Ing1 - Examen de Bases de Données
EISTI 2011-2012 - CORRIGE

Mardi 17 janvier 2012 - Durée : 2h

1 feuille A4 recto-verso autorisée. Ordinateur et calculatrice interdits.

1ère Partie – Questions de cours – 8 points

Toutes les questions sont relatives à un énoncé et indépendantes entre deux énoncés différents.

Enoncé 1 :
On considère la relation musique de schéma suivant : Musique(artiste, année, titre, genre, pays).

Voici la relation complète constituée des seuls enregistrements suivants :

Musique	artiste	année	Titre	genre	pays
	Lady Gaga	2011	Marry the Night	Pop	Etats-Unis
	Adele	2010	Rolling In The Deep	Blues	Royaume-Uni
	Lady Gaga	2010	Telephone	Pop	Etats-Unis
	The Rolling Stones	1969	Honky Tonk Women	Rock	Royaume-Uni
	The Pogues	1988	Honky Tonk Women	Folk Rock	Royaume-Uni

Q1 [0.5 pt] Proposez une clé candidate pour cette relation ?

Au choix : (artiste,année), (artiste, titre), (titre, année), (genre, année), (pays, année), (titre, genre)

Q2 [1 pt] Pour chacune des dépendances fonctionnelles suivantes précisez si elles sont vraies ou fausses :

a) année -> genre
b) artiste -> genre, pays
c) titre, année -> artiste, genre, pays
d) genre -> pays

a) F
b) V
c) V
d) V

Q3 [0.5 pt] Les dépendances fonctionnelles suivantes sont-elles vraies et élémentaires :

a) année, titre -> genre
b) titre, année -> pays

a) V
b) F

Q4 [0.5 pt] Les dépendances fonctionnelles suivantes sont-elles vraies et directes :

a) titre -> pays
b) artiste -> pays

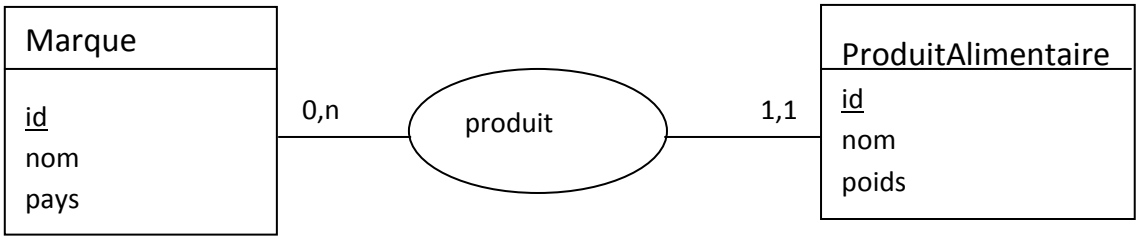
a) V
b) F

Q5 [1 pt] Proposez un nouveau modèle relationnel en BCNF (sans introduire d'autres attributs) pour ces données.

(artiste, pays), (genre, artiste) et (titre, année, artiste)
ou (artiste, pays, genre) et (titre, année, artiste)
ou (genre, artiste, pays) et (titre, année, artiste)

Enoncé 2 :

On considère le MCD suivant :



Q6 [0.5 pt] Donner la signification des 4 cardinalités de ce MCD.

0,n : 1 marque produit entre 0 et n produit
1,1 : 1 produit alimentaire est produit obligatoirement par une marque unique

Enoncé 3 :

On considère les relations suivantes :

- Film(idFilm, titre, genre, année, *idRealisateur*)
- Artiste(idArtiste, nom, prenom)
- Jouer(*idArtiste, idFilm*, rôle)
- *Film.idRealisateur* référence Jouer.idArtiste

Les autres clés étrangères ont le même nom que la clé primaire référencée. On suppose qu'un artiste n'a pas plus d'un rôle par film (d'où la clé primaire choisie pour l'association Jouer).

On a 1 000 artistes, 2 000 films et 10000 enregistrements pour Jouer. On considère que les enregistrements de la table Jouer sont répartis de manière homogène par film.

Q7 [0.5 pt] Quel est le coût estimé – en nombre d'enregistrements produits – des deux jointures suivantes ?

