

---

## CORRIGE DU CONTROLE FINAL

---

### Avertissements

**1/ Lisez attentivement le sujet.**

**2/ Les questions sont indépendantes les unes des autres.**

**3/ Soyez clair et précis.**

**4/ Soignez la présentation, dans la mesure du possible.**

**5/ Barème indicatif**

Question 1 : 3 points

Question 2 : 3,5 points (Q2.1 = 1,5 pt, Q2.2 = 2 pt)

Question 3 : 3 points (Q3.1 = 1,5 pts, Q3.2 = 1,5 pts)

Question 4 : 4 points (Q4.1 = 2 pts, Q4.2 = 2 pts)

Question 5 : 2,5 points

Question 6 : 2 points (Q6.1 = 1,5 pts, Q6.2 = 0,5 pts)

Question 7 : 2 points

Des étudiants de Télécom SudParis souhaitent développer dans le cadre d'une start-up une application de comparaison des circuits proposés par les tours opérateurs. Après une phase de conception, ils ont généré le modèle relationnel suivant :

Pays(nomPays, langue)  
Villes(nomVille, **nomPays**, population)  
TourOperateurs(nomTO, adresse, siteWeb, specialite)  
Circuits(nomCircuit, **nomTO**, descriptif, nbJours, prix, type, personnalisable)  
Etapes(**nomCircuit**, **nomTO**, **nomVille**, **nomPays**, noEtape, jour, duree)  
AVisiter(nomSite, **nomVille**, **nomPays**, prixEntree)

Les clés sont soulignées et les clés étrangères sont en gras.

La sémantique des diverses relations est la suivante :

Pays(nomPays, langue) : un pays est identifié par son nom (nomPays) et est décrit par sa langue. Un exemple de tuple de cette relation est ('Indonésie', 'Indonésien').

Villes(nomVille, **nomPays**, population) : une ville est identifiée par son nom (nomVille) et le pays (nompays) dans lequel elle est située. Il faut effectivement différencier 'Paris' en 'France' et 'Paris' aux 'USA'. Il existe bien d'autres exemples d'ailleurs (Valence en France et en Espagne, Vienne en France et en Autriche, Oran en Algérie, en Argentine, en Inde et aux USA). Une ville est caractérisée par sa population (en milliers d'habitants). nomPays est une clé étrangère qui référence nomPays dans la relation Pays.

Un exemple de tuple de cette relation est ('Ubud', 'Indonésie', 34).

TourOperateurs(nomTO, adresse, siteWeb, specialite) : un tour opérateur est identifié par son nom (nomTO). Il est caractérisé par l'adresse de son siège social, l'adresse (URL) de son site Web et sa spécialité.

Un exemple de tuple de cette relation est ('Huwans', 'rue Seguiet, Paris', 'Tourisme responsable').

Circuits(nomCircuit, **nomTO**, descriptif, nbJours, prix, type, personnalisable) : un circuit est identifié par son nom (nomCircuit) et par le nom du tour opérateur qui le propose. Il se peut que deux tours opérateurs choisissent le même nom pour leur circuit ... Un circuit est caractérisé par un descriptif, une durée en nombre de jours (nbJours), un prix, un type et par le fait qu'il soit ou non personnalisable. Les valeurs possibles de type sont 'Voyage de noce', 'En famille', 'Culture', 'Trek/sport'. Les valeurs possibles de personnalisable sont 'oui' ou 'non'.

Un exemple de tuple de cette relation est ('Entre mer, volcans et rizières', 'Huwans', 'Randonnées pédestres dans les rizières et sur le volcan Batur ...', 15, 2000, 'Trek/sport', 'oui').

Etapes(**nomCircuit**, **nomTO**, **nomVille**, **nomPays**, noEtape, jour, duree) : Les circuits sont composés d'étapes dans des villes. Ainsi, une étape est identifiée par le circuit auquel elle appartient et par la ville dans laquelle elle a lieu.

nomCircuit, nomTO est une clé étrangère qui référence nomCircuit, nomTO dans la relation Circuits.

nomVille, nomPays est une clé étrangère qui référence nomVille, nomPays dans la relation Villes.

