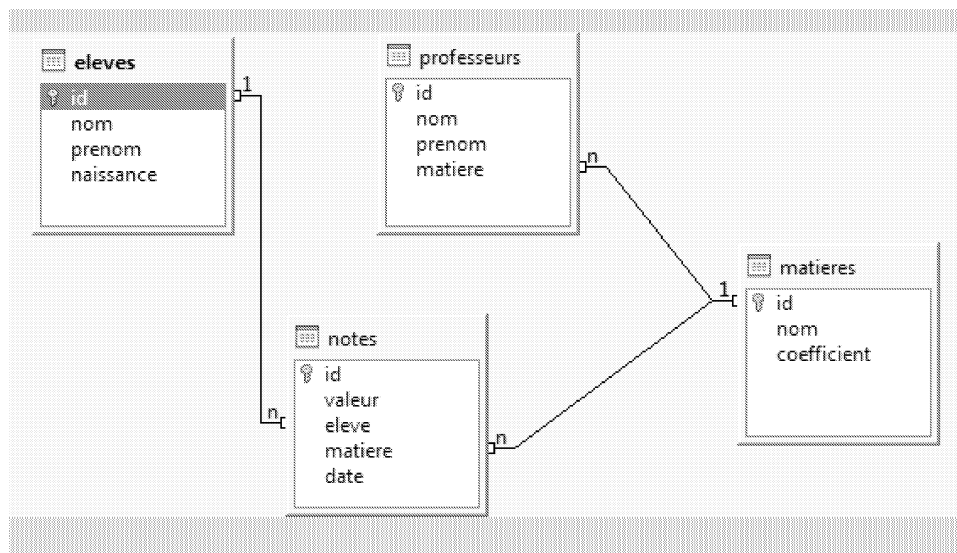
Rédigé par Pascal Delahaye

25 mars 2017

1 Introduction

Le but de ce TD est de revoir de façon progressive les différentes façons de formuler des requêtes en SQL. Nous allons ici travailler sur une base de données d'un établissement scolaire.

Commencez par ouvrir le fichier "notes.odt" contenant la base de données qui se trouve sur le bureau de votre ordinateur.



Modèle relationnel de la base de données étudiée

Rappelons la structure de base d'une requête SQL :

```
SQL
SELECT nom des différents attributs souhaités
FROM nom de la table nécessaire à la requête
WHERE condition(s) sur la ou lesquelles porte la sélection des enregistrements
(AND, OR, =, <=, <>, ...)
```

Rappelons également les opérations ensemblistes nécessaires à l'élaboration de certaines requêtes complexes :

UNION ou UNION ALL : Pour réunir deux ensembles d'enregistrements
 INTERSECT : Pour prendre l'intersection de deux ensembles d'enregistrements
 EXCEPT ou NOT IN : Pour éliminer certains éléments d'une ensemble

A ces différentes instructions, s'ajoutent :

ORDER BY : Pour ordonner les enregistrements sélectionnés selon un ou plusieurs attributs (tout à la fin)
 DISTINCT : Pour éliminer les doublons éventuels (juste après SELECT)
 -- : A mettre en début de ligne pour commenter vos instructions

Pour les jointures, il faut ajouter les différents liens entre les tables utilisées dans les conditions de sélections.

SQL

Jointure directe :

```
SELECT nom des différents attributs souhaités
FROM Table1 JOIN Table2 ON égalité(s) entre attributs assurant la jointure
WHERE condition(s) diverse(s)
```

Jointure indirecte :

```
SELECT nom des différents attributs souhaités
FROM Table JOIN Table1 ON égalité(s) entre attributs assurant la jointure 1
      JOIN Table2 ON égalité(s) entre attributs assurant la jointure 2
WHERE condition(s) diverse(s)
```

2 Requêtes simples

Commençons pour nous échauffer par des requêtes de base...

A] Projection simple

Formuler en langage SQL les requêtes suivantes et observer le résultat obtenu.

1. Donner la liste des noms et prénoms des élèves.
2. Donner la liste des noms des enseignants et l'identifiant de la matière enseignée.

B] Selection simple

Formuler en langage SQL les requêtes suivantes et observer le résultat obtenu.

1. Donner les enregistrements de la table "notes" qui ont eu la note 20 dans la matière numéro 2.
2. Donner les enregistrements de la table "matières" dont le coefficient est compris entre 2 et 3.

C] Selection et projection

Formuler en langage SQL les requêtes suivantes et observer le résultat obtenu.

