

Exo	1a	1t	1s	2	3a	3t	41	42s	42a	43	44	45	46	NOTE
sur	1	2	1	1	1	2	1	1	1	2	2	3	3	21

## ISIFAR-FI&SR et Licence d'informatique – BD – LE QUICK du 22/03/2007 corrigé

Durée 45' – notes manuscrites A4 autorisées – livres interdits

### 1. Une jointure

Etant donné les tables:

R	A	B	C
	2	7	10
	4	2	3
	6	1	3

S	B	E	C
	3	Z	10
	3	Z	9
	9	Y	3

calculez la jointure

$R \bowtie_{R.A > S.B} S$	A	R.B	R.C	S.B	E	S.C	← Attributs (1pt)
	4	2	3	3	Z	10	
	4	2	3	3	Z	9	
	6	1	3	3	Z	10	
	6	1	3	3	Z	9	← Tuples (2pt)

Complétez la requête SQL définissant cette même jointure (1pt):

```
SELECT *
FROM R,S
WHERE R.A>S.B
```

### 2. Comprendre un UPDATE

Décrivez en français l'effet de l'ordre SQL suivant :

UPDATE	Produits
SET	Prix=Prix/2
WHERE	Couleur ='Marron' ;

Ça signifie (1pt) Diviser par 2 les prix de tous les produits marrons

### 3. Evaluer une requête

Pour la table

R	X	y	z
	F	4	8
	G	5	12
	H	6	4
	K	7	5

on applique la requête

```
SELECT AVG(z) AS b
FROM R
WHERE y = ANY ( SELECT z
FROM R
)
```

Quel en sera le résultat ?

result	b	← Attributs (1pt)
	10	← Tuples (2pt)

#### 4. Exprimer des requêtes

La base de données d'une agence de guides parisiens contient 5 tables de schémas suivants (les types des attributs sont évidents et omis) :

Lieu(nomLieu, adresse, telephone, prixEntree)

Itineraire(nomIti, noGuide, durée, prixIti)

Guides(noGuide, nomGuide, portable)

Contient(nomIti, nomLieu)

Connait(noGuide, nomLieu)

1. Comment afficher les noms de tous les lieux triés en ordre croissant de prix d'entrée? (1pt)

```
SELECT nomLieu
FROM Lieu
ORDER BY prixEntree;
```

2. Un touriste cherche le no de portable et le nom d'un guide qui connait bien la Tour Eiffel. Affichez (1pt)

```
SELECT portable, nomGuide
FROM Guides g, Connait c
WHERE g.noGuide=c.noGuide AND c.nomLieu='Tour Eiffel';
```

Exprimez la même requête en algèbre relationnelle (1pt)

$$R = \pi_{portable, nomGuide} (\sigma_{nomLieu=TourEiffel} (Guides \bowtie Connait))$$

3. Doubler les prix de tous les itinéraires qui contiennent la Saint-Sulpice (2pts)

```
UPDATE Itineraires
SET prixIti=prixIti*2
WHERE nomIti IN ( SELECT nomIti
                  FROM Contient
                  WHERE nomLieu='Saint-Sulpice' );
```

4. Pour chaque guide qui fait au moins 3 itinéraires afficher la durée maximale et minimale des itinéraires qu'il accompagne (2pts)

```
SELECT noGuide, MAX(duree), MIN(duree)
FROM Itineraire
WHERE true
GROUP BY noGuide
HAVING COUNT(*)>=3;
```

5. Afficher tous les itinéraires qui coutent moins cher que les lieux visités (3pts)

```
SELECT nomIti
FROM Itineraire i
WHERE prixIti < ( SELECT SUM(prixEntree)
                  FROM Lieu l, Contient c
                  WHERE l.nomLieu=c.nomLieu AND c.nomIti=i.nomIti);
```

6. Afficher tous les itinéraires accompagnés par un guide incompetent (= qui ne connait pas au moins un lieu visité) (3pts)

```
SELECT nomIti
FROM Itineraires i
WHERE EXISTS (
    (SELECT nomLieu
     FROM Contient ct
     WHERE ct.nomIti=i.nomIti)
    EXCEPT
    (SELECT nomLieu
     FROM Connait c
     WHERE ct.noGuide=i.noGuide)
);
```