Une étape est caractérisée par son numéro (noEtape) et par le numéro de jour dans le circuit (jour) et sa durée en heures.

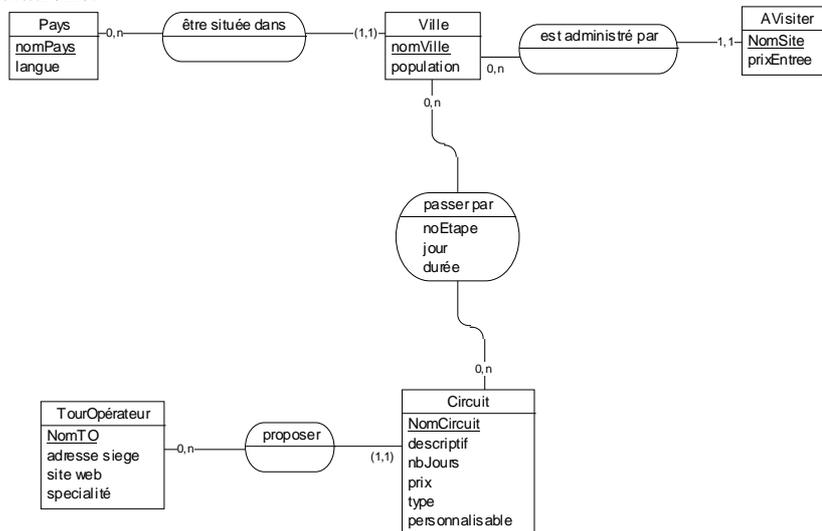
Un exemple de tuple de cette relation est ('Entre mer, volcans et rizières', 'Huwans', 'Ubud', 'Indonésie', 3, 5, 36). La troisième étape du circuit passe donc par Ubud le 5ème jour.

AVisiter(nomSite, **nomVille**, **nomPays**, prixEntree) : un « site » (cela peut être un monument, un site géographique, une place dans une ville) à visiter est identifié par son nom (nomSite). Il est caractérisé par la ville qui le gère et par son prix d'entrée. nomVille, nomPays est une clé étrangère qui référence nomVille, nomPays dans la relation Villes.

Un exemple de tuple de cette relation est ('Temple de Goa Gajah', 'Ubud', 'Indonésie', 0).

### Question 1 : Rétro-conception

En vous aidant des règles de passage d'une modélisation Entité/association à une modélisation relationnelle "à l'envers", proposer un schéma Entité/Association équivalent à ce schéma relationnel. Ce schéma devra comporter la description des entités (avec leurs propriétés), des entités faibles (s'il y a lieu), des associations (avec leurs propriétés) ainsi que les cardinalités minimum et maximum des associations.



### Barème : 3 points

Erreurs graves à pénaliser absolument :

- Recopie du modèle relationnel avec les clés étrangères dans le modèle entité-association : -1,5 points. Un modèle E/A ne contient jamais deux fois le même attribut ;
- Incompréhension des entités faibles (oubli des parenthèses autour des cardinalités 1,1 ; ajouter des liens d'association entre « pays » et « passer par » et « TourOpérateur » et « passer par ») : - 1 point
- Pas d'association (ellipse) entre deux entités. Non respect du formalisme. -1 point
- Oubli des clés dans les entités : -1 point. Toute entité doit posséder une clé.

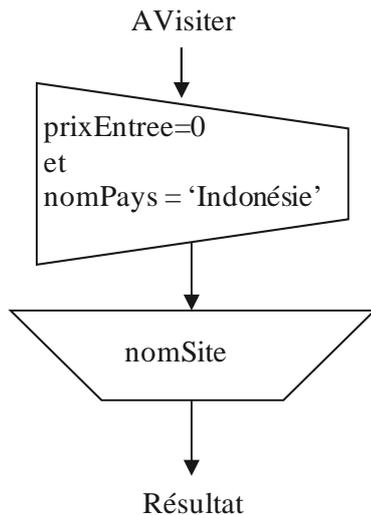
Erreurs moins graves :

- Erreurs de cardinalité maximale : -0,5 par cardinalité erronée ; Ne pas pénaliser les cardinalités minimales ;
- Oubli d'un attribut : -0,25 point par attribut oublié

### Question 2 : Algèbre relationnelle

Exprimer chacune des questions suivantes en algèbre relationnelle.

