

Statut	Terminée
Commencé	mercredi 16 avril 2025, 09:03
Terminé	mercredi 16 avril 2025, 09:57
Durée	54 min 50 s
Note	7,0 sur 22,0 (31,8%)

Question 1

Terminé

Note de 3,0 sur 7,0

Créez une vue VW_VENDEUR_NB_BIENS renvoyant, pour tous les vendeurs, le nombre de biens qu'ils gèrent par catégorie. Si un vendeur ne gère aucun bien, il apparaîtra quand même dans le résultat avec <null> pour la catégorie et 0 pour le nombre de biens. Le résultat affichera le nom du vendeur suivi de son prénom et de son pseudo entre parenthèses, le nom de la catégorie du bien et le nombre de biens.

Cette vue est-elle "modifiable" ? Répondez par oui ou par non et expliquez pourquoi (en texte libre en-dessous de votre proposition de requête)

L'exécution de la requête

```
SELECT *
FROM VW_VENDEUR_NB_BIENS
ORDER BY VENDEUR
```

renverra les données selon le format suivant :

VENDEUR	CATÉGORIE	NOMBREBIENS
Baltus Claudine (cbaltus)	Villa	2
Baltus Claudine (cbaltus)	Maison	3
Baltus Claudine (cbaltus)	Fermette	2
Baltus Claudine (cbaltus)	Maison de maître	3
Barmik Laura (lbarmik)	(null)	0
Becot Stéphane (sbecot)	Maison	4

```
CREATE VIEW VW_VENDEUR_NB_BIENS
AS
SELECT v.nom || ' ' || v.prenom || ' (' || v.pseudo || ' )' as vendeur,
       cb.description as catégorie,
       count(cb.description) as nombrebiens
FROM vendeur v
LEFT JOIN gerer g ON v.id_vendeur = g.id_vendeur
LEFT JOIN bien b ON g.id_bien = b.id_bien
LEFT JOIN categorie_bien cb ON b.id_catbien = cb.id_catbien
GROUP BY v.nom, v.prenom, v.pseudo, cb.description, g.id_bien
order by vendeur

-- Non, je ne pense pas car de 1 il y a par exemple des jointures et des group by mais
-- en plus ici on demande le nombre de biens plutôt qu'une liste, ce qui évoque quelque chose
-- qu'on ne cherche pas à modifier
```

Commentaire :

FEEDBACK GENERAL

Les requêtes avec regroupement doivent être revues. Certains concepts ne sont pas encore acquis.

N'hésite pas à prendre rdv avec un membre de l'équipe BoostUp pour éventuellement ré-expliquer certains concepts.

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

Le groupement n'est pas correct : En groupant aussi par g.id_bien, on aura toujours un comptage qui vaut 1, cela empêche le groupement par vendeur et catégorie.

Exemple de solution :

```
CREATE OR REPLACE VIEW Vendeur_Nb_Biens  
AS  
SELECT nom || ' ' || prenom || '(' || pseudo || ')' AS Vendeur, cb.description AS Catégorie, COUNT(b.id_bien) AS NombreBiens  
FROM VENDEUR v  
LEFT OUTER JOIN GERER g ON v.ID_VENDEUR = g.ID_VENDEUR  
LEFT OUTER JOIN BIEN b ON g.ID_BIEN = b.ID_BIEN  
LEFT OUTER JOIN CATEGORIE_BIEN cb ON b.ID_CATBIEN = cb.ID_CATBIEN  
GROUP BY nom || ' ' || prenom || '(' || pseudo || ')', cb.description
```

Question **2**

Terminé

Note de 4,0 sur 10,0

On vous demande d'écrire un trigger qui assure qu'à tout moment, un bien appartenant à la catégorie de bien « Villa » possède 4 façades et au moins une salle de bain.

