

# Base de données - Cours III

## Exercices SQL sur les fonctions d'agrégation, les sous-requêtes et la gestion des privilèges

Auteur : E. Thirion - Dernière mise à jour : 11/07/2015

### Liens utiles:

- [Voir le cours associé à ce document](#)
- [Installation des dossiers et logiciels nécessaires.](#)
- [Conditions d'accès au téléchargement.](#)
- [Voir tous les cours du même auteur.](#)

## I - Exercices sur l'utilisations des fonctions d'agrégation

### I-A - Fonctions d'agrégation sans GROUP BY

#### Exercice avec la base de donnée BDVols

Il s'agit d'afficher deux tables donnant des informations sur les vols de la compagnie Air France.

La première peut être obtenue sans utiliser les fonctions d'agrégation. Elle donne le nombre de passagers, ainsi que le taux d'occupation (défini pourcentage du nombre de passagers par rapport à la capacité de l'avion) de chaque vol. Voilà à quoi devrait ressembler les trois premières lignes de cette table de 512 lignes (votre ordinateur n'affichera pas forcément les mêmes valeurs numériques):

NumeroV	Compagnie	NbrPassager	TauxOccup
3	Air France	79	73.8318
4	Air France	141	64.0909
7	Air France	109	43.6000

La deuxième table donne des statistiques sur la table précédente et affiche:

- Le nombre total de vols Air France
- les valeurs minimales et maximales du nombre de passagers.
- les valeurs minimales et maximales du taux d'occupation.
- le nombre total de passagers transportés.

Voilà comment devrait se présenter votre table de une ligne (votre ordinateur n'affichera pas forcément les mêmes valeurs numériques):

NbrVolAirFrance	MinOccup	MaxOccup	MinPassager	MaxPassager	TotalPassager
512	2.800000000	97.200000000	7	243	45774

Indication : pour la première table utilisez une vue (cf requete **CREATE VIEW** cours II).

### Extrait de l'étude de cas CMM - 2008 - Option R - Dossier 3

À partir du schéma relationnel donné en annexe 2, établir la requête SQL qui calcule le nombre de connexions en échec (type\_message = "refusé") dans la journée du 1<sup>er</sup> janvier 2008.

### Extrait de l'étude de cas SCCP - 2010 - Option R - Dossier 4

À partir du schéma relationnel donné en annexe 3, établir la requête SQL qui calcule la quantité totale de maïs, de blé et d'orge livrée en 2009.

## **I-B - Fonctions d'agrégation avec GROUP BY et sans HAVING**

### **Exercice avec la base de données BDVols**

On voudrait obtenir le nombre moyen de places libres ainsi que le nombre de vols partant avant 8H en fonction de la ville de départ et de l'heure de départ comme suit:

VilleDepart	HeureDepart	NbrVol	MoyPlacesLibres
Londre	6	16	97.5625
Londre	7	16	105.4375
Londre	8	16	106.3125
Paris	6	16	109.1875
Paris	7	16	99.0625
Paris	8	16	111.0625
Strasbourg	6	16	106.2500
Strasbourg	7	16	78.6250
Strasbourg	8	16	82.1250

### **Exercice avec la base de donnée BDNotes**

Cette base de données se trouve dans le répertoire **BD**. Pour y accéder depuis **PhpMyAdmin** recopiez ce répertoire sous: `INSTALL_WAMP\wamp\bin\mysql\mysql5.1.36\data` où `INSTALL_WAMP` représente le chemin d'installation de **Wamp** sur votre machine (typiquement **C:**).

À partir du schéma relationnel donné en annexe 5, établir la requête SQL qui donne pour chaque contrôle: son numéro, son nom, sa date, la moyenne des notes, la note la plus faible et la note la plus élevée.

