

Statut	Terminé
Commencé	mercredi 19 mars 2025, 08:29
Terminé	mercredi 19 mars 2025, 09:44
Durée	1 heure 15 min
Note	9,5 sur 25,0 (38%)

Question 1

Terminé

Note de 2,5 sur 5,0

En utilisant les requêtes imbriquées et sans utiliser de jointure, afficher les candidats (initiale du prénom, nom et numéro de téléphone) qui n'ont fait aucune offre de montant inférieur à 150000 € depuis le 1^{er} août 2012.

Le résultat sera présenté comme ci-dessous :

CANDIDAT	TELEPHONE
P. Paul	087/45.56.52
J. Jacques	087/22.26.45
V. Véronique	0495/12.13.54
P. Pierre	063/46.56.59
T. Thierry	04/221.45.98

```
SELECT SUBSTR(prenom,1,1) || '. ' || nom as candidat, telephone
FROM candidat
WHERE id_candidat = ANY (
    SELECT id_candidat
    FROM offre o
    WHERE o.montant > 150000 AND date_offre > ANY (
        SELECT o2.date_offre
        FROM offre o2
        WHERE date_offre > TO_DATE('01/08/2012', 'DD/MM/YYYY'
    )));
```

Commentaire :

Erreur de conception des requêtes imbriquées : -2.5pts (*qui n'ont fait aucune offre* ==> **NOT IN**)

Ta requête sélectionne les candidats ayant fait au moins une offre de plus de 150000 postérieure au 1/08/2012

WHERE id_candidat = ANY (...) => Filtre les candidats ayant un identifiant présent dans la sous-requête

De plus, la 2e partie de la requête est inutile => WHERE o.montant > 150000 AND date_offre > TO_DATE('01/08/2012', 'DD/MM/YYYY');

Exemple de solution :

```
SELECT SUBSTR(c.prenom,1,1) || '. ' || c.prenom AS CANDIDAT, c.telephone
```

```
FROM CANDIDAT c
```

```
WHERE c.id_candidat NOT IN (SELECT o.id_candidat
```

```
FROM Offre o
```

```
WHERE o.montant < 150000 AND o.date_offre > TO_DATE('01/08/2012','ddmmyyyy');
```

WHERE c.id_candidat NOT IN (...) => exclut les candidats présents dans la sous-requête => candidats ayant fait une **offre < 150000 €** après le **01/08/2012**

Question **2**

Terminé

Note de 5,0 sur 5,0

En utilisant un opérateur ensembliste, afficher l'identifiant, l'adresse (rue et localité) et la description des biens de la catégorie 'Maison' qui ont fait l'objet d'une offre provenant d'un candidat habitant Liège, Spa ou Verviers.

Le résultat sera formaté comme suit :

ID_BIEN	ADRESSE	DESCRIPTION
1	rue des clarisses - Esneux	Superbe maison 3 façades avec gros volumes idéale pour profession libérale ou pour grande famille
5	impasse des Ursulines - Liège	Maison 3 façades en bon état proche du centre ville.
14	place du marché - Ans	Composition: Immeuble de rapport composé de trois appartements 1 chambre et un appartement 2 chambres.
17	allée des braconniers - Liège	Maison bourgeoise 4 façades, sit. exceptionnelle dans l cadre de verdure romantique à 1 min. à pied de

```
SELECT b.id_bien, b.rue || ' ' || b.localite as adresse, b.description
FROM bien b
JOIN categorie_bien cb ON b.id_catbien = cb.id_catbien
WHERE cb.description = 'Maison'

INTERSECT

SELECT b2.id_bien, b2.rue || ' ' || b2.localite as adresse, b2.description
FROM bien b2
JOIN offre o ON b2.id_bien = o.id_bien
JOIN candidat c ON o.id_candidat = c.id_candidat
WHERE c.localite IN('Liège','Spa','Verviers');
```

Commentaire :

Question 3

Terminé

Note de 2,0 sur 5,0

On voudrait obtenir une liste des villas, bungalow, maisons et château. Cette liste affichera le nom et le prénom du client, la catégorie de bien ainsi qu'un message. Pour le message, on affichera 'Possibilité de négocier' s'il n'y qu'une salle de bain et que le bien a été mis en vente il y a plus d'un an ; on affichera 'Bien luxueux - Possibilité de négocier' s'il y a plus d'une salle de bain et que le bien a été mis en vente il y a plus d'un an ; on affichera 'Nouveauté ! Bien en vente depuis peu' si le bien est en vente depuis moins de trois mois.

