**Sujet 1**

**Trois Questions :**

**Q1**

Considérez le schéma relationnel suivant qui gère les factures d’une compagnie téléphonique mobile.

CLIENT (NumClient, Nom, Prénom, NumTéléphone, TypeAbonnement)

TARIFS\_ABONNEMENT ( TypeAbonnement, FraisConnexion, PrixParSeconde )

APPEL\_TELEPHONIQUE (NumClient, Date, Heure, NumAppel, Secondes)

FACTURE ( NumClient, Mois, Année, Montant )

* Écrivez un déclencheur (trigger) qui met à jour la facture d’un client après chaque appel téléphonique. **(1 points)**
* Écrivez un déclencheur qui réduit les factures avec 5 DA par appel aux clients de la compagnie à la fin de chaque mois, si le montant mensuel total de la facture dépasse 1000 DA**. (3 points)**

**Q2**

Considérez le schéma relationnel suivant

EMPLOYE (idE**,** *depart\_id*, nom)

DEPARTEMENT (idD, nameD)

PROJET (idP, *departement\_id*)

AFFECTATION (employe\_id, projet\_id)

**Q3**

Utiliser l’optimisation à base de règles (RBO : Rule-Base Optimisation) pour trouver un arbre optimal de la requête SQL suivante? **(2 pt).**

(SELECT DISTINCT idP

FROM PROJET, DEPARTEMENT, EMPLOYE

WHERE idD = departement\_id AND depart\_id = idD

AND nom = ‘Mohamed’)

UNION

(SELECT DISTINCT idP

FROM PROJET, AFFECTATION, EMPLOYE

WHERE idP = projet\_id AND idE = employe\_id

AND nom = ‘Mohamed’)

**Sujet 2**

**Q1**

Quelles sont les données validées après l’exécution des opérations suivantes ? **(1)**

SET AUTOCOMMIT =0 ;

INSERT INTO Coordonnées(X,Y) values (5, 6);

SAVEPOINT my\_savepoint\_1;

INSERT INTO Coordonnées(X,Y) values (7, 8);

SAVEPOINT my\_savepoint\_2;

INSERT INTO Coordonnées(X,Y) values (9, 10);

ROLLBACK TO my\_savepoint\_1;

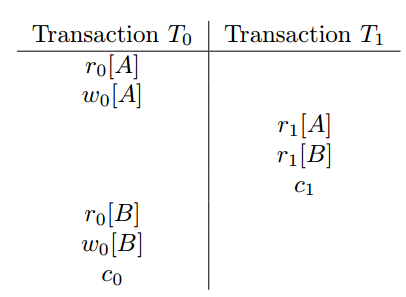
INSERT INTO Coordonnées(X,Y) values (11, 12);

INSERT INTO Coordonnées(X,Y) values (23, 6);

**Q2**

Soit les trois transactions , et et les opérations de lecture () et d’écriture (*)*. On dénote respectivement par() et (), l’accès à la ressource par la transaction pour la lecture ou l’écriture.

Considérez les deux transactions et l’ordonnancement suivant :



On dénote respectivement par() et (), l’accès à la ressource par la transaction pour la lecture ou l’écriture.

1. Cet ordonnancement est sérialisable par conflit ? Expliquez ? **(0.5)**
2. Utiliser l’algorithme 2PL (*two phase locking*) peut garantir un ordonnancement sérialisable par conflit pour les mêmes transactions ci-dessus. **(2)**

**Q3**

On considère deux relations R(A,B,C,D) et S(D,E) avec les statistiques suivantes :

T(R )=100, V(R, A) =100, V(R,B)=10, V(R,C)=1, V(R,D)=50, T(S)=500, V(S,D) =30 ,

V(S,E) =100, B(R) = 50 et B(S )= 100.

On dénote par  V(X, Y) le nombre de valeurs distinctes de l’attribut Y de la relation X , T(Y) Le nombre de tuples de la relation Y,  et B(Y)  le nombre de blocks pour stocker la relation Y.