(Q2.1) Quels sont les sites (nom de site) gratuits en Indonésie ?



**Barème : 1,5 point**

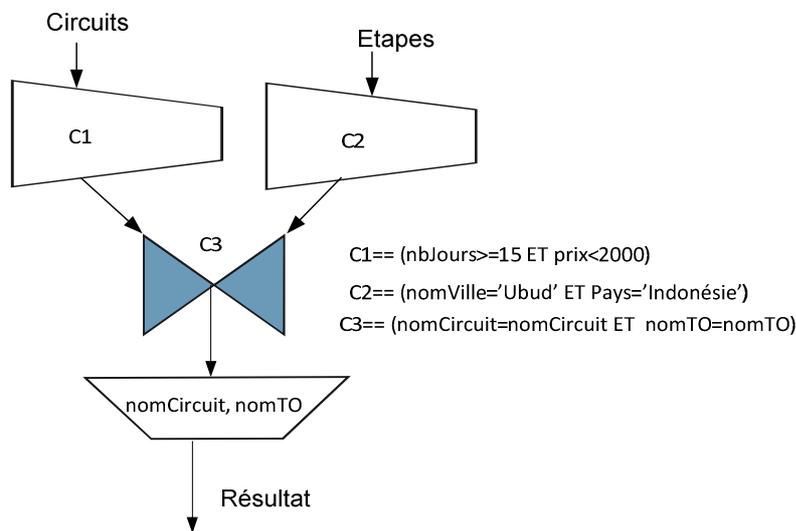
Noter positivement :

- Restriction : 1 point avec condition complexe correcte
- Projection : 0,5 point avec attribut correct

Erreurs à pénaliser :

- Inversion des opérateurs. On ne peut pas faire la restriction après la projection. Les attributs ne sont plus présents ;
- Complexité inutile (avoir par exemple mis d'autres relations) : -0,75 ;
- Erreur sur le nom de la relation : -0,25 point
- Oubli des " autour d'une valeur de type chaîne de caractères : -0,5

(Q2.2) Quels sont les circuits (nom de circuit et nom du tour opérateur) d'au moins 15 jours et coûtant moins de 2000 euros passant par Ubud en Indonésie ?



**Barème : 2 points**

Noter positivement :

- Restriction : 0,5 point avec condition complexe correcte
- Jointure : 0,5 point avec condition de jointure correcte
- Projection : 0,5 avec attribut correct
- Ordre des opérateurs (assemblage correct) : 0,5 points