Le résultat sera formaté comme suit :

CLIENT	CATEGORIE_BIEN	MESSAGE
Barvaux Marc	Maison	Possibilité de négocier
Barvaux Marc	Maison	Possibilité de négocier
Pirotton Luc	Bungalow	Possibilité de négocier
Desonnav Vincianne	Château	Bien luxueux - Possibilité de négocier
Beaujean Emile	Maison	Possibilité de négocier
Rensonnet Pascal	Maison	Bien luxueux - Possibilité de négocier
Offermans Dany	Maison	Possibilité de négocier
Degey Etienne	Maison	Possibilité de négocier

```
SELECT c.nom || ' ' || prenom as client, cb.description as categorie_bien, ' ' as message
FROM categorie_bien cb
JOIN bien b ON cb.id_catbien = b.id_catbien
JOIN client c ON b.id_client = c.id_client
WHERE cb.description IN('Villa','Bungalow','Maison','Château');

-- le code ci-dessus s'exécute, je n'ai pas mis le switch dedans, car
-- je ne me souviens plus de la syntaxe du switch (pas noté sur mon aide mémoire)
-- mais je ne pense pas être très éloigné de la réponse pour le message :
SWITCH
CASE b.nb_sdb = 1 AND b.date_mise_en_vente < ADD_MONTHS(b.date_mise_en_vente, -12) THEN 'Possibilité de négocier'
CASE b.nb_sdb > 1 THEN 'Bien luxueux - Possibilité de négocier',
CASE b.date_mise_en_vente > ADD_MONTHS(b.date_mise_en_vente, -3) THEN 'Nouveauté !'
AS message
END
```

Commentaire :

Non-utilisation ou utilisation erronée de CASE...WHEN : -3pts

Erreur de syntaxe (exemple à placer dans le memo ;))

Attention : erreur dans de logique :

- 'Possibilité de négocier' s'il n'y qu'une salle de bain et que le bien a été mis en vente il y a plus d'un an

CASE b.nb_sdb = 1 AND **b.date_mise_en_vente < ADD_MONTHS(b.date_mise_en_vente, -12)** THEN 'Possibilité de négocier',

b.date_mise_en_vente < ADD_MONTHS(b.date_mise_en_vente, -12) =>

Cette expression est toujours fausse et même illogique. Tu compares une date à elle-même moins 12 mois -> jamais vrai.

Tu aurais dû utiliser : b.date_mise_en_vente < ADD_MONTHS(SYSDATE, -12) ou

MONTHS_BETWEEN(SYSDATE,b.date_mise_en_vente) >12

- on affichera 'Bien luxueux - Possibilité de négocier' s'il y a plus d'une salle de bain et que le bien a été mis en vente il y a plus d'un an

CASE b.nb_sdb > 1 THEN 'Bien luxueux - Possibilité de négocier',

Tu as oublié la condition mis en vente il y a plus d'un an

- on affichera 'Nouveauté ! Bien en vente depuis peu' si le bien est en vente depuis moins de trois mois.

```
CASE b.date_mise_en_vente > ADD_MONTHS(b.date_mise_en_vente, -3) THEN 'Nouveauté !'
```

Cette expression est toujours vraie. Tu compares une date avec elle-même moins 3 mois.