- Film JOIN Artiste on idRealisateur = idArtiste;

- Artiste NATURAL JOIN Jouer

- *2000 enregistrements produits*

- *10000 enregistrements produits*

Q8 [0.5 pt] Quel est le coût estimé – en nombre d'enregistrements produits – de la jointure suivante ?

- (Jouer NATURAL JOIN Film) JOIN Artiste on idRealisateur = idArtiste

- *10000 enregistrements produits*

Enoncé 4 :
Soit la table initialement vide suivante :

CREATE TABLE T (a INTEGER PRIMARY KEY, b INTEGER) ;

On a la séquence suivante d'ordres qui proviennent de 2 sessions S1 et S2 :

1 S2 : INSERT INTO T VALUES (1,2) ;
2 S2 : COMMIT;
3 S2 : INSERT INTO T VALUES (3,4) ;
4 S1 : INSERT INTO T VALUES (7,8);
5 S2 : INSERT INTO T VALUES (5,6) ;
6 S2 : SAVEPOINT PT1;
7 S2: UPDATE T SET b = b *2 WHERE a = 1;
8 S1 : DELETE FROM T WHERE a = 1
9 S2 : UPDATE T SET a = a * 2 WHERE a < 6;
10 S2 : INSERT INTO T VALUES (3,4);
11 S2 : ROLLBACK TO PT1;
12 S1 : INSERT INTO T VALUES (10,6);
13 S2 : COMMIT;
14 S1 : UPDATE T SET a = a * 2;
15 S1 : COMMIT;
16 S2 : ROLLBACK;
17 S1 : SELECT * FROM T;
18 S2 : SELECT * FROM T;

Q9 [1pt] Indiquer pour chaque session si une transaction (non vide) est en cours ou bloquée et le contenu affiché au final si on y parvient.

Deux réponses acceptées

- *En théorie : les 2 sessions se terminent, le contenu du T : (6,4) ; (14,8) ; (10,6) ; (20,6)*
- *En pratique sur Oracle : S1 est bloquée à l'ordre numéro 8, l'annulation de S2 à l'ordre numéro 11 ne débloque pas la ligne verrouillée, on ne peut pas continuer l'ordre numéro 12.*

Enoncé 5 :
Soit les types de requêtes suivantes parmi les plus fréquemment utilisées dans une BDD :

a) SELECT titre, genre FROM Musique WHERE genre LIKE '%Rock%';
b) SELECT titre FROM Musique WHERE genre LIKE 'Rock%';
c) SELECT artiste FROM Musique WHERE EXTRACT(YEAR FROM dateS) BETWEEN 2000 AND 2009;
d) SELECT titre FROM Musique ORDER BY duree;

Q10 [1pt] Pour chaque requête donner la liste des index à créer explicitement. Les attributs titre, genre, artiste et dateS ne font pas partie d'une clé primaire ou ne sont pas déclaré uniques.

a) 0
b) genre
c) 0
d) duree

Enoncé 6 :
Soit la vue suivante considérée comme valide pour faire des modifications :

CREATE VIEW MusiqueRock AS SELECT * FROM Musique WHERE genre like '%Rock%' WITH CHECK OPTION;

Q11 [0.5 pt] Que donne l'ordre suivant ?

INSERT INTO MusiqueRock (id, artiste, titre, genre)
VALUES (17345, 'Lady Gaga', 'Marry The Night', 'Pop');

Contrainte violée (with check option)

Q12 [0.5 pt] Que donne l'ordre suivant si l'id 10 référence une instance de Musique de genre 'Pop' ?

DELETE MusiqueRock WHERE id = 10 ;

0 ligne supprimée (pas dans la vue)

2ème Partie – Exercice – 12 points

Soit une base de données relationnelle qui permet de gérer les stages des étudiants de l'EISTI :

- Etudiant(<u>id</u>, nom, prénom, promotion)
- Professeur(<u>id</u>, nom, prénom, spécialité)
- Entreprise(<u>id</u>, nom, adresse, nombreStagesProposés)
- Stage(<u>id</u>, titre, niveau, durée, année, *idEnterprise, idEtudiant, idProfesseur*)

Les clés primaires sont soulignées ; les clés étrangères en italique.

