
Exercice 1

On considère un schéma local avec les deux sources `Contrat` et `Historique` contenant des informations sur des contrats relatifs à des projets.

```
Contrat(numContrat, montant, nomProjet)
Historique(numContrat, date)
```

Nous voulons développer une application pour deux catégories d'utilisateurs :

Un chercheur veut connaître pour chaque projet le contrat qui lui est associé, la date d'obtention, et le montant du financement. L'agent comptable (administrateur) veut connaître seulement le montant et la date d'un contrat.

1.1. Proposer un schéma de médiation, avec une approche de médiation qui vous semble appropriée, pour construire cette application.

Schéma global (exposé à l'utilisateur/application)

chercheurs (nomProjet, numContrat, date, montant)

administrateurs (nomProjet, date, montant)

1.2. Spécifier en SQL la construction d'un tel schéma.

```
CREATE VIEW chercheurs (nomProjet, numContrat, date, montant) AS
SELECT C.nomProjet, C.numContrat, H.date, C.montant FROM Contrat
C, Historique H
WHERE C.numContrat= H.numContrat;
```

```
CREATE VIEW administrateurs (nomProjet, date, montant) AS
SELECT C.nomProjet, H.date, C.montant FROM Contrat C,
Historique H
WHERE C.numContrat= H.numContrat;
```

1.3. Ecrire la requête qui retourne les projets dont le financement est supérieur à 50.000 Euros

```
SELECT      nomProjet, montant
FROM        administrateurs
WHERE       montant > 50000
UNION
SELECT      nomProjet, montant
FROM        chercheurs
WHERE       montant > 50000 ;
```

Remarque : les 2 vues « contiennent » les mêmes projets

1.4. Donner la traduction de cette requête sur le schéma local.

```
SELECT    nomProjet, montant  
FROM      administrateurs  
WHERE     montant > 50000
```

Elle se réécrit :

```
SELECT C.nomProjet, H.date,  
C.montant FROM Contrat C, Historique  
H WHERE C.numContrat= H.numContrat  
AND C.montant > 50000 ;
```

Contrat

- nomProjet
- numContrat
- montant

Historique

- numContrat
- date

Exercice 2

Une entreprise A vend des ordinateurs et utilise la base de données suivante pour son catalogue :

UC (numUC, processeur, vitesse, mémoire, disque) <i>Par exemple, le tuple UC (123, Core i3, 2.3, 8, 512) désigne le modèle numéro 123, ayant un processeur Intel Core i3 cadencé à 2.3 GHz, avec 8 Go de RAM et 512 Go de disque.</i>
ECRAN (numEcran, taille, resX, resY) <i>ECRAN (5521, 17, 1600, 1200) désigne un écran de 17'' de résolution maximum de 1600 x 1200.</i>

Une autre entreprise B vend des ordinateurs compacts et stocke les informations dans une seule table :

PC (numPC, processeur, mem, disque, écran) <i>Par exemple, le tuple PC (221, Core i5, 16, 256, 19) désigne un ordinateur ayant un processeur Intel Core i7, 16 Go de RAM, 256 Go de disque et un écran de 19''.</i>
--

2.1. Si l'entreprise A veut insérer dans ses relations le maximum d'informations sur les produits vendus par B, quelles seraient les instructions SQL à écrire ?

/* UC_B : vue UC sur B (données viennent de B)

```
INSERT INTO UC_B (numUC, processeur, vitesse, mémoire,
disque) VALUES (
    SELECT B.PC.numPC AS numUC, B.PC.processeur AS processeur,
    NULL, B.PC.mem AS mémoire, B.PC.disque AS disque
FROM B.PC;
)
```

Autre version

```
INSERT INTO UC_B (numUC, processeur, vitesse, mémoire, disque )
VALUES (
    SELECT B.PC.numPC AS numUC, B.PC.processeur AS processeur,
    NULL, B.PC.mem AS mémoire, B.PC.disque AS disque
FROM B.PC;
)
```

/* ECRAN_B : vue ECRAN sur B (données viennent de B)

```
INSERT INTO ECRAN_B (numEcran, taille, resX, resY)
VALUES (
    SELECT B.PC.numPC, B.PC.ecran, NULL, NULL
FROM B.PC;
)
```