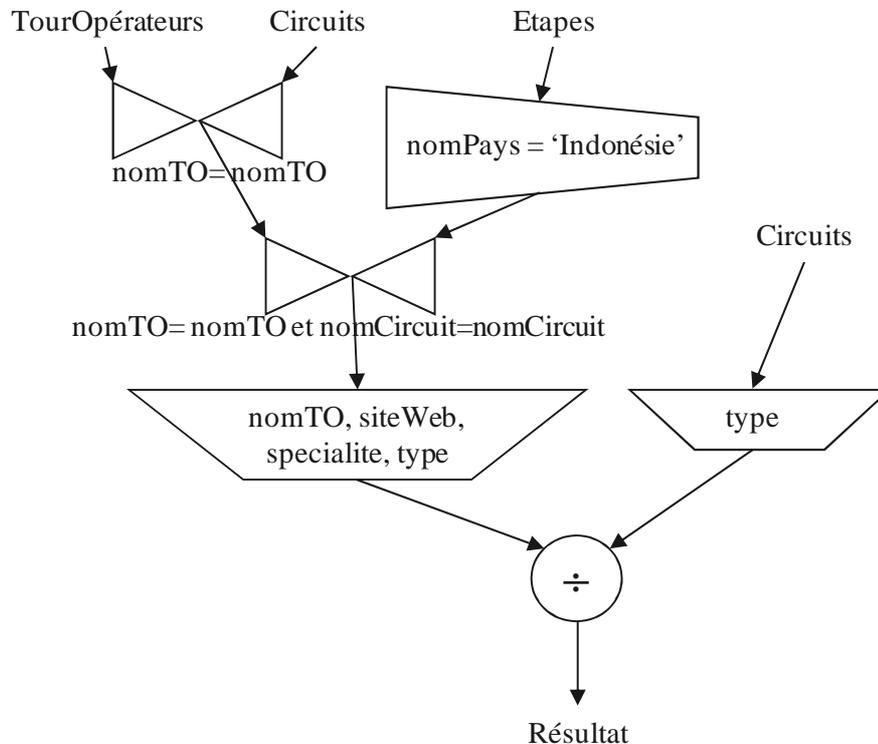
Erreurs à pénaliser :

- Complexité inutile (avoir par exemple mis d'autres relations) : -0,75 ;
- Erreur sur les noms de la relation : -0,25 point par relation erronée

### Question 3 : Requêtes algèbre relationnelle ou SQL

Exprimer chacune des ces questions suivantes en algèbre relationnelle **ou** en SQL .

(Q3.1) Quels sont les tours opérateurs (nom du tour opérateur, site web, spécialité) qui proposent en Indonésie tous les types de circuit ? Hypothèse : tous les types de circuits sont représentés dans la relation Circuits.



#### Barème : 1,5 point - Algèbre relationnelle

Noter positivement :

- Présence d'une division : 0,5 point
- Partie gauche construisant l'association des informations sur les tours opérateurs et les type de circuits proposés : 0,5
- Partie droite avec la liste des types de circuits : 0,25
- Projection (avant ou après la division) : 0,25

```
SELECT T.nomTO, siteWeb, specialite
FROM TourOperateurs T, Circuits C, Etapes E
WHERE T.nomTO = C.nomTO
AND E.nomCircuit = C.nomCircuit
AND E.nomTO = E.nomTO
AND E.nomPays='Indonésie'
GROUP BY T.nomTO, siteWeb, specialite
HAVING COUNT(DISTINCT type) = (
    SELECT COUNT(DISTINCT type)
    FROM Circuits C
);
```

#### Barème : 1,5 point - SQL

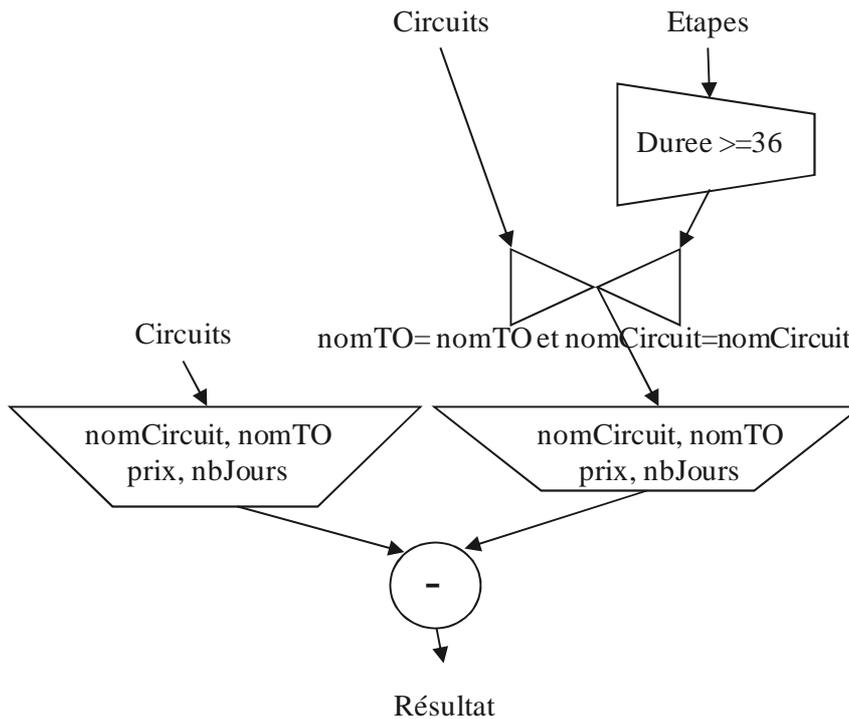
Noter positivement :