Question 1 : Pour chaque question, proposer une requête en **SQL** :

a) 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;
```

b) 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;
```

Question 2 : Pour chaque question, proposer une requête en **algèbre relationnelle** et une requête équivalente en **SQL** :

a) 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.

$$F_{COUNT(Etudiant.id)}(((\sigma_{promotion = 2012}(Etudiant)) \bowtie (\sigma_{(durée > 5) AND (année = 2010 OR année = 2011)}(Stage))) \bowtie (\sigma_{nom = 'Capgemini'}(Entreprise)))$$

Ou

$R1 = \sigma_{nom = 'Capgemini'}(Entreprise)$

$R2 = \sigma_{(durée > 5) AND (année = 2010 OR année = 2011)}(Stage)$

$R3 = \sigma_{promotion = 2012}(Etudiant)$

$R4 = R3 \bowtie_{R2} \bowtie_{R1}$

$R5 = F_{COUNT(Etudiant.id)}(R4)$

```
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;

b) Donner le nom et le prénom des professeurs qui ont suivi des stages de **toutes** les entreprises.

$\rho_{nom, prenom, idEntreprise}(\prod_{nom, prenom, idEntrepise}(Stage) \bowtie_{P.id=idProfesseur} Professeur)) \div (\prod_{id}(Entreprise))$

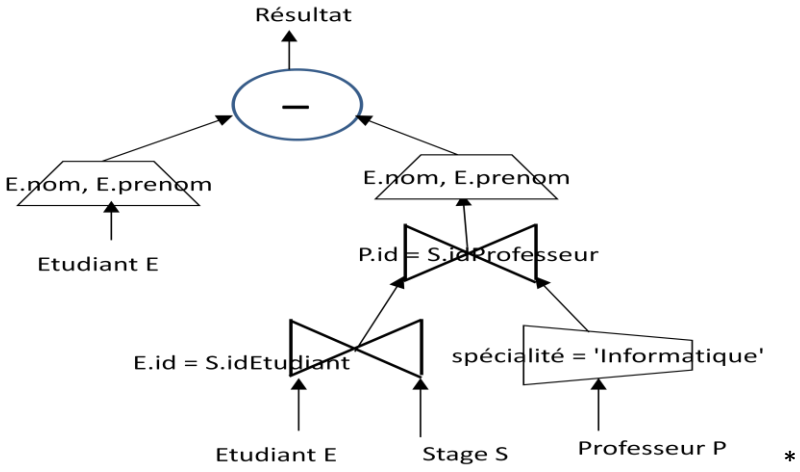
ou

$R1 = \prod_{nom, prenom, idEntrepise}(Stage \bowtie_{P.id=idProfesseur} Professeur))$
