

INFO-H-303 - Projet 08/09 - Requêtes

Requête 1

Tous les utilisateurs qui apprécient au moins 3 établissements que l'utilisateur "Brenda" apprécie.

Algèbre

$$\begin{aligned} Brenda &\leftarrow \pi_{Nom}(\sigma_{Score>3 \wedge Identifiant='Brenda'}(Commentaire)) \\ Autres &\leftarrow \pi_{Nom, Identifiant}(\sigma_{Score>3 \wedge Identifiant \neq 'Brenda'}(Commentaire)) \\ Communs &\leftarrow Brenda * Autres \\ 3Communs &\leftarrow \alpha_{Nom:N1}(Communs) * \alpha_{Nom:N2}(Communs) * \alpha_{Nom:N2}(Communs) \\ Reponse &\leftarrow \pi_{Identifiant}(\sigma_{N1 \neq N2 \wedge N1 \neq N3 \wedge N2 \neq N3}(3Communs)) \end{aligned}$$

Calcul tuple

$$\{c. Identifiant \mid Commentaire(c) \wedge c. Identifiant \neq 'Brenda' \wedge c. Score > 3 \wedge \exists c_1, c_2, c_3, c_4, c_5 (Commentaire(c_1) \wedge Commentaire(c_2) \wedge Commentaire(c_3) \wedge Commentaire(c_4) \wedge Commentaire(c_5) \wedge c_1. Identifiant = 'Brenda' \wedge c_2. Identifiant = 'Brenda' \wedge c_3. Identifiant = 'Brenda' \wedge c_1. Identifiant = c. Identifiant \wedge c_2. Identifiant = c. Identifiant \wedge c_1. Nom = c. Nom \wedge c_2. Nom = c_4. Nom \wedge c_3. Nom = c_5. Nom \wedge c_1. Nom \neq c_2. Nom \wedge c_2. Nom \neq c_3. Nom \wedge c_1. Nom \neq c_3. Nom \wedge c_1. Score > 3 \wedge c_2. Score > 3 \wedge c_3. Score > 3 \wedge c_4. Score > 3 \wedge c_5. Score > 3)\}$$

SQL

```
SELECT *
FROM Utilisateur user
WHERE exists (SELECT com_user. Identifiant
              FROM Commentaire com_user, Commentaire com_brenda
              WHERE com_brenda. Identifiant = 'Brenda'
                  AND com_user. Identifiant = user. Identifiant
                  AND com_brenda. Nom = com_user. Nom
                  AND com_brenda. Score >= 4
                  AND com_user. Score >= 4
              GROUP BY com_user. Identifiant
              HAVING count(*) >=3)
```

Requête 2

Tous les établissements qu'apprécie au moins un utilisateur qui apprécie tous les établissements que "Brenda" apprécie.

Algèbre

$$\begin{aligned} Brenda &\leftarrow \pi_{Nom}(\sigma_{Score>3 \wedge Identifiant='Brenda'}(Commentaire)) \\ Autres &\leftarrow \pi_{Nom, Identifiant}(\sigma_{Score>3 \wedge Identifiant \neq 'Brenda'}(Commentaire)) \end{aligned}$$

$MemersGouts \leftarrow Autres/Brenda$
 $Reponse \leftarrow \pi_{Nom}(\sigma_{Score>3}(Commentaire) * Memesgouts)$

Calcul tuple

$\{c.Nom \mid Commentaire(c) \wedge c.Score > 3 \wedge c.Identifiant \neq 'Brenda' \wedge$
 $\forall c_1(Commentaire(c_1) \wedge c_1.Score > 3 \wedge c_1.Identifiant = 'Brenda') \rightarrow \exists c_2(Commentaire(c_2) \wedge$
 $c_2.Score > 3 \wedge c.Identifiant = c_2.Identifiant \wedge c_1.Nom = c_2.Nom)\}$

SQL

```

SELECT *
FROM Etablissement etab, Commentaire com
WHERE etab.Nom = com.Nom
      AND com.Score >=4
      AND NOT EXISTS(SELECT *
                     FROM Commentaire com_brenda
                     WHERE com_brenda.Identifiant = 'Brenda'
                           AND com_brenda.Score >= 4
                           AND NOT EXISTS (SELECT *
                                           FROM Commentaire com_user
                                           WHERE com_user.Identifiant = com.Identifiant
                                               AND com_user.Nom = com_brenda.Nom
                                               AND com_user.Score >= 4))

```

Requête 3

Tous les établissements pour lesquels il y a au plus un commentaire.

Algèbre

$1Com \leftarrow \pi_{Identifiant,Nom,Date}(Commentaire)$
 $2Com \leftarrow \alpha_{Identifiant:u1,Date:d1}(1Com) * \alpha_{Identifiant:u2,Date:d2}(1Com)$
 $2ComDif \leftarrow \sigma_{u1 \neq u2 \vee d1 \neq d2}(2Com)$
 $Reponse \leftarrow \pi_{Nom}(Etablissement) - \pi_{Nom}(2ComDif)$

Calcul tuple

$\{e.Nom \mid Etablissement(e) \wedge \forall c(Commentaire(c) \wedge c.Nom = e.Nom \rightarrow \nexists c_2(Commentaire(c_2) \wedge$
 $c_2.Nom = c.Nom \wedge (c_2.Identifiant \neq c.Identifiant \vee c_2.Date \neq c.Date)))\}$

Autre solution :

$\{e.Nom \mid Etablissement(e) \wedge \exists c(Commentaire(c) \wedge c.Nom = e.Nom \wedge \nexists c_2(Commentaire(c_2) \wedge$
 $c_2.Nom = c.Nom \wedge (c_2.Identifiant \neq c.Identifiant \vee c_2.Date \neq c.Date))) \vee \nexists c_3(Commentaire(c_3) \wedge$
 $c_3.Nom = e.Nom)\}$

SQL

```
SELECT etab.Nom
FROM Etablissement etab
WHERE (SELECT COUNT(*)
       FROM Commentaire com
       WHERE com.Nom=etab.Nom)<=1
```

Requête 4

La liste des administrateurs n'ayant pas commenté tous les établissements qu'ils ont créés.

Algèbre

$$Etab \leftarrow \pi_{Identifiant, Nom}(\alpha_{Createur:Identifiant}(Etablissement))$$
$$Reponse \leftarrow \pi_{Identifiant}(Etab - \pi_{Identifiant, Nom}(Commentaire))$$

Calcul tuple

$$\{e.Createur \mid Etablissement(e) \wedge \nexists c(Commentaire(c) \wedge c.Nom = e.Nom \wedge c.Identifiant = e.Createur)\}$$

SQL

```
SELECT admin.*
FROM Administrateur admin
WHERE EXISTS (SELECT *
             FROM etablissement etab
             WHