



Série N 3/ Module 16
Système de Gestion de Base de Données (I)
LMD (Select)

Exercice I :

Soit le modèle relationnel suivant relatif à la gestion des notes annuelles d'une promotion d'étudiants :

ETUDIANT (N_Etudiant, Nom, Prénom)
MATIERE (CodeMat, LibelleMat, CoeffMat)
EVALUER (N_Etudiant, CodeMat, Date, Note)

Questions: Etablir les Commande SQL Correspondant aux requêtes suivantes:

- 1 - Quel est le nombre total d'étudiants ?
- 2 - Quelles sont, parmi l'ensemble des notes, la note la plus haute et la note la plus basse ?
- 3 - Quelles sont les moyennes de chaque étudiant dans chacune des matières ?
- 4 - Quelles sont les moyennes par matière ?
- 5 - Quelle est la moyenne générale de chaque étudiant ?
- 6 - Quelle est la moyenne générale de la promotion ?
- 7 - Quels sont les étudiants qui ont une moyenne générale supérieure ou égale à la moyenne générale de la promotion ?

Solution :

- 1) SELECT Count(N_Etudiant) FROM Etudiant
- 2) SELECT Max(Note) as 'Note Max', Min(Note) as 'Note Min' FROM Evaluer
- 3) SELECT Etudiant.N_Etudiant ,Nom, Matiere.CodeMat , LibelleMat, AVG(Note) as 'Moyenne'
FROM Etudiant, Evaluer, Matiere
WHERE Etudiant.N_Etudiant=Evaluer.N_Etudiant AND Evaluer.CodeMat=Matiere.CodeMat
GROUP By Etudiant.N_Etudiant, Matiere.CodeMat
- 4) SELECT LibelleMat, AVG(Note) as 'Moyenne' FROM Evaluer, Matiere
WHERE Evaluer.CodeMat=Matiere.CodeMat
GROUP By CodeMat
- 5) On crée une vue pour laQuestion 3 (V3) et on l'utilise ICI.
SELECT Nom, SUM(Moyenne* CoeffMat)/SUM(CoeffMat) as 'MoyenneNote'
FROM Etudiant, V3, Matiere
WHERE Etudiant.N_Etudiant=V3.N_Etudiant AND V3.CodeMat=Matiere.CodeMat
GROUP By Etudiant.N_Etudiant
- 6) On crée la question 5 sous forme de Vue (V5) et on l'utilise ici :
Select AVG(MoyenneNote) From V5
- 7) SELECT Nom, SUM(Note* CoeffMat)/SUM(CoeffMat) as 'MoyenneNote'
FROM Etudiant, Evaluer, Matiere
WHERE Etudiant.N_Etudiant=Evaluer.N_Etudiant AND Evaluer.CodeMat=Matiere.CodeMat
GROUP By N_Etudiant
HAVING SUM(Note* CoeffMat)/SUM(CoeffMat)>=(Select AVG(MoyenneNote) From V5)

Exercice II :

Soit le schéma relationnel suivant:

EQUIPE (CodeEquipe, NomEquipe, DirecteurSportif)
COUREUR (NumeroCoureur, NomCoureur, CodeEquipe*, CodePays*)



PAYS (CodePays, NomPays)

TYPE_ETAPE (CodeType, LibelléType)

ETAPE (NuméroEtape, DateEtape, VilleDép, VilleArr, NbKm, CodeType*)

PARTICIPER (NuméroCoureur*, NuméroEtape*, TempsRéalisé)

ATTRIBUER_BONIFICATION (NuméroEtape*, km, Rang, NbSecondes, NuméroCoureur*)

Questions: Etablir les Commande SQL Correspondant aux requêtes suivantes:

- 1 - Quelle est la composition de l'équipe Festina (Numéro, nom et pays des coureurs) ?
- 2 - Quel est le nombre de kilomètres total du Tour de France 97 ?
- 3 - Quel est le nombre de kilomètres total des étapes de type "Haute Montagne" ?
- 4 - Quels sont les noms des coureurs qui n'ont pas obtenu de bonifications ?
- 5 - Quels sont les noms des coureurs qui ont participé à toutes les étapes ?
- 6 - Quel est le classement général des coureurs (nom, code équipe, code pays et temps des coureurs) à l'issue des 13 premières étapes sachant que les bonifications ont été intégrées dans les temps réalisés à chaque étape ?