Vous devriez en principe obtenir le résultat suivant:

NumC	NomC	Date	Moyenne	NoteMin	NoteMax
1	BTSBlanc1	2010-11-24	9.4000	4	16
2	ControleFevrier	2011-02-10	7.8000	1	17
3	BTSBlanc2	2011-03-15	7.2000	1	18

#### **Extrait de l'étude de cas SCCP - 2010 - Option R - Dossier 4**

À partir du schéma relationnel donné en annexe 3, établir la requête SQL qui donne la quantité totale de céréales livrée par chaque adhérent en 2009, céréale par céréale ? *Le résultat sera trié par nom d'adhérent et par nom de céréale.*

#### **Extrait de l'étude de cas SEG - 2009 - Option D - Dossier 2**

À partir du schéma relationnel donné en annexe 4, établir la requête SQL qui :

1. donne la liste des abonnements avec indication du nom du client et du nombre de relevés effectués en 2008.
2. crée une vue nommée *MaxCompteur* présentant pour chaque compteur d'eau la référence de l'abonnement, le numéro d'ordre du compteur et l'index maximum relevé.

### **I-C - Fonctions d'agrégation avec GROUP BY et clause HAVING**

#### **Exercice avec la base de données BDVols**

On voudrait afficher des statistiques sur le taux d'occupation des vols, regroupées par avion. Voilà en gros à quoi devraient ressembler les trois premières lignes de votre table (les valeurs numériques ne seront pas exactement les mêmes sur votre ordinateur, mais elles devraient être du même ordre):

Avion	MinOccup	MaxOccup	MinPassager	MaxPassager	TotalPassager
50	24.8	92.4	62	231	1292
49	17.6	91.6	44	229	1296
91	29.2	96.8	73	242	1200

- **MinOccup**: minimum du taux d'occupation.
- **MaxOccup**: maximum du taux d'occupation.
- **MinPassager**: nombre minimum de passagers.
- **MaxPassager**: nombre maximum de passagers.
- **TotalPassager**: nombre total de passagers.

La table ne doit contenir que les avions dont le nombre total de passagers est supérieur à 1000.

On vous demande de plus de trier la table par nombre total de passagers dans l'ordre décroissant.

### Extrait de l'étude de cas ERGOSUM - 2008 - R - Dossier 3

À partir du schéma relationnel donné en annexe 1, écrire la requête SQL donnant la liste des séjours linguistiques se déroulant en Allemagne (référence, résumé et nombre d'excursions). *On ne retiendra que les séjours pour lesquels il y a plus de trois excursions.*

## II-Exercices sur l'utilisation des sous-requêtes

### II-A- Sous-Requêtes scalaire

#### Exercice avec la base de données BDVols (Annulation de Vols)

On voudrait annuler toutes les réservations des vols Paris-Strasbourg du 17/02/2009 partant avant midi. Cela signifie que le nombres de places libres de chacun de ces vols devra être remplacé par la capacité de l'avion qui l'effectue.

Lorsque vous aurez écrit votre requête, vous pourrez vérifier son fonctionnement en cliquant sur le lien **Vérification II-A** du menu **Tests Php**. Si votre requête fonctionne correctement vous obtiendrez l'affichage suivant:

NumeroV	VilleDepart	VilleArrivee	JourSem	Jour	HeureDepart	HeureArrivee	Compagnie	Constructeur	Modele	PlacesLibres
1	Paris	Strasbourg	Mardi	2009-02-17	6	8	Air France	Boeing	747	250
3	Paris	Strasbourg	Mardi	2009-02-17	7	9	Air France	Airbus	A320	107
5	Paris	Strasbourg	Mardi	2009-02-17	8	10	Air France	Airbus	A300	220
7	Paris	Strasbourg	Mardi	2009-02-17	9	10	Air France	Airbus	A320	107
9	Paris	Strasbourg	Mardi	2009-02-17	10	11	British Airways	Airbus	A320	107
11	Paris	Strasbourg	Mardi	2009-02-17	11	13	Air France	Boeing	747	250

**Exercice avec la base de données BDNNotes: notes les plus basses**

À partir du schéma relationnel donné en annexe 5, écrire la requête SQL donnant les notes étudiant ayant obtenus les notes les plus basses à chaque contrôle (nom du contrôle, note minimale, nom de l'étudiant.

Vous devriez obtenir le résultat suivant:

Controle	NoteMin	Nom
BTSBlanc1	4	nullos
ControleFevrier	1	fourien
BTSBlanc2	1	nullos
BTSBlanc2	1	fourien

**Exercice avec la base de données BDNNotes: notes par étudiant**

À partir du schéma relationnel donné en annexe 5, écrire la requête SQL donnant les notes que chaque étudiant a obtenu à chacun des trois contrôles. Ce résultats doit être obtenu directement à partir des tables ETUDIANT, TNOTE et CONTROLE, sans utiliser de vues.