On suppose que les données suivent une distribution uniforme, estimer le nombre de tuples satisfaisant le prédicat suivant :

1. **(1)**

*Calculer le nombre d'opérations d'entrée/sortie de disque pour l’opérations de la jointure suivante :*

*b)*

avec l’utilisation de l'algorithme de la jointure par deux boucles imbriquées et *101 pages mémoire sont disponibles.* **(1)**

**Sujet 3**

**Q1**

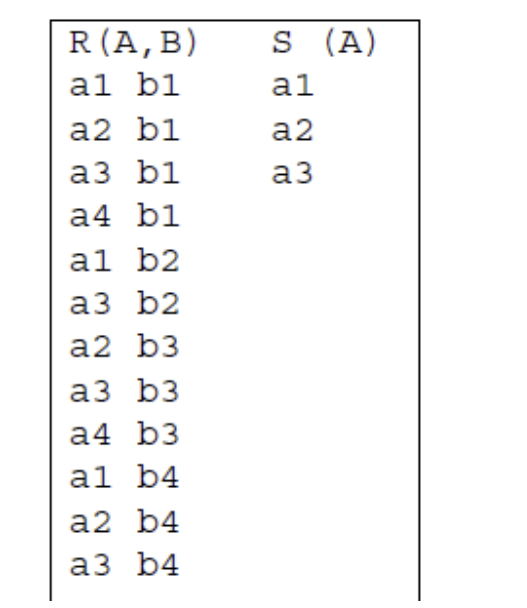
Soit le schéma relationnel de la base de données suivante :

* **PERSONNE** (idPersonne, nomPersonne, age)
* **AMI** (idPersonne1, idPersonne2)
* **FAMILLE** (idPersonne1, idPersonne2)

1. **Ecrire la requête SQL qui liste les personnes ayant une personne dans leur famille mais aucun ami.**
2. **Ecrire la requête SQL qui liste qui liste les noms des personnes avec des amis plus jeunes qu’elles**

**Q2**

Soit les deux relation R (A, B) et S (A) avec ces occurrences



1. Calculer le résultat de la division de R sur S : T=R%S
2. Ecrire T=R%S à partir des opérations de l’algèbre relationnel (projection, différence et produit cartésiens) :

**Q3**

Chaque groupe de mots ci-dessous appartient à une catégorie. A vous de trouver la catégorie en essayant d’être le plus précis que possible. La première ligne est un exemple.

|  |  |  |
| --- | --- | --- |
|  |  | **Catégorie** |
|  | Oracle, DB2, SQL Server, PostgreSQL | SGBD Relationnels |
|  |  | |
| 1 | COMMIT, ROLLBACK, SAVEPOINT, GRANT, REVOKE |  |
| 2 | Haute Disponibilité, Scalabilité,SQL Support, ACID |  |
| 3 | Photocopie, Copie vivante, Fragmentation |  |
| 4 | Haute disponibilité, Scalabilité, Pas de ACID |  |
| 5 | Entité/Association (EA), UML, Express. |  |
| 6 | Relationnel, Multidimensionnel, Réseau, Relationnel-Objet |  |
| 7 | Clé-valeur, Graphes, Documents, Colonnes |  |
| 8 | Shared memory, Shared disks, Shared nothing |  |

**Correction Sujet 1**

**Réponse Q1**

Considérez le schéma relationnel suivant qui gère les factures d’une compagnie téléphonique mobile.

CLIENT (NumClient, Nom, Prénom, NumTéléphone, TypeAbonnement)

TARIFS\_ABONNEMENT ( TypeAbonnement, FraisConnexion, PrixParSeconde )

APPEL\_TELEPHONIQUE (NumClient, Date, Heure, NumAppel, Secondes)

FACTURE ( NumClient, Mois, Année, Montant )

* Écrivez un déclencheur (trigger) qui met à jour la facture d’un client après chaque appel téléphonique. **(1 points)**

**create** **trigger** CallCharges  
**after insert of** APPEL\_TELEPHONIQUE  
**for each** **row**  
**begin**  
**update** FACTURE B  
set Montant = Montant + ( **select** PP. FraisConnexion +  
PP. PrixParSeconde \* new. Secondes  
**from** TARIFS\_ABONNEMENT PP join CLIENT C  
**on** C. TypeAbonnement = PT. TypeAbonnement  
**where** new. NumClient = C. NumClient)  
**where** B. NumClient = new. NumClient  
**and** B. Année = new.Date.year **and** B. Mois = new.Date.month  
**end**

