Cours administration et programmation SQL

Bachelor 3 2023-2024

Laurent Jouanno

Interrogation de données

Laurent Jouanno

Les agrégats

Agrégats

- Les agrégats permettent de regrouper des lignes ayant des valeur commune dans une ou plusieurs colonnes

- Les fonctions d'agrégats permettent d'effectuer des opération statistiques sur un ensemble d'enregistrements

- Par exemple, on peut regrouper les abonnés par leur ville afin de pouvoir faire des statistiques sur les abonnés de chaque ville

Les agrégats

- La clause GROUP BY permet de préciser sur quelles colonnes va s'effectuer le regroupement.
- La clause GROUP BY se positionne après la clause WHERE
- Les colonnes présentes dans le SELECT doivent apparaitre dans le GROUP BY.
- Sauf si le GROUP BY se fait sur la clé primaire de la table. Dans ce cas n'importe quelle colonne de la table est utilisable dans le SELECT

Les agrégats

- Seules les fonctions d'agrégat peuvent utiliser des colonnes hors GROUP BY
- Les alias de colonne ne sont pas utilisable dans le GROUP BY

Les agrégats

Fonctions d'agrégats

- Les fonctions d'agrégat s'applique à une colonne non incluse dans le GROUP BY
- Elles effectuent, pour chaque valeur groupée, un calcul sur l'ensemble des valeur de la colonne pour ce groupe

Les agrégats

Fonctions d'agrégats

- Moyenne : avg(col)
- Nombre d'enregistrement : count(col)
- Valeur la plus petite : min(col)
- Valeur la plus grande : max(col)
- Total des valeur : sum(col)

Les agrégats

Fonctions d'agrégats

- ➔ Les fonctions d'agrégat sont utilisable sans GROUP BY
- ➔ Dans ce cas, la fonction est appliquée sur l'ensemble des lignes de la table

Les agrégats

- Syntaxe :
  - SELECT col1, fonction(col2) FROM table GROUP BY col1

Les agrégats

➔ Imaginons la table suivante :

id	client	tarif	date
1	Pierre	102	2012-10-23
2	Simon	47	2012-10-27
3	Marie	18	2012-11-05
4	Marie	20	2012-11-14
5	Pierre	160	2012-12-03

Les agrégats

- SELECT avg(tarif) FROM vente ;
  - Calculera la moyenne de la colonne tarif pout toute les ventes

- SELECT client, max(tarif) FROM vente GROUP BY client ;
  - Affichera la liste des clients avec pour chaque le tarif le plus grand

Les agrégats

Le filtrage des agrégats

- ➔ Il est possible de filtrer le résultat obtenu en ajoutant une condition sur une fonction d'agrégat
- ➔ On utilise la clause HAVING
- ➔ Cette clause le place après la clause GROUP BY et avant le ORDER BY
- ➔ Les alias de colonne ne peuvent pas être utilisés dans le HAVING

Les agrégats

Le filtrage des agrégats

- ➔ Certains moteurs de base de donnée ne permettent pas l'utilisation d'alias de colonne dans le HAVING
- ➔ MariaDB le permet

Les agrégats

- SELECT client, max(tarif) FROM vente GROUP BY client HAVING max(tarif) > 100;
  - Affichera la liste des clients avec pour chaque le tarif le plus grand
  - Le résultat est filtré en n'affichant que les clients ayant acheté un achat supérieur à 100

Mise en pratique

Exercice 25 :

- ➔ Compter le nombre total d'abonnés

Mise en pratique

Exercice 26 :

- Trouver l'âge (en années) de l'abonné le plus jeune
- Trouver l'âge (en années) de l'abonné le plus vieux

Mise en pratique

Exercice 27 :

- Afficher le nombre d’abonné pour chaque ville et trier par ordre descendant du nombre d'abonné

Mise en pratique

Exercice 28 :

- Affiche le nombre de livre pour chaque genre et catégorie
- Utiliser la table livre

Mise en pratique

Exercice 29 :

- Afficher le nombre d’abonné pour chaque ville et trier par ordre descendant du nombre d'abonné
- Limiter le résultat au villes ayant au moins 20 abonnés

Mise en pratique

Exercice 30 :

- Afficher le nombre d'abonné par famille
- Une famille correspond au personnes de même nom habitant dans la même ville
- Trier par ordre descendant du nombre

Mise en pratique

Exercice 31 :

- Compter le nombre de membre de chaque famille (même nom) et retourner pour chacune la date de naissance du plus jeune et du plus vieux

Mise en pratique

Exercice 32 :