2.2. Si B veut insérer dans la table PC le maximum de configurations construites à partir des paires (unité centrale, écran) vendues par A, quelles seraient les instructions SQL à écrire ?

```
CREATE VIEW PC_A (numPC, processeur, mem, disque, ecran)
AS
SELECT UC.numUC, UC.processeur, UC.memoire, UC.disque, NULL)
FROM UC
UNION
SELECT ECRAN.numEcran, NULL, NULL, NULL, ECRAN.taille
FROM ECRAN
```

2.3. Proposer un schéma de médiation pour intégrer les schémas des 2 entreprises.

Version 2.3.a Schéma de médiation = « Celui de A »

ECRAN_G et UC_G

```
CREATE VIEW UC_B (numUC, processeur, vitesse, mémoire,
disque) AS
SELECT B.PC.numPC AS numUC, B.PC.processeur AS processeur, NULL,
B.PC.mem AS mémoire, B.PC.disque AS disque FROM B.PC;
```

/* ECRAN_B : vue ECRAN sur B (données viennent de B)

```
CREATE VIEW ECRAN_B (numEcran, taille, resX, resY)
AS
SELECT B.PC.numPC, B.PC.ecran, NULL, NULL
FROM B.PC;
```

Schéma de médiation

```
ECRAN_G = ECRAN UNION ECRAN_B
UC_G = UC UNION UC_B
```

Implémentation (code SQL)

```
CREATE VIEW ECRAN_G (numEcran, taille, resX, resY)
AS
SELECT * FROM ECRAN
UNION
SELECT * FROM ECRAN_B;
```

```
CREATE VIEW UC_G (numUC, processeur, vitesse, mémoire, disque)
AS
SELECT * FROM UC
UNION
SELECT * FROM UC_B ;
```

Question 2.3.b : Le schéma global est « identique au schéma de B »

```
CREATE VIEW PC_A (numPC, processeur, mem, disque, ecran)
AS
SELECT UC.numUC, UC.processeur, UC.memoire, UC.disque, NULL)
FROM UC
UNION
SELECT ECRAN.numEcran, NULL, NULL, NULL, ECRAN.taille
FROM ECRAN
```

```
CREATE VIEW PC_B (numPC, processeur, mem, disque, ecran)
AS
SELECT * FROM PC;
```

```
CREATE VIEW PC_G (numPC, processeur, mem, disque, ecran)
AS
SELECT * FROM PC_A
UNION
SELECT * FROM PC_B;
```

2.4. Ecrire la requête Q0 qui retourne les ordinateurs ayant la plus grande capacité disque et dont le processeur tourne à une fréquence de 3,1 GHz.

On revient à notre schéma global construit « à partir de A »

Requête Q0 :

```
SELECT *
FROM UC_G
WHERE vitesse = 3.1
AND disque = ( SELECT MAX(disque)
                FROM UC_G
                WHERE vitesse = 3.1) ;
```

Question : pourquoi a-t-on besoin de préciser 2 fois `vitesse = 3.1` ?

2.5. Quelles sont les traductions de Q0 que ferait un médiateur sur les relations sources ?

```
Q0_A
SELECT  numUC
FROM    UC
WHERE   Vitesse=3.1
AND     disque = ( SELECT MAX(disque)
                   FROM UC
                   WHERE Vitesse=3.1);
```

Q0_A'

```
SELECT  numUC
FROM    UC
WHERE   Vitesse=3.1
AND     disque = ( SELECT MAX(disque)
                  FROM UC
                  WHERE Vitesse=3.1);
```

Les traductions se font par dépliage de la requête Q0 (des vues) sur les sources

```
/* Ce qui vient de A
```

```
SELECT *
FROM ( /* UC_A = A
      SELECT *
      FROM A.UC
      WHERE A.vitesse = 3.1
      AND A.disque = (SELECT MAX (disque)
                     FROM A.UC
                     AND A.vitesse = 3.1)
      )
```

```
/* ce qui vient de B
```

B n'est pas concernée car on n'a pas la vitesse dans B.

Exercice 3

On rencontre ici les problèmes (classiques) d'hétérogénéité et de qualité des données.

- On retrouve plus de régions : la Région Provinces Alpes Côte d'Azur, s'appelle désormais Région SUD mais on peut aussi trouver PACA. Ce qui fait au moins 3 entités pour la même région.
- On trouve plus de « clients » à cause des noms de clients qui peuvent être transcrits différemment selon les bases, des fautes de frappe dans les noms, etc.