Tu aurais dû utiliser MONTHS_BETWEEN(SYSDATE, b.date_mise_en_vente) < 3

Exemple de solution :

```
SELECT c.nom || ' ' || c.prenom AS CLIENT, cb.description AS CATGORIE_BIEN,  
CASE  
  WHEN b.nb_sdb = 1 AND MONTHS_BETWEEN(SYSDATE,b.date_mise_en_vente) >12 THEN 'Possibilité de négociier'  
  WHEN b.nb_sdb > 1 AND MONTHS_BETWEEN(SYSDATE,b.date_mise_en_vente) >12 THEN 'Bien luxueux - Possibilité de  
négociier'  
  WHEN MONTHS_BETWEEN(SYSDATE,b.date_mise_en_vente) <3 THEN 'Nouveauté ! Bien en vente depuis peu'  
END AS Message  
FROM Client c  
JOIN Bien b ON b.id_client = c.ID_CLIENT  
JOIN Categorie_bien cb ON b.id_catbien = cb.ID_CATBIEN  
WHERE cb.description IN ('Villa','Maison','Château','Bungalow')
```

Question 4

Terminé

Note de 0,0 sur 5,0

On veut lister les messages qui n'ont pas été lus par au moins deux vendeurs (destinataires du message). Pour ces messages, on affichera le mois de publication, l'identifiant du message et le nombre de vendeurs (destinataires du message) n'ayant pas lu le message. Le résultat sera trié par ordre croissant du nombre de vendeurs n'ayant pas lu le message. Le résultat sera formaté comme suit :

MOIS	ID_MESSAGE	NBVENDEURS
10-2012	24	10
09-2012	14	10
08-2012	6	8
07-2012	1	8
08-2012	7	2

La première ligne du résultat ci-dessus indique donc qu'en octobre 2012, le message dont l'ID est 24 n'a pas été lu par 10 vendeurs à qui ce message était destiné.

```
SELECT ' ' as mois, id_message, SUM(id_vendeur) as nbvendeurs
FROM alerter
WHERE statut = 'N'
GROUP BY id_message
ORDER BY nbvendeurs;
```

Commentaire :

La requête ne répond pas à l'énoncé.

Pour ces messages, on affichera le mois de publication, l'identifiant du message et le nombre de vendeurs (destinataires du message) n'ayant pas lu le message.

=> tu dois afficher des données provenant de la table message et alerter => il te manque une jointure

nombre de vendeurs

=> la fonction à utiliser est count()

n'ont pas été lus par au moins deux vendeurs

=> tu dois avoir une condition qui porte la fonction count()

Exemple de solution :

```
SELECT TO_CHAR(m.date_message, 'MM-YYYY') AS MOIS, a.id_message, COUNT(*) AS NBVENDEURS
FROM ALERTER a
JOIN MESSAGE m ON m.id_message = a.id_message
WHERE a.statut = 'N'
GROUP BY TO_CHAR(m.date_message, 'MM-YYYY'), a.id_message
HAVING COUNT(*) > 1
ORDER BY COUNT(*) DESC
```

Question **5**

Terminé

Note de 0,0 sur 5,0

Pour toutes les catégories de biens, on veut afficher le nombre d'offres reçues. Si, pour une catégorie de bien, aucune offre n'a été reçue, elle doit apparaître dans le résultat avec un nombre d'offres égal à zéro. Le résultat sera trié par ordre alphabétique de la catégorie de bien. Le résultat sera formaté comme suit :

CATEGORIE_BIEN	NB_OFFRES
Appartement	0
Bien exceptionnel	0
Bungalow	2
Château	0
Ferme	5
Fermette	3
Maison	0

```
SELECT cb.description as categorie_bien, SUM(o.id_bien) as nb_offres
FROM categorie_bien cb
LEFT JOIN bien b ON cb.id_catbien = b.id_catbien
LEFT JOIN offre o ON b.id_bien = o.id_bien
GROUP BY cb.description;
```

Commentaire :

Tri manquant ou erroné : -2pts

Le résultat sera trié par ordre alphabétique de la catégorie de bien.

Erreur Groupement et/ou de fonction de calcul : -4pts

nombre d'offres reçues => count()

[◀ Précisions sur l'interro certificative de labo BD - mars 2025](#)

Aller à...

[QCM BD- système de cotation ▶](#)