➔ Compter le nombre de membre de chaque famille (même nom) et retourner pour chacune l'âge du plus jeune et du plus vieux en années

Gestion de la valeur nulle

- Tout calcul avec une valeur à null renvoi null comme résultat
- A l'exception de la fonction count, les fonction d'agrégat ignorent les valeurs à null

Gestion de la valeur nulle

Fonction coalesce

- Cette fonction accepte un nombre variable d'arguments.
- Elle retourne le premier argument non null
- Exemple :
  - SELECT coalesce(libelle_long, libelle, 'vide')
    FROM table1

Gestion de la valeur nulle

Fonction ifnull

- Cette fonction prend 2 arguments en entrée
- Si le premier argument est null, le second est retourné
- Exemple :
  - SELECT ifnull(libelle, 'vide') FROM table1

Gestion de la valeur nulle

Fonction nullif

- Cette fonction prend 2 arguments en entrée
- Si les 2 arguments sont identiques, la fonction renvoi null.
- Sinon elle renvoi le 1er argument
- Exemple :
  - SELECT nullif(colonne, 'vide') FROM table1

Mise en pratique

Exercice 34 :

- Afficher les différentes catégories de la table livre en affichant '(autre)' si la catégorie est une chaîne vide

Requêtes multi-tables

Laurent Jouanno

Requêtes multi-tables

- Comprendre la notion de jointure
- Réaliser des extractions sur plusieurs tables

Cours donné par Laurent Jouanno

Relation entre les tables

Relation 1 à n

table : edition	
noEdition	clé primaire
edition	
adresse	
telephone	

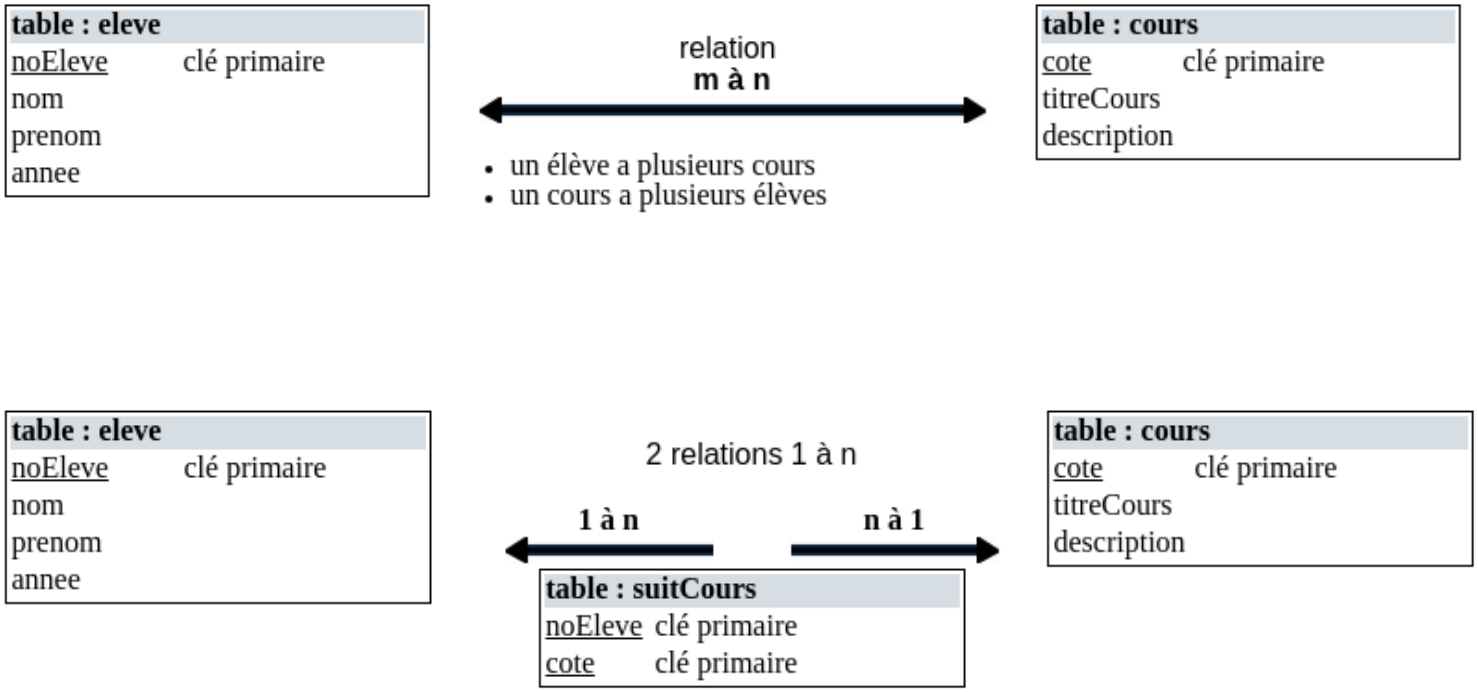
relation
1 à n

- une maison d'édition peut avoir plusieurs livre
- un livre à qu'une seule maison d'édition

table : livre	
no	clé primaire
titre	
sujet	
auteur	
pages	
noEdition	clé étrangère