Vous devriez obtenir le résultat suivant:

Numero	Nom	BtsBlanc1	ControleFevrier	BtsBlanc2
1	toto	12	17	2
2	schtroumpf	9	3	18
3	nullos	4	6	1
4	fourien	6	1	1
5	bosseur	16	12	14

**II-B- Sous-Requêtes colonne****Exercice avec la base de données BDVols**

Pour illustrer le principe des sous-requêtes colonne, nous avons introduit une nouvelle table dans la base: la table des tarifs. Elle donne le prix de chaque trajet selon la compagnie de transport, la plage horaire et le jour de la semaine. La colonne réduction précise le pourcentage de réduction sur le prix. Vous pouvez la visualiser en cliquant sur le lien **Tarifs** en haut à gauche.

La plage horaire est valable pour les vols dont l'heure de départ est strictement supérieure à l'heure minimale et inférieure ou égale à l'heure maximale.

Ecrivez à l'aide d'une sous\_requête colonne la requête permettant d'obtenir tous les vols **Londre-Paris** à tarif réduit partant à **11h** un **Dimanche**.

Vous devriez obtenir la table suivante:

NumeroV	JourSem	Jour	VilleDepart	VilleArrivee	Compagnie	HeureDepart	HeureArrivee
524	Dimanche	2009-02-22	Londre	Paris	Air France	11	12

### **Extrait de l'étude de cas CMM - 2008 - Option R - Dossier 3**

À partir du schéma relationnel donné en annexe 2, établir la requête SQL qui liste toutes les connexions (*login\_saisi*, *date*, *adresse*) qui ne concernent aucun employé enregistré dans la base.

### **Extrait de l'étude de cas ERGOSUM - 2008 - Option R - Dossier 3**

À partir du schéma relationnel donné en annexe 1, écrire la requête SQL donnant la liste des séjours linguistiques ne proposant pas d'excursion (référence et prix)

## **III-Exercices sur la gestion des droits**

### **III-1 Exercice avec la base de données BDVols**

Pour cet exercice utiliser cliquer sur le lien **Gestion des Privilèges** de la partie **Tests PHP** du menu. Cela fera apparaître un formulaire dans le cadre droit. Ce formulaire vous permettra d'exécuter différentes requêtes exemples en vous connectant avec le login de différents utilisateur (root, rantanplan et gaston).

Suivez les instructions suivantes dans l'ordre. Les requêtes utilisées sont résumées en annexe 6.

#### **1. Début**

**root** a tous les droits, rantanplan et gaston n'en ont aucun. Vérifiez le en exécutant quelques requêtes et en visualisant les tables **user**, **db** et **tables\_priv** avec **phpMyAdmin**.

#### **2. Droit de sélection à rantanplan**

Voir la requête **grant1**

Exécuter cette requête

Vérifier que rantanplan récupère ce droit (exécuter **select1**) et aucun autre.

Dans quelle table de la base MySQL est enregistré ce droit ?

Voir et exécuter la requête **revoke1**

Vérifier que rantanplan n'a plus ce droit

#### **3. Droit de création**

(a) Une table nommée à l'avance

Voir et exécuter la requête **grant2**

Vérifier que rantanplan peut créer la table **Aéroport** (voir et exécuter **create2**), mais ne peut rien faire d'autre ! (essayer par exemple **insert2** et **create3**)

Dans quelle table de la base MySql est enregistrée ce droit ?

*Remarque: En MySQL, le droit de créer ne donne aucun autre droit. Ce n'est apparemment pas le cas avec le SQL d'ORACLE.*

Retirer ce droit à rantanplan avec **revoke2** et supprimer la table **Aéroport** avec **drop2**.

(b) N'importe quelle table d'une base

Voir et exécuter **grant3**

Vérifier que rantanplan peut à présent créer les table **Aéroport** (**create2**) et **Personnel** (**create3**).

Dans quelle table de la base MySql est enregistrée ce droit ?

Retirer ce droit à rantanplan avec **revoke3**.

Supprimer les tables **Aéroport** et **Personnel** avec **drop2** et **drop3**.

#### 4. Tous les droits sur une table

Voir et exécuter **grant4**

Vérifier que rantanplan peut à présent exécuter toutes les requêtes sur la table Aéroport: **create2**, **insert2**, **delete2**, **drop2**

Dans quelle table de la base MySql est enregistrée ce droit ?

Vérifier que rantanplan ne peut pas donner les mêmes droits à gaston avec la requête **grant7**.

Retirer le droit de tout faire sur la table **Aéroport** à rantanplan avec **revoke4**.

#### 5. Le droit de donner des droits

Constater que ni rantanplan, ni gaston ne peuvent exécuter **select1**.

Exécuter **grant1**. Rantanplan peut donc à présent exécuter **select1**.

Voir la requête **grant6**

Constater que rantanplan ne peut pas l'exécuter : il a donc le droit de faire un select sur la table avions, mais il ne peut pas le redistribuer.

Voir la requête **grant5** et l'exécuter par **root**.

Constater que rantanplan peut à présent exécuter **grant6**

Vérifier que gaston peut à présent exécuter **select1**

Vérifier qu'il ne peut par contre pas retirer ce droit à rantanplan avec **revoke1**

Vérifier que rantanplan peut par contre retirer ce droit à gaston avec **revoke6**

Vérifier que gaston a effectivement perdu ce droit.

## III-2 Extraits d'études de cas

### III-2-A Cas CMM - 2008 - Option R - Dossier 3

Écrire la requête SQL qui autorise l'accès en lecture à la vue CONN-CHANTIER-4 au responsable du chantier n° 4, dont l'identifiant est DEMICHOT.

**III-2-B Cas Tholdi - 2009 - Option R - Dossier 4**

M. Peguse, administrateur de la base, est chargé de la mise à disposition des différentes tables à M. Bonus qui saisit les informations. Il souhaite donner accès en lecture à M. Bonus à toutes les tables de la base, sauf à la table EMPLOYE, et en insertion aux tables REPARATION et INTERVENIR. M. Bonus pourra redistribuer ses droits de lecture sur la table CONTAINEUR à qui bon lui semble. Le compte utilisateur de M. BONUS a pour nom RBonus.

Écrire la ou les requêtes SQL que doit réaliser M. Peguse pour attribuer les droits à cet utilisateur.

**III-2-C Cas Fefort - 2009 - Option R - Dossier 3**

Pour les interventions sur la base de données, le DSI souhaite créer un compte utilisateur dans le SGBD à qui on attribuerait tous les droits sur les tables EQUIPEMENT, LOGICIEL et INSTALLER et seulement le droit d'interroger la table USER.

Écrire les requêtes SQL qui affectent ces droits au compte utilisateur *gpiTechUser*.

**III-2-D Cas ASDOMI - 2010 - Option D - Dossier 2**

Écrire les ordres SQL permettant d'autoriser tous les utilisateurs de la base de données à consulter la table INTERVENTION et l'utilisateur d'identifiant RDeltour à faire toute insertion, modification ou suppression de données sur cette même table.

**III-2-E Cas ACLICK - 2008 - Option R - Dossier 3**

Vous êtes l'administrateur de cette base de données et vous devez enregistrer les droits de l'utilisateur Dupont. Sur la table PARUTION celui-ci est autorisé à consulter tous les attributs mais ne peut modifier que la date de parution.

Écrire la requête SQL permettant d'attribuer les droits à l'utilisateur Dupont.



## ANNEXES

### Annexe 1 : Schéma relationnel pour ERGOSUM

Lieu (id, libellé, pays)  
Clé primaire : *id*

Séjour (ref, résumé, dateDébut, dateFin, prix, type, idLieu)  
Clé primaire : *ref*  
Clé étrangère : *idLieu* en référence à *id* de *Lieu*

*Remarque : l'attribut type est de type caractère et contient 'T' pour un séjour thématique et 'L' pour un séjour linguistique.*

VilleDépart (id, nom)  
Clé primaire : *id*

Partir (refSéjour, idVille, supplément)  
Clé primaire : *refSéjour, idVille*  
Clé étrangère : *refSéjour* en référence à *ref* de *Séjour*  
Clé étrangère : *idVille* en référence à *id* de *VilleDépart*