**Réponse Q2**

* Écrivez un déclencheur qui réduit les factures avec 5 DA par appel aux clients de la compagnie à la fin de chaque mois, si le montant mensuel total de la facture dépasse 1000 DA**. (3 points)**

**create trigger** Offre **after** FIN\_ Mois

**begin**

**update** Facture B

**set** Montant = Montant – 5 \* ( **select** count(\*)

**from** APPEL\_TELEPHONIQUE P

**where** P. NumClient = B. NumClient

**and** P.Date.month = ( sysdate() – 1 ).month

**and** P.Date.year = ( sysdate() – 1 ).year

**and** P. NumAppel in ( **select** NumTéléphone **from** CLIENT) )

**where** B. Montant > 1000 and B. Année = (sysdate() - 1).year **and**

B. Mois = (sysdate() - 1).month

**end**

**Réponse Q3**

Considérez le schéma relationnel suivant

EMPLOYE (idE**,** *depart\_id*, nom)

DEPARTEMENT (idD, nameD)

PROJET (idP, *departement\_id*)

AFFECTATION (employe\_id, projet\_id)

Utiliser l’optimisation à base de règles (RBO : Rule-Base Optimisation) pour trouver un arbre optimal de la requête SQL suivante? **(2 pt).**

(SELECT DISTINCT idP

FROM PROJET, DEPARTEMENT, EMPLOYE

WHERE idD = departement\_id AND depart\_id = idD

AND nom = ‘Mohamed’)

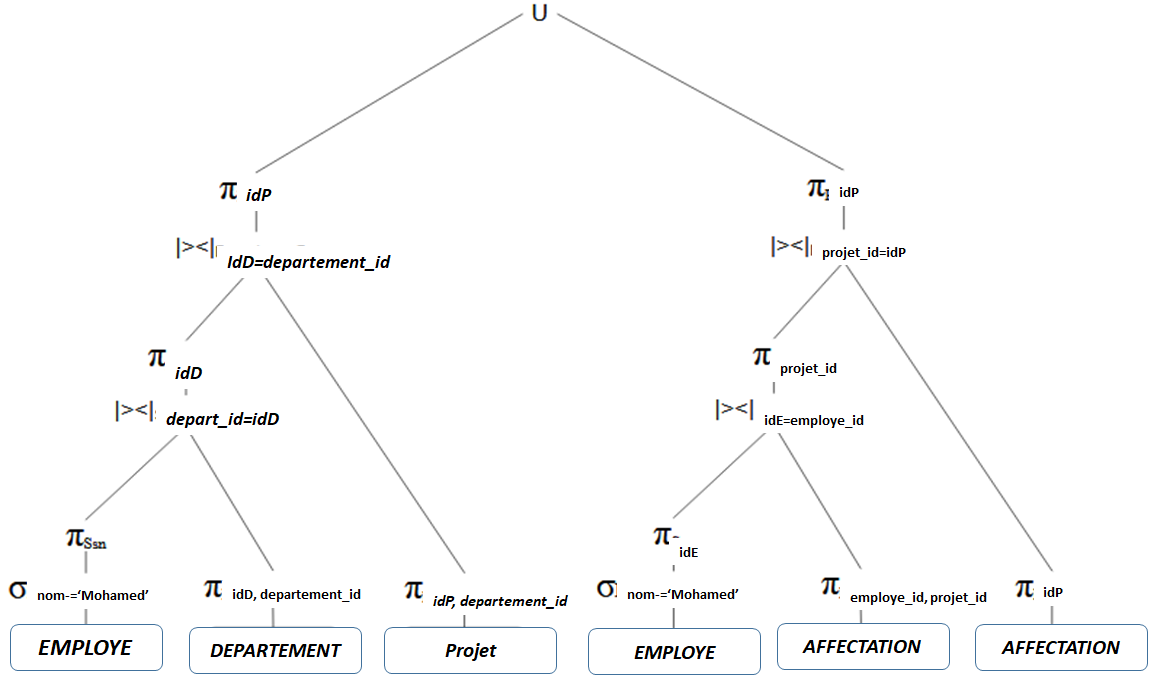
UNION

(SELECT DISTINCT idP

FROM PROJET, AFFECTATION, EMPLOYE

WHERE idP = projet\_id AND idE = employe\_id

AND nom = ‘Mohamed’)

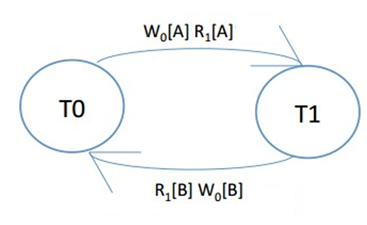


**Correction Sujet 2**

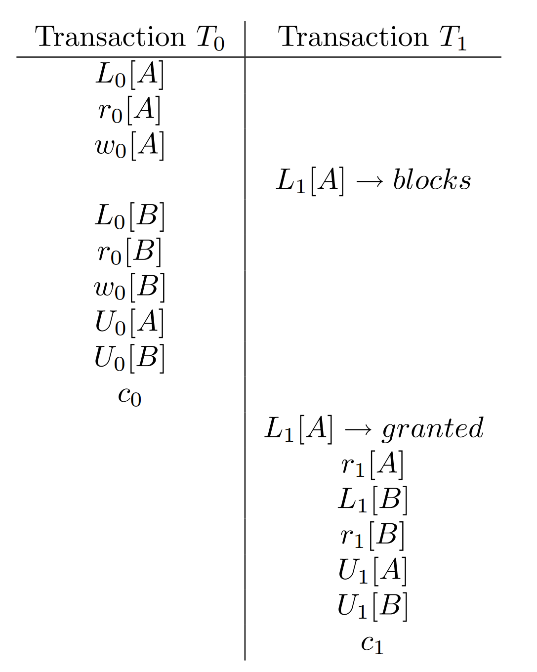
**Réponse Q1**

1) L’ordonnancement **n'est pas sérialisable** **par conflit** car le graphe de précédence contient un cycle.

* Le graphe a une arête T0 → T1 car l’ordonnancement contient w0[A] → r1 [A].
* Le graphe a une arête T1 → T0 car l’ordonnancement contient r1 [B] → w0 [B].



1. Montrez comment l’algorithme 2PL (*two phase locking*) peut garantir un ordonnancement sérialisable par conflit pour les mêmes transactions ci-dessus. **(L :Lock, U : Unlock)**



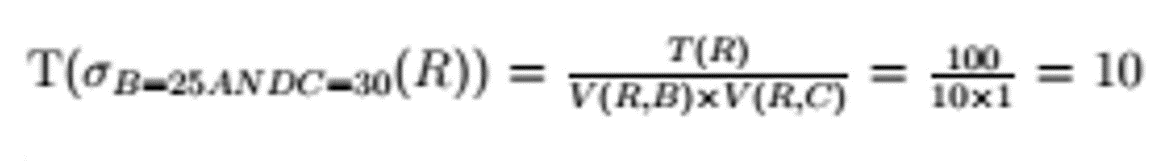
**Réponse Q2**

|  |  |
| --- | --- |
| **X** | **Y** |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

Résultat vide

Transactions pas encore validées (coMMiTées)