- Présence d'une partition (GROUP BY) correctement défini par rapport au SELECT (qui contient tous les attributs du SELECT) : 0,5 point

- Présence de la clause **HAVING** et de la requête imbriquée comptant le nombre de circuits différents
- Construction de l'association des tours opérateurs passant en Indonésie (clauses **FROM** et **WHERE**) avec les jointures complètes

Erreurs à pénaliser : **HAVING sans GROUP BY** : -0,5

**(Q3.2)** Quels sont les circuits « hyper-denses » ? Un circuit est dit « hyper-dense » s'il ne possède aucune étape de repos. Une étape de repos dure au moins 36 heures. Donner le nom du circuit, le nom du tour opérateur, le prix et la durée du circuit.



**Barème : 1,5 point – Algèbre relationnelle**

Noter positivement :

- Utilisation d'une différence : 0,5 point. Ordre (gauche et droite) des opérandes correct
- Partie droite donnant les circuits avec des étapes de repos : 0,5 point
- Alignement des schémas en entrée de l'opérateur de différence (deux projections identiques) : 0,5 point

```
SELECT nomCircuit, nomTO, prix, nbJours
FROM Circuits
MINUS
SELECT C.nomCircuit, C.nomTO, prix, nbJours
FROM Etapes E, Circuits C
WHERE duree >= 36
AND E.nomTO = C.nomTO
AND E.nomCircuit = C.nomCircuit
```

**Barème : 1,5 point – SQL**

Noter positivement :

- Différence : 0,25 point
- 1<sup>er</sup> SELECT donnant la liste de tous les circuits : 0,5
- 2<sup>eme</sup> SELECT donnant la liste des circuits ayant des étapes de repos : 0,75

Pénaliser :

- Contenu des SELECT différents (schémas différents) : -0,75

#### Question 4 : Requêtes SQL d'interrogation

Exprimer les questions suivantes en SQL :

(Q4.1) Donner pour chaque tour opérateur, le nom du tour opérateur, l'adresse de son site Web, le nombre de circuits proposés, le prix moyen des circuits, le prix le plus élevé et le prix le plus bas et la durée moyenne des circuits. Pour ces statistiques, les circuits d'exception (ceux dont le prix est supérieur à 4000 euros) ne doivent pas être considérés.

```
SELECT C.nomTO, siteWeb, COUNT(DISTINCT nomCircuit),  
       AVG(prix), MAX(prix), MIN(prix), AVG(nbJours)  
FROM Circuits C, TourOperateurs T  
WHERE C.nomTO =T.nomTO  
AND   prix<4000  
GROUP BY C.nomTO, siteWeb
```

#### Barème : 2 points

Noter positivement :

- Partitionnement, avec les deux attributs : 0,5 point ;
- FROM contenant les relations Circuits et TourOperateurs, pas de relation superflue : 0,25 point
- Condition de jointure : 0,25 point
- Condition de restriction : 0,25 point
- SELECT attributs simples corrects : 0,25 point
- SELECT agrégats corrects : 0,5 point

