Etudiant(id, nom, prénom, promotion)
- Professeur(id, nom, prénom, spécialité)
- Entreprise(id, nom, adresse, nombreStagesProposés)
- Stage(id, titre, niveau, durée, année, *idEnterprise, idEtudiant, idProfesseur*)
Les clés primaires sont soulignées ; les clés étrangères en italique.

Donner les noms des entreprises dont le nombre de stages proposés est supérieur à la moyenne; le résultat doit être trié selon le nombre de stages.

*SELECT nom*

*FROM Entreprise*

*WHERE nombreStagesProposés > (SELECT AVG(nombreStagesProposés) FROM Entreprise)*

*ORDER BY nombreStagesProposés;*

Compter pour chaque professeur le nombre de stages dont il est professeur référant. Nous nous intéressons seulement aux stages de niveau 'ING2' et un nombre de stages de ce niveau supérieur ou égal à 2.
*SELECT nom, prénom, COUNT(s.id)*
*FROM Professeur P, Stage S*
*WHERE P.id = S.idProfesseur*
*AND niveau = 'ing2'*
*GROUPE BY nom, prénom, idProfesseur*
*HAVING COUNT(S.id) >= 2;*

Donner le nombre des élèves de promotion 2012 qui ont effectué des stages à Capgemini les années 2010 ou 2011, de durée supérieure à 5 mois.

*SELECT count(E.id)
FROM Etudiant E, Stage S, Entreprise EE
WHERE E.id = S.idEtudiant
AND EE.id = S.idEntreprise
AND promotion = 2012
AND EE.nom = 'Capgemini'
AND annee BETWEEN 2010 AND 2011
AND durée > 5;

R1 = nom = 'Capgemini' (Entreprise)
R2 = (durée > 5) AND (année = 2010 OR année = 2011) (Stage)
R3 = promotion = 2012 (Etudiant)
R4 = R3 R2 R1
R5 = FCOUNT(Etudiant.id)(R4)*

Donner le nom et le prénom des professeurs qui ont suivi des stages de **toutes** les entreprises.
*SELECT P.nom, P.prénom*

*FROM Professeur P, Stage S*

*WHERE P.id = S.idProfesseur*

*GROUP BY P.nom, P.prénom, P.id*

*HAVING COUNT(DISTINCT.idEnterprise) = (SELECT COUNT(id) FROM Entreprise);
ρ nom, prenom,idEntreprise (Πnom, prenom, idEntrepise(Stage) P.id=idProfesseur Professeur)) ÷ (Πid (Entreprise)*Donner le nom et le prénom des étudiants qui n'ont jamais effectué des stages suivis par un professeur en mathématique.

*SELECT E.nom, E.prénom*

*FROM Etudiant E*

*MINUS*

*SELECT E. nom, E.prénom*

*FROM Etudiant E, Professeur P, Stage S*

*WHERE P.id = S.idProfesseur
AND E.id = S.idEtudiant*

*AND spécialité = 'Mathématique';*Donner pour chaque professeur en informatique son nom et le nombre de stages suivis durant l'année 2011, y compris les professeurs qui n'ont pas encadré d'élèves.

*SELECTnom, COUNT(s.id)*

*FROM Professeur P LEFT JOIN (SELECT \* FROM Stage WHERE année = 2011) ON P.id = S.idProfesseur*

*WHERE spécialité = 'Informatique'*

*GROUPE BY nom, P.id;*Donner un arbre algébrique optimisé en utilisant l’heuristique du cours pour la requête SQL suivante qui permet de trouver le titre ainsi que l'année des stages effectués à Paris de l'étudiant Martin Dupont, suivi par des professeurs de département informatique.
*SELECT titre, année*

*FROM Stage S, Etudiant E, Entreprise EE, Professeur P*

*WHERE E.id = S.idEtudiant*

*AND EE.is = S.idEntreprise*

*AND P.id = S.idProfesseur*

*AND E.prénom = 'Martin'*

*AND E.nom = 'Dupont'*

*AND P.spécialité = 'Informatique'*

*AND EE.adresse LIKE '% Paris%';*

Quel est le prix d'une chambre double de New York Palace le soir du réveillon 31/12/2012 ?

*SELECT prix*

*FROM Hotel H, Chambre C, Tarif*

*WHERE H.id = idHotel*

*AND idChambre = C.id*

*AND nom = ‘New York Palace’ AND tdate = to\_date(‘31/12/2012’,’DD/MM/YYYY’) AND type = ‘double’ ;*

Calculer la note moyenne de chaque hôtel. On veut afficher dans le résultat le nom, la ville et le pays de chaque hôtel ainsi que la note moyenne.

*SELECT nom, ville, pays, AVG(note)*

*FROM Hotel, Evaluation*

*WHERE id = idHotel*

*GROUP BY idHotel, nom, ville, pays ;*

*R1 = idHotelFidHotel, AVG(note) (Evaluation)*

*R2 = R1 idHotel = id Hotel*

*Res = Пnom, ville, pays, AVG(note) (R2)*

Quel est le client qui a évalué tous les hôtels de la ville Durbuy ? Donner son nom et son prénom.

*R1 = ρidHotel (Пid (σville = ‘Durbuy’ (Hotel)))*

*R2 = Evaluation ÷ R1*

*Res = Пnom,prenom(R2 idClient = id Client)*

*SELECT C.nom, prénom*

*FROM Client C, Hotel H, Evaluation*

*WHERE C.id = idClient*

*AND H.id = idHotel*

*AND ville = ‘Durbuy’*

*GROUP BY idClient, C.nom, prénom*

*HAVING COUNT(idHotel) = (SELECT COUNT(id)*

*FROM Hotel*

*WHERE ville = ‘Durbuy’);*

*SELECT nom, prénom*

*FROM Client C*

*WHERE NOT EXISTS*

*(SELECT \**

*FROM Hotel H*

*WHERE NOT EXISTS*

*(SELECT \* FROM Evaluation*

*WHERE C.id = idClient*

*AND H.id = idHotel*

*AND H.ville = ‘Durbuy’));*

Classer en ordre décroissant les hôtels selon le nombre de commentaires laissés par les clients, y compris les hôtels qui n'ont pas eu d'évaluation. On affichera le nom, la ville et le pays des hôtels ainsi que le nombre d’évaluations.

*SELECT nom, ville, pays, COUNT(commentaire)*

*FROM Hotel LEFT JOIN Evaluation ON id = idHotel*

*GROUP BY id, nom, ville, pays*

*ORDER BY COUNT(commentaire) DESC ;*

*R1 = HotelJOINTURE idHotel = id Evaluation*

*R2 = id, nom, ville, paysFnom, ville, pays, COUNT(commentaire) (R1)*

*Res = Tri(R2, COUNT(commentaire) DESC)*
Projection/Restriction/fonction/Agragat/Jointure

AVG(),SUM(),MIN(),MAX()



Compter le nombre de clients de chaque catégorie

*SELECT catégorie, COUNT(\*)*

*FROM Client*

*GROUP BY catégorie ;*

Spectacles dont la durée est >à durée moyenne
*SELECT titre
FROM spectacle
WHERE duree > (SELECT AV G(duree)
FROM spectacle);*noms des compagnies dont le nombre d'artistes est supérieur ou égal à10. *SELECT c.nom, COUNT(a.id)
FROM compagnie c, artiste a
WHERE c.id = a.idCompagnie
GROUP BY c.id, c.nom
HAVING COUNT(a.id) > 10;*