**Réponse Q3**



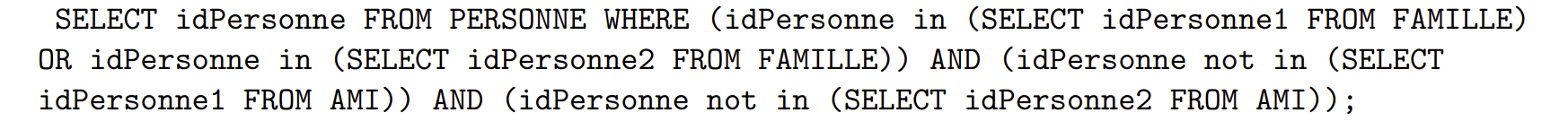
*Jointure par deux boucles imbriquées*

*= B(R )+ (T(R )\* B(S) / pages-1)= 50 + (*100 x 100)/(101-1)=150

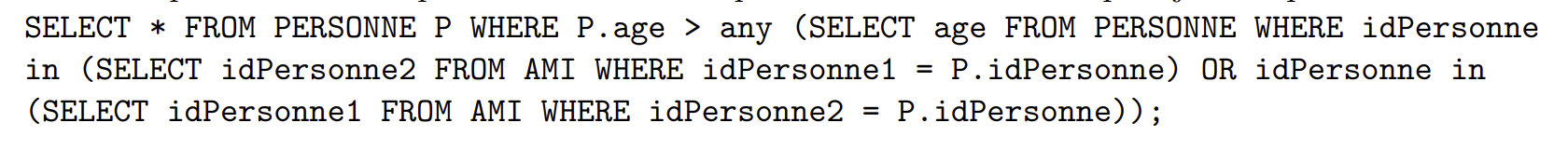
**Correction Sujet 3**

**Réponse Q1**

1. **Ecrire la requête SQL qui liste les personnes ayant une personne dans leur famille mais aucun ami.**

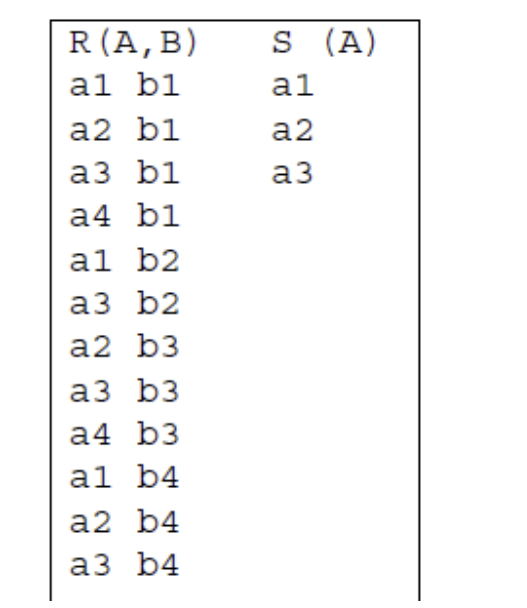


1. **Ecrire la requête SQL qui liste qui liste les noms des personnes avec des amis plus jeunes qu’elles**



**Réponse Q2**

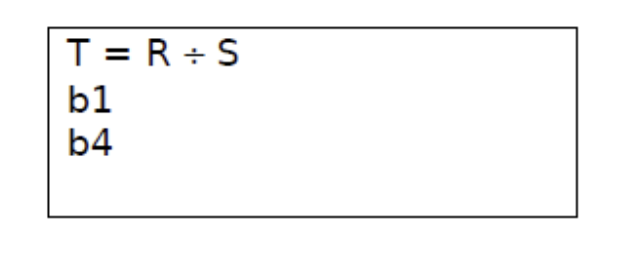
Soit les deux relation R (A, B) et S (A) avec ces occurrences