Relation entre les tables

Relation m à n



Produit cartésien

- Le fait de mettre deux tables dans la clause FROM va composer un résultat à partir de toutes les combinaisons des lignes des deux tables
- Chaque ligne de la table 1 est répétée autant de fois qu'il y a de lignes dans la table 2
- Syntaxe :
  - SELECT champ1, champ2
    FROM table1, table2 ;

Nom des tables

- On peut préfixer les nom de colonne par le nom de la table avec un point entre les deux.
- Ceci améliore la lisibilité lorsque la requête comporte un nombre important de champs ou tables

Nom des tables

- Si une colonne de même nom existe dans plusieurs table de la requête, le moteur de base de données ne saura pas laquelle utiliser
- Dans ce cas, il faut obligatoirement préfixer le nom de la colonne avec le nom de la table

Jointure

- Pour lier les données de deux table, on va effectuer une jointure entre ces deux tables
- Il existe deux syntaxes :
  - Syntaxe "relationelle" (SQL89) : la condition de la jointure est placée dans la clause WHERE
  - Syntaxe normalisée (SQL2) : la condition de la jointure est placée dans la clause FROM avec le mot clé JOIN

Jointure

- Syntaxe "relationelle"
  - SELECT champ1, champ2
    FROM table1, table2
    WHERE table1.colonne = table2.colonne ;

Jointure

- Syntaxe normalisée
  - SELECT champ1, champ2
    FROM table1 JOIN table2 ON table1.colonne = table2.colonne

Nom des tables

- Il est possible de donner un nom d'alias aux tables
- Cet alias pourra être utilisé comme préfix des noms de colonne
- Syntaxe
  - SELECT t1.champ1, t2.champ2
    FROM table1 AS t1 JOIN table2 AS t2 ON
    t1.colonne_fk = t2.colonne_pk

Jointure externe

- Dans certains cas, le champ servant de clé étrangère n'est pas obligatoire
- Dans ce cas, la jointure ne pourra pas se réaliser pour les lignes où la valeur est nulle et elle ne seront pas ramenées par la requête
- Si on veut quand même ramener les lignes où la valeur est nulle, on va utiliser une jointure externe.

Jointure externe

- Syntaxe
  - SELECT champ1, champ2
    FROM table1 LEFT OUTER JOIN table2 ON
    table1.colonne = table2.colonne

Syntaxe

- Toutes les syntaxes vues précédemment sur les requêtes simples (filtrage, tri, agrégats, …) s'utilisent de même manière avec les requêtes multi-tables

Mise en pratique

Exercice 34 :

- La table auteur contient tous les auteurs avec pour chacun un identifiant
- La table livre contient tous les livres avec pour chacun l'auteur sous la forme de son identifiant
- Lister tous les livres avec leur auteur

Mise en pratique

Exercice 35 :

- La table editeur contient tous les éditeurs avec pour chacun un identifiant
- La table livre contient tous les livres avec pour chacun l'éditeur sous la forme de son identifiant
- Lister tous les livres avec leur éditeur
- Trier sur le titre des livres

Mise en pratique

Exercice 36 :

- ➔ Lister tous les livres avec leur auteur et leur éditeur
- ➔ Trier sur le titre des livres

Mise en pratique

Exercice 37 :

- Afficher la liste des éditeurs avec le nombre de livre de chacun
- Trier par ordre alphabétique des nom d'éditeur

Mise en pratique

Exercice 38 :

- Lister tous les livres avec l'ensemble des date d'emprunt de chacun
- Un livre emprunté plusieurs fois apparaitra donc autant de fois que d'emprunt
- Les livres jamais empruntés doivent aussi apparaitre
- Trier par titre et date d'emprunt

Mise en pratique

Exercice 39 :

- Lister tous les livres actuellement empruntés avec l'abonné ayant emprunté le livre
- Un livre emprunté est un livre pour lequel la date_retour dans la table emprunt n'est pas renseigné

Questions

Questions / réponse sur la séance