(Q4.2) Quel est le circuit du Maroc passant par le plus de sites à visiter gratuits ? Donner le nom du circuit, le nom du tour opérateur, le descriptif du circuit et le nombre de sites gratuits visités.

```
SELECT C.nomCircuit, C.nomTO, descriptif, COUNT(*)  
FROM Circuit C, Etape E, AVisiter A  
WHERE C.nomCircuit=E.nomCircuit  
      AND C.nomTO =E.nomTO  
      AND E.nomVille=A.nomVille  
      AND E.nomPays=A.nomPays  
      AND Prix=0  
      AND E.nomPays='Maroc'  
GROUP BY C.nomCircuit, C.nomTO, descriptif  
HAVING COUNT(*) >= ALL(  
    SELECT COUNT(*)  
    FROM Etape E, AVisiter A  
    WHERE E.nomVille=A.nomVille  
          AND E.nomPays=A.nomPays  
          AND Prix=0  
          AND E.nomPays='Maroc'  
    GROUP BY nomCircuit, nomTO)
```

#### Barème : 2 points

Noter positivement :

- Requête imbriquée calculant la liste des nombre de sites gratuits par circuit : 1 point
  - 0,25 point pour un SELECT limité à l'agrégat COUNT
  - 0,25 pour les trois relations dans la clause FROM et les conditions de jointure

- 0,25 pour les deux restrictions
- 0,25 pour le GROUP BY
- 0,25 pour le HAVING avec le prédicat ALL
- Requête principale (du dessus) : 1 point :
  - 0,25 point pour le SELECT avec attributs et agrégat
  - 0,25 pour les deux relations dans la clause FROM et les conditions de jointure
  - 0,25 pour les deux restrictions
  - 0,25 pour le GROUP BY

Donner 2,25 points si requête sans aucune erreur (bonus 0,25).

Erreurs à pénaliser : HAVING sans GROUP BY : -0,5

### Question 5 : Création de table SQL

Donner l'ordre de création complet (comprenant la définition des attributs, domaines, contraintes d'intégrité, clé et clés étrangères) de la relation Circuits(nomCircuit, nomTO, descriptif, nbJours, prix, type, personnalisable) :

- nomCircuit est une chaîne de 50 caractères ;
- nomTO est une chaîne de 40 caractères ;
- descriptif est une chaîne de 500 caractères. Il est obligatoire de donner un descriptif pour tout circuit proposé ;
- nbJours est un entier compris entre 5 et 31 ;
- prix est un entier. Cette information est obligatoire ;
- type est une chaîne de 14 caractères dont les valeurs possibles sont 'Voyage de noce', 'En famille', 'Culture', 'Trek/sport'.
- personnalisable est une chaîne de 3 caractères dont les valeurs possibles sont 'oui' ou 'non'.

```
CREATE TABLE Circuits(
  nomCircuit CHAR(50),
  nomTO CHAR(40),
  descriptif CHAR(500) NOT NULL,
  nbJours INTEGER CONSTRAINT CnbJours
    CHECK(nbJours BETWEEN 5 AND 31),
  prix INTEGER,
  type CHAR(14) NOT NULL CONSTRAINT Ctype
    CHECK(Ctype IN ('Voyage de noce', 'En famille', 'Culture',
    'Trek/sport')),
  personnalisable CHAR(3) CONSTRAINT Cpersonnalisable
    CHECK(Cpersonnalisable IN ('oui', 'non'))
);
```

```
ALTER TABLE Circuits ADD PRIMARY KEY (nomCircuit, nomTO) ;
```

```
ALTER TABLE Circuits ADD FOREIGN KEY nomTO REFERENCES
TourOperateurs(nomTO);
```

### Barème : 2,5 points

Noter positivement :

- 0,25 utilisation d'un CREATE TABLE
- 0,25 respect des types donnés dans le sujet
- 0,25 respect des noms d'attributs et de relation donnés dans le sujet
- 0,25 définition de la clé composée de deux attributs
- 0,25 CHECK nbJours
- 0,5 CHECK type