1. Donner la liste des élèves (uniquement noms et prénoms) nés en 1994.
2. Donner la liste des identifiants des élèves qui ont eu la note 20 au moins une fois à leur devoir.
Éliminer les doublons.

D] Jointure, selection et projection

Formuler en langage SQL les requêtes suivantes et observer le résultat obtenu.

1. Donner la liste des matières suivies par l'élève Delors (cad les matières pour lesquelles il a été noté).

2. Donner la liste des élèves ayant passé l'examen de Math
3. Donner la liste des professeurs (noms) dont la matière enseignée à un coefficient inférieur ou égal à 3, en précisant la matière correspondante.
4. Donner la liste des élèves qui n'ont pas eu la moyenne dans la matière de M. Aloui.
5. Donner la liste des élèves qui ont eu une note supérieure à 15 en math.
6. Donner la liste ordonnée des élèves (noms et prénoms) qui ont eu au moins une fois 20 en devoir, en précisant la matière dans laquelle cette note a été obtenu.

Exemple 1. Un exemple "simple" de jointure entre 4 tables :

Déterminer les matières suivies par l'élève Gomez et le nom de ses professeurs.

3 Requêtes plus complexes

A] Jointure et réunion ou intersection

Formuler en langage SQL les requêtes suivantes et observer le résultat obtenu.

Déterminer les élèves qui ont eu au moins une fois, une note inférieure à 10 dans la matière 1 OU la matière 2.

B] Sélection et différence

Formuler en langage SQL les requêtes suivantes et observer le résultat obtenu.

Déterminer la liste des identifiants des élèves qui n'ont jamais eu au moins une fois une note inférieure à 15.

C] Sélection, jointure et différence

Formuler en langage SQL les requêtes suivantes et observer le résultat obtenu.

1. Déterminer la liste (noms et prénoms) des élèves qui n'ont jamais eu moins de 15 à un devoir.
2. Déterminer les matières dans lesquelles aucun élève a eu moins de 10.

4 Requêtes portant sur des fonctions d'agrégation

Rappelons qu'une fonction d'agrégation s'exerce sur un attribut (souvent numérique) et porte sur des sous-ensembles d'une table. Sauf lorsque la fonction porte sur l'ensemble des enregistrements de la table, la requête SQL doit contenir l'instruction GROUP BY afin de préciser la nature des sous-ensembles.

Cette requête prend donc la forme :

SQL
SELECT attribut(s) définissant les sous-ensembles, fonctions d'agrégation
FROM nom des différentes tables nécessaires à la requête
GROUP BY attribut(s) définissant les sous-ensembles

Les fonctions d'agrégation les plus usuelles sont : COUNT(), MIN(), MAX(), AVG(), SUM().

A] Agrégation sur l'ensemble de la table

Formuler en langage SQL la requête suivante et observer le résultat obtenu.

Donner le nombre de notes, la note minimal, la note maximale, la moyenne des notes de la table "notes".

B] Agrégation sans condition de sélection et sans jointure

Formuler en langage SQL la requête suivante et observer le résultat obtenu.

Donner les différentes notes obtenues et leur occurrence dans la table "notes".

C] Agrégation sans condition de sélection et avec jointure

Formuler en langage SQL les requêtes suivantes et observer le résultat obtenu.

Dans cette question, les moyennes sont calculées sans tenir compte des coefficients.

1. Donner la liste des élèves (noms et prénoms) et la moyenne de leurs notes.
2. Donner la liste des élèves et leur moyenne par matière.
3. Composition de requêtes :
 - (a) Déterminer la matière dans laquelle la moyenne des élèves est la plus élevée.
 - (b) Déterminer pour chaque matière, les élèves qui ont eu la moyenne la plus élevée.

Pour cela, on pourra :

- Déterminer la liste des moyennes par matière.
- Déterminer alors la moyenne la plus élevée.
- En déduire à l'aide d'une sélection en aval, la matière dont la moyenne est la plus élevée.

REPONSE : On obtient :

SI	19,75
----	-------

D] Agrégation avec une condition simple de sélection

L'utilisation d'une ou plusieurs fonctions de sélection peut s'accompagner d'une (ou plusieurs) condition(s) de sélection. Il y a alors 2 façons de formuler cette condition, avec WHERE (en amont du regroupement et portant sur les enregistrements pris en compte) ou HAVING (en aval du regroupement et portant sur les résultats de la fonction d'agrégation). Attention : ces deux formulations ne sont pas équivalentes!