Excursion (refSéjour, num, objet, idLieu)  
Clé primaire : *refSéjour, num*  
Clé étrangère : *refSéjour* en référence à *ref* de *Séjour*  
**Clé étrangère : idLieu en référence à id de Lieu**

### Annexe 2: Schéma relationnel pour l'étude de cas CMM

**CONNEXION**(numéro, type\_message, code\_erreur, adresse\_mac, date, login\_saisi, ip\_authentificateur)

**EMPLOYÉ**(numéro, login, nom, fonction)

**CHANTIER**(numéro, adresse, responsable, ip\_authentificateur)

**POSTE**(numéro, adresse\_mac, marque, type)

**HABILITER**(#utilisateur, #poste)

*utilisateur, poste : Clé primaire*

*utilisateur : Clé étrangère en référence à numéro de EMPLOYÉ*

*poste : Clé étrangère en référence à numéro de POSTE*

### Annexe 3: Schéma relationnel pour l'étude de cas SCCP

**SILO** (numero, designation)  
numero : clé primaire

**CEREALE** (code, nom)  
code : clé primaire (BLE pour blé, ORG pour Orge, MAI pour Mais)

**ADHERENT** (numero, nom, rue, codePostal, ville, silo)  
numero : clé primaire  
silo : clé étrangère en référence à numero de SILO

**LIVRAISON** (numero, date, refAdherent)  
numero : clé primaire  
refAdherent : clé étrangère en référence à numero de ADHERENT

**LIVRER** (numLivraison, refCereale, quantite)  
 numLivraison, refCereale : clé primaire  
 numLivraison : clé étrangère en référence à numero de LIVRAISON  
 refCereale : clé étrangère en référence à code de CEREALE

#### **Annexe 4:** Schéma relationnel pour l'étude de cas SEG

**CLIENT** (id, nom, prenom, adresse, cp, ville)  
 id : Clé primaire

**ABONNEMENT** (ref, date, client, adresse, cp, ville)  
 ref : Clé primaire  
 client : Clé étrangère en référence à id de CLIENT

**COMPTEUR** (abonnement, numOrdreCompteur, dateInstallation, marque)  
 abonnement, numOrdreCompteur : Clé primaire  
 abonnement : Clé étrangère en référence à ref de ABONNEMENT

**RELEVÉ** (abonnement, numOrdreCompteur, numOrdreReleve, date, index)  
*La clé primaire et les clés étrangères ne sont pas indiquées ici.  
 Chaque relevé est identifié par l'identifiant du compteur concerné et un numéro séquentiel. L'attribut index de la table RELEVÉ indique le nombre de m<sup>3</sup> d'eau mesuré par le compteur à la date du relevé (consommation réalisée depuis la mise en service du compteur). Il convient de remarquer que lors de l'installation d'un nouveau compteur, l'index du compteur est initialisé à zéro.*

#### **Annexe 5:** Schéma relationnel pour la base de données BDNNotes

**ETUDIANT** (NumE, Nom)  
 NumE : numéro de l'étudiant = clé primaire

**CONTROLE** (NumC, NomC, Date)  
 NumC : numéro du contrôle = clé primaire  
 NomC : nom du contrôle

**TNOTE** (NumE, NumC, Note)  
 NumE : clé étrangère en référence à NumE de ETUDIANT  
 NumC : clé étrangère en référence à NumC de CONTROLE

Annexe 6: Requêtes tests pour la gestions des privilèges

Nom	Table	Ligne
select1	Avions	
insert1	Avions	Avion 100
delete1	Avions	Avion 100
create2	Aeroport	
drop2	Aeroport	
insert2	Aeroport	Charles de Gaule
delete2	Aeroport	Charles de Gaule
create3	Personnel	
drop3	Personnel	

Requête du type grant (g) et revoke (r)

Requete	Priv	On	To	Gr Opt.
G1+R1	Select	Avions	Rantanplan	Non
G2+R2	Create	Aeroport	Rantanplan	Non
G3+R3	Create	bdvols.*	Rantanplan	Non
G4+R4	All	Aeroport	Rantanplan	Non
G5	Select	Avions	Rantanplan	Oui
G6+R6	Select	Avions	Gaston	Non
G7	All	Aeroport	Gaston	Non