- 0,25 CHECK personnalisable
- 0,5 clé étrangère

### Question 6 : Création d'une vue SQL et droit

(Q6.1) Créer une vue PositionnementsTO(nomTO, pays, ville, nbCircuitsProposes, dureeMoyenne) qui permet d'avoir pour chaque ville (une ville est une étape de circuit proposé par un tour opérateur), le nombre de circuits proposés passant par cette ville et la durée moyenne des étapes dans cette ville. Pour cette vue, seuls les tours opérateurs ayant une présence réelle dans cette ville doivent être considérés. Un tour opérateur a une présence réelle s'il propose au moins 3 circuits passant par cette ville pour ce pays.

```
CREATE VIEW PositionnementsTO(nomTO, pays, , ville, nbCircuitsProposes,
dureeMoyenne) AS
SELECT nomTO, nomPays, nomVille, COUNT(*), AVG(duree)
FROM Etapes E
GROUP BY nomTO, nomPays, nomVille
HAVING COUNT(*) >= 3
```

#### Barème : 1,5 points

Noter positivement :

- 0,25 utilisation d'un CREATE VIEW
- 0,25 définition des attributs de la vue (dans la parenthèse, 1ère ligne)
- 0,25 clause SELECT avec les attributs et les agrégats
- 0,25 clause FROM simple ne contenant qu'une relation
- 0,25 GROUP BY avec les trois attributs
- 0,25 clause HAVING

Erreurs à pénaliser : HAVING sans GROUP BY : -0,5

(Q6.2) Donner le droit d'interrogation avec le droit de transmettre ce droit à une application identifiée par le login compareteur sur la vue que vous venez de créer.

```
GRANT SELECT ON Positionnements TO compareteur
WITH GRANT OPTION
```

#### Barème : 0,5 points

Noter positivement :

- 0,25 utilisation d'un GRANT SELECT
- 0,25 droit sur le bon objet, au bon utilisateur (ON Positionnements TO compareteur)
- 0,25 droit de transmettre le droit (WITH GRANT OPTION)

Conserver 0,75 point si aucune erreur

### Question 7 : Question de cours

Répondre par un texte court (moins de 10 lignes) à la question de cours suivante.

Que veut dire le I dans l'acronyme ACID des propriétés des transactions ? Illustrez par un exemple pourquoi cette propriété est nécessaire.

I veut dire Isolation. C'est une des quatre propriétés (ACID) des transactions.

Exemple de défaut d'isolation (tiré de wikipedia)

Supposons les deux transactions suivantes : T1 transfert 10 de A à B, T2 transfert 5 de B à A. Si les transactions s'exécutent en séquence, nous avons les actions suivantes :

1. première partie de la transaction T1 : A est réduit de 10 ;

2. seconde partie de la transaction T1 : B est augmenté de 10 ;
3. première partie de la transaction T2 : B est réduit de 5 ;
4. seconde partie de la transaction T2 : A est augmenté de 5.

Si les quatre opérations sont exécutées dans l'ordre précédent, l'isolation est assurée. Si la transaction T1 est interrompue après sa première action, le système effacera l'effet partiel de T1 avant le début de T2 et T2 s'exécutera sur des données valides.

Par contre, si les transactions tentent de s'exécuter simultanément et que la transaction T2 commence avant la fin de T1, on peut obtenir la suite d'actions suivante :

1. première partie de la transaction T1 : A est réduit de 10 ;
2. première partie de la transaction T2 : B est réduit de 5 ;
3. seconde partie de la transaction T2 : A est augmenté de 5 ;
4. seconde partie de la transaction T1 : B est augmenté de 10.

Si la transaction T1 est interrompue avant sa deuxième partie, mais après la complétion de la transaction T2, le rétablissement de A à sa valeur d'avant la transaction T1 annulera la réduction valide de A faite par la transaction T2 et laissera la base de données dans un état corrompu parce que A sera revenu à sa valeur initiale, mais B aura été augmenté de 5. Ce serait un échec d'isolation.

### **Barème : 2 points**

Noter positivement :

- I veut dire Isolation : 0,5 point
- 1,5 point pour un scénario montrant une exécution concurrente de deux transactions. L'une de ces deux transactions échoue alors qu'elle a lu une valeur obtenue par l'autre transaction. Cette dernière transaction elle est validée.