Remarque : pour éviter tout problème lié aux éventuels triggers fait en laboratoire sur la même table, vous pouvez les supprimer (click-droit + delete).

```
CREATE TRIGGER trg_villa_needs_4_facades ON bien
AFTER INSERT
AS
BEGIN

IF EXISTS(SELECT * FROM INSERTED i WHERE i.id_catbien <> (SELECT cb.id_catbien FROM categorie_bien cb WHERE cb.description = 'Villa')) RETURN

    DECLARE @typeBien          VARCHAR(100)
    DECLARE @idBien            INT
    DECLARE @nbFacades         INT
    DECLARE @nbSdb             INT

    DECLARE CRSbiens CURSOR
    FOR
        SELECT i.id_bien, i.nb_facade, i.nb_sdb,
               (SELECT cb.description FROM categorie_bien cb WHERE cb.id_catbien = i.id_catbien)
        FROM INSERTED i

    OPEN CRSbiens
    FETCH CRSbiens INTO @idBien, @nbFacades, @nbSdb, @typeBien

    WHILE @@fetch_status = 0
    BEGIN
        if (@nbFacades <> 4 OR @nbSdb < 1)
        BEGIN
            RAISERROR ('[trg_villa_needs_4_facades] Une villa doit avoir 4 façades et au moins une salle de bain')
            ROLLBACK TRANSACTION
            RETURN
        end
        FETCH CRSbiens INTO @idBien, @nbFacades, @nbSdb, @typeBien
    end
    CLOSE CRSbiens
    DEALLOCATE CRSbiens
end
```

Commentaire :

Ton trigger met en place un curseur mais tu ne précises pas qu'il doit s'exécuter après un update. Un oubli ?

AFTER INSERT, **UPDATE**

Ta vérification de début de trigger n'est pas optimale :

```
IF EXISTS(SELECT * FROM INSERTED i WHERE i.id_catbien <> (SELECT cb.id_catbien
FROM categorie_bien cb WHERE cb.description = 'Villa')) RETURN
```

Si on utilise la requête : update bien set nb_sdb = 0 ; Le trigger ne réalise pas de vérification

Curseur : tu récupères la description de la catégorie du bien mais elle ne sert à rien

=> ton curseur devrait uniquement récupérer les tuples qui sont en lien avec la catégorie 'Villa'

Gestion de la boucle while => ok

Erreur au niveau de la récupération des données concernées (via INSERTED) : -5 pts

Optimisation en début de curseur non adéquate : -1

Question 3

Terminé

Note de 0,0 sur 5,0

On vous demande d'afficher, par vendeur (prénom + nom), le nombre total de façades des biens dont ils s'occupent pour autant que ce nombre soit supérieur à 4 et que les vendeurs aient un numéro de téléphone commençant par '0474' ou '0475'.

Le résultat sera trié dans l'ordre décroissant du nombre total de façades.

Le résultat sera présenté comme ci-dessous :

Vendeur	Total_facades
Martine Pottier	19
Claudine Baltus	15
Luc Cront	12
Stéphane Becot	11

```
SELECT v.prenom || ' ' || v.nom as vendeur,
       SUM(b.nb_facade)
FROM vendeur v
JOIN gerer g ON v.id_vendeur = g.id_vendeur
JOIN bien b ON b.id_bien = g.id_bien
WHERE b.nb_facade >= 4 AND v.telephone LIKE '0474%' OR v.telephone LIKE '0475%'
GROUP BY v.prenom, v.nom, nb_facade
ORDER BY nb_facade DESC
```

Commentaire :

- Mauvais formatage du résultat (renommage colonnes) : -1pt
- Mauvaise compréhension de l'énoncé : *le nombre total de façades des biens dont ils s'occupent pour autant que ce nombre soit supérieur à 4* => on a donc une condition sur le total : n'afficher que les vendeurs dont la somme des façades > 4
La clause porte donc sur la fonction -> having
Erreur clause HAVING ou manquante : -4pts

Exemple de solution :

```
SELECT v.prenom || ' ' || v.nom AS Vendeur, SUM(b.nb_facade) AS Total_facades
FROM Vendeur v
JOIN Gerer g ON v.id_vendeur = g.id_vendeur
JOIN Bien b ON g.id_bien = b.id_bien
WHERE v.telephone LIKE '0474%' OR v.telephone LIKE '0475%'
GROUP BY v.nom, v.prenom
HAVING SUM(b.nb_facade) >= 5
ORDER BY SUM(b.nb_facade) DESC
```

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

Aller à...

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