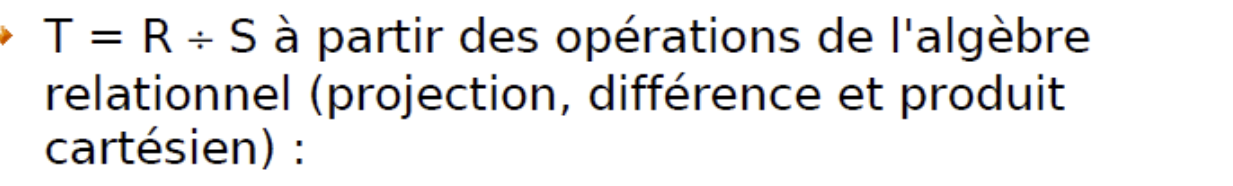


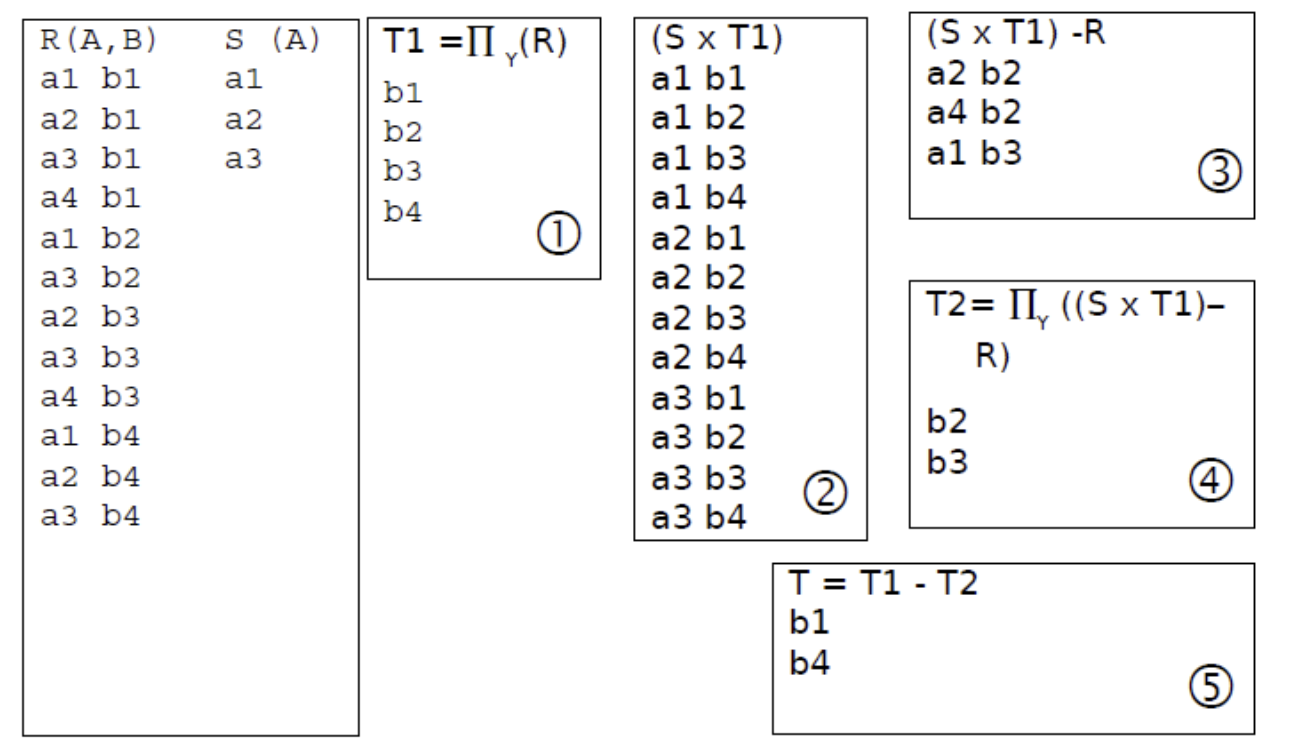
1. Calculer le résultat de division de R sur S : T=R%S
2. Ecrire T=R%S à partir des opérations de l’algèbre relationnel (projection, différence et produit cartésiens) :

Réponse :

Calculer le résultat de division de R sur S : T=R%S







**Réponse Q3**

Chaque groupe de mots ci-dessous appartient à une catégorie. A vous de trouver la catégorie en essayant d’être le plus précis que possible. La première ligne est un exemple.

|  |  |  |
| --- | --- | --- |
|  |  | **Catégorie** |
|  | Oracle, DB2, SQL Server, PostgreSQL | SGBD Relationnels |
|  |  | |
| 1 | COMMIT, ROLLBACK, SAVEPOINT, GRANT, REVOKE | Contrôle de transaction, LCD : langage de contrôle de données (**1)** |
| 2 | Haute Disponibilité, Scalabilité,SQL Support, ACID | NewSQL (**0,5)** |
| 3 | Photocopie, Copie vivante, Fragmentation | Modèle de distribution de BD (ou base de données repartie) (**0,5)** |
| 4 | Haute disponibilité, Scalabilité, Pas de ACID | NoSQL(**0,5)** |
| 5 | Entité/Association (EA), UML, Express. | MCD (Modèle conceptuel des données) **(1)** |
| 6 | Relationnel, Multidimensionnel, Réseau, Relationnel-Objet | MLD Modèle logique des données(**0,5)** |
| 7 | Clé-valeur, Graphes, Documents, Colonnes | Modèle de données NoSQL |
| 8 | Shared memory, Shared disks, Shared nothing | Evolution des Base de données distribuées |