SQL
SELECT attribut(s) définissant les catégories, fonctions d'agrégation
FROM nom des différentes tables nécessaires à la requête
WHERE condition(s) de sélection (portant sur les enregistrements)
GROUP BY attribut(s) définissant les catégories

ou

SQL
SELECT attribut(s) définissant les catégories, fonctions d'agrégation
FROM nom des différentes tables nécessaires à la requête
WHERE première(s) condition(s) de sélection (portant sur les données de la table initiale)
GROUP BY attribut(s) définissant les catégories
HAVING deuxième(s) condition(s) de sélection (portant sur les statistiques obtenues)

Formuler en langage SQL les requêtes suivantes et observer le résultat obtenu :

1. Combien d'élèves sont nés en 1993 ?

REPONSE : 9 élèves

2. Combien d'élèves ont déjà eu une note strictement inférieure à 10 ? (attention à ne pas compter plusieurs fois le même élève).

REPONSE : 6 élèves

3. Déterminer les moyennes des différents élèves (on affichera leur nom) sans tenir compte des notes au dessous de la moyenne.

REPONSE : 11 enregistrements dont :

Aumaitre	15
Balez	18,8
Barnoud	15,75
Drevet	16,5
Pioz	19,5
⋮	⋮

4. Donner la liste ordonnée des noms des élèves et des matières dans lesquelles ceux-ci ont eu une moyenne supérieure à 15.

REPONSE : 13 enregistrements dont :

Aumaitre	Physique	20
Balez	Anglais	17
Balez	Français	19
Balez	Allemand	20
⋮	⋮	⋮

5. Donner la liste des professeurs qui n'ont pas mis de note au dessous de 10 dans leur matière, en précisant la matière correspondante.

REPONSE :

Jouk	SI	19,5
Kuhn	Physique	13
Molliex	Informatique	11
Pichard	Espagnol	16

6. Déterminer la liste des profs qui ont au moins 4 notes dans leur matière, en précisant le nombre de notes.

REPONSE :

Alaoui	Anglais	7
Aumard	Français	8
Escande	Math	4
Frot	Allemand	5

7. Déterminer la liste des élèves qui ont eu plusieurs fois la note 20.

E] Agrégation avec une condition complexe de sélection

Il s'agit ici d'effectuer une sélection des données selon une condition qui fait apparaître un attribut et une fonction d'agrégation de cet attribut. Par exemple :

- "avoir une note inférieure à la moyenne"
- "avoir la meilleure note"
- "être dans le groupe d'étudiants les plus notés"...

Dans ce cas, il faudra calculer à part la valeur de comparaison grâce à une instruction de type :

"SELECT f(A) FROM ..."

Exemple :

Pour afficher les étudiants ayant obtenus une note supérieure à la moyenne générale, on écrira :

SQL

```
SELECT eleves.nom, notes.valeur FROM notes, eleves
WHERE notes.eleve = eleves.id
AND notes.valeur > (SELECT AVG(notes.valeur) FROM notes)
```

Formuler en langage SQL les requêtes suivantes et observer le résultat obtenu :

1. Déterminer le ou les élèves ayant obtenu la meilleure note en maths, en précisant cette note et la matière.

REPONSE : Math - ROGER - 20

2. Déterminer les élèves qui ont été les plus notés, en précisant leur nombre de notes.

REPONSE : BALEZ : 5 notes

F] Moyenne par élève

L'objectif est ici de rédiger une commande SQL permettant de déterminer la moyenne de chaque élève. Pour cela, on décompose la requête en sous-requêtes en prenant soin si nécessaire de renommer les attributs :

1. Construire la table associant à chaque identifiant d'élève le produit "note par coefficient"
2. Construire la table associant à chaque identifiant d'élève la somme des produits précédents
3. Construire la table associant à chaque identifiant d'élève la somme des coefficients des notes obtenues
4. Construire la table associant à chaque identifiant d'élève, sa moyenne
5. En déduire la table qui à chaque élève (nom + prénom) associe sa moyenne, ordonnée par "nom"

REPONSE : On obtient :

nom	prenom	moyenne
Arnaud	Clement	7,2
Aumaitre	Florent	14,69
Balez	Anna	18,85
Barnoud	Vincent	15,4
Delors	Thibaut	8,5
Drevet	Anthony	16,5
Favre	Anissa	12
Gillet	Damien	2
Gomez	Bertrand	14,4
Pioz	Herve	19,5
Raynaud	Olivier	14,13
Roger	Valentine	18,75
Sapaly	Clemence	11
Tenaud	Laure	14,56