$R2 = \prod_{id}(Entreprise)$
$R3 = \rho_{nom, prenom, idEntreprise}(R1 \div R2)$

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);

Question 3 : Pour chaque question, proposer une requête en **arbre algébrique** et une requête équivalente en **SQL** :

a) Donner le nom et le prénom des étudiants qui n'ont jamais effectué des stages suivis par un professeur en mathématique.



*

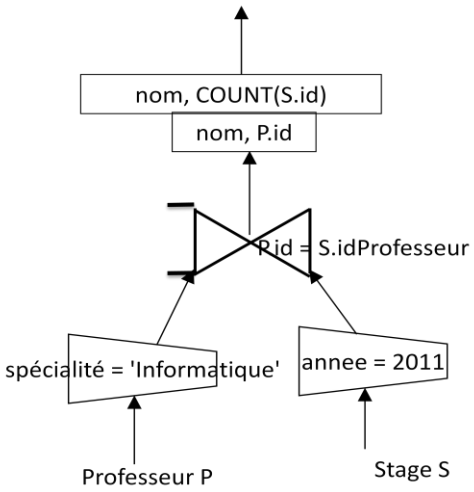
SELECT E.nom, E.prénom
FROM Etudiant E

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';

b) 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.



SELECT nom, 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;

Question 4 : 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

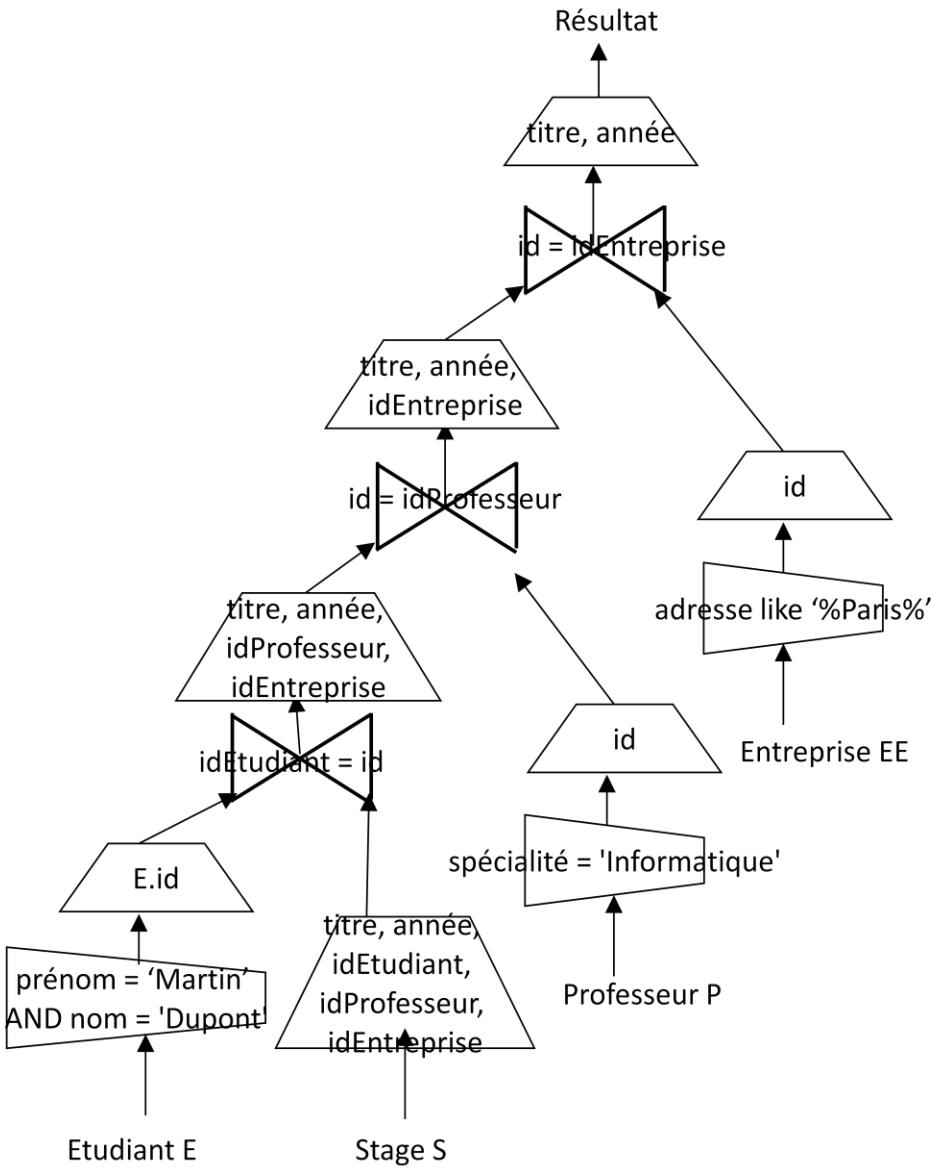
AND P.id = S.idProfesseur

AND E.prénom = 'Martin'

AND E.nom = 'Dupont'

AND P.spécialité = 'Informatique'

AND EE.adresse LIKE '% Paris%';


Résultat
titre, année
id = idEntreprise
titre, année, idEntreprise
id = idProfesseur
id
titre, année, idProfesseur, idEntreprise
adresse like '%Paris%'
idEtudiant = id
id
Entreprise EE
E.id
spécialité = 'Informatique'
prénom = 'Martin' AND nom = 'Dupont'
titre, année, idEtudiant, idProfesseur, idEntreprise
Professeur P
Etudiant E
Stage S