

QCM

Pour chaque question le nombre de réponses est variable.

Il s'agit pour vous de trouver toutes les bonnes réponses à la question et de les cocher.

Le nombre de cases à cocher peut varier entre 0 et le nombre total de cases proposées.

Barème :

Toutes les bonnes réponses à une question : 2 points

Toutes les mauvaises réponses à une question : -2 points

Entre les deux : proportionnel au nombre de bonnes et mauvaises réponses

Absence de réponse : 0 point

La table classe a été créée par la requête suivante :

```
CREATE TABLE classe (  
  num INT PRIMARY KEY,  
  libelle VARCHAR(30) NOT NULL  
);
```

Parmi les requêtes suivantes, cochez celles qui sont
SYNTAXIQUEMENT correctes

Cochez la ou les réponses

- SELECT num FROM classe;
- SELECT 'num' FROM classe;
- SELECT ALL num FROM classe;
- SELECT DISTINCT num FROM classe;
- SELECT 2*num FROM classe;
- SELECT num, libelle FROM classe;
- SELECT num numero FROM classe;
- Je préfère ne pas répondre à cette question

Parmi les requêtes suivantes, cochez celles qui sont
SYNTAXIQUEMENT correctes

Cochez la ou les réponses

- SELECT count(*) FROM classe c1, classe c2;
- SELECT count(*) FROM classe c1 JOIN classe c2;
- SELECT count(*) FROM classe c1 NATURAL JOIN classe c2;
- SELECT count(*) FROM classe c1 LEFT JOIN classe c2 USING (num);
- SELECT count(*) FROM classe c1 LEFT JOIN classe c2 ON c1.num=c2.num;

- SELECT count(*) FROM classe AS c1 , classe AS c2;
- SELECT count(*) FROM classe AS c1 , c2;
- Je préfère ne pas répondre à cette question

Les tables societe et enr ayant été créées par les requêtes suivantes :

```
CREATE TABLE societe (  
  id INT PRIMARY KEY,  
  nom VARCHAR(40) NOT NULL,  
  ville VARCHAR(20),  
  pays CHAR(2) NOT NULL REFERENCES pays  
);  
CREATE TABLE enr (  
  marque INT NOT NULL REFERENCES marque,  
  num INT NOT NULL,  
  pays CHAR(2) NOT NULL REFERENCES pays,  
  deposant INT NOT NULL REFERENCES societe,  
  date_enr DATE NOT NULL,  
  CONSTRAINT cle_enr PRIMARY KEY (num,pays)  
);
```

La requête suivante est elle syntaxiquement correcte ?

```
SELECT societe.id FROM enr, societe WHERE societe.id = enr.deposant  
GROUP BY enr.deposant;
```

Cochez la ou les réponses

- OUI
- NON
- Je préfère ne pas répondre à cette question

Quelles sont parmi les requêtes suivantes, celles qui affichent le nom et la classe des marques en regroupant les marques par classe, et à l'intérieur d'une classe en affichant les marques par ordre alphabétique (de A à Z)

Cochez la ou les réponses

- SELECT DISTINCT nom, classe FROM marque ORDER BY nom, classe;
- SELECT DISTINCT nom, classe FROM marque ORDER BY classe, nom;
- SELECT DISTINCT nom, classe FROM marque ORDER BY classe ASC , nom;
- SELECT DISTINCT nom, classe FROM marque ORDER BY nom, classe DESC;
- SELECT DISTINCT nom, classe FROM marque ORDER BY 1,2;
- SELECT DISTINCT nom, classe FROM marque ORDER BY 2,1;

Je préfère ne pas répondre à cette question

La table marque a été créée par la requête

```
CREATE TABLE marque (  
  id INT NOT NULL PRIMARY KEY,  
  nom VARCHAR(30) NOT NULL,  
  classe INT NOT NULL REFERENCES classe,  
  pays CHAR(2) NOT NULL REFERENCES pays,  
  prop INT NOT NULL REFERENCES societe  
);
```

et des tuples ont été insérés dans cette table.

Les deux requêtes suivantes renvoient-elles toujours le même résultat ?

```
SELECT classe FROM marque;
```

```
SELECT classe FROM marque UNION SELECT classe FROM marque;
```

Cochez la ou les réponses

- OUI
- NON
- Je préfère ne pas répondre à cette question

Cochez les phrases qui sont exactes

Cochez la ou les réponses

- Un attribut qui apparaît dans le SELECT doit aussi apparaître après le WHERE
- Un attribut qui apparaît dans le WHERE doit aussi apparaître après le SELECT
- Les comparaisons de chaînes de caractères sont sensibles à la casse
- Je préfère ne pas répondre à cette question

Cochez un agrégat s'il peut être utilisé sur des colonnes de n'importe lequel des trois types entier, chaîne de caractères, date.

Cochez la ou les réponses

- AVG
- COUNT
- MAX
- MIN
- SUM

Je préfère ne pas répondre à cette question

Parmi les assertions suivantes, cochez celles qui sont exactes

Cochez la ou les réponses

- On ne peut pas utiliser une jointure naturelle lorsque les tables contiennent des attributs de même nom sur lesquels on ne veut pas faire la jointure
- La clause "using" d'une jointure naturelle ne joindra pas des lignes où l'attribut correspondant contient une valeur null
- lorsque le même nom d'attribut est utilisé dans plusieurs tables lors d'un calcul de jointure, le nom de l'attribut doit être précédé du nom de la table
- Je préfère ne pas répondre à cette question

Cochez les descriptions exactes de ce que calcule la requête suivante :

```
SELECT classe, S.nom, S.pays, COUNT (*)
FROM marque AS M1, societe AS S
WHERE M1.prop = S.id
GROUP BY classe, S.nom, S.pays HAVING
COUNT(*) >= ALL
(SELECT COUNT(*) FROM marque AS M2 GROUP BY classe, prop);
```

Cochez la ou les réponses

- Le(s) propriétaire(s) qui possède(nt) le plus de marques dans toutes les classes
- Pour l'ensemble des classes, le(s) propriétaire(s) qui possède(nt) le plus grand nombre de marques dans une seule et même classe
- Pour chaque classe, le(s) propriétaire qui possède (nt) le plus de marques
- Je préfère ne pas répondre à cette question

Parmi les requêtes suivantes, cochez celles qui calculent les sociétés dont les marques sont toutes dans la classe 14

Cochez la ou les réponses

- SELECT DISTINCT S.nom, S.pays FROM societe S, marque M WHERE S.id = M.prop AND M.classe <>14;
- SELECT S.nom, S.pays FROM societe S, marque M WHERE S.Id=M.prop EXCEPT SELECT S.nom, S.pays FROM societe S, marque M WHERE S.Id=M.prop AND M.classe = 14;

Je préfère ne pas répondre à cette question

nom

Cochez la ou les réponses

SI4

MAM4

<< Précédent

Dernier >>

[Sortir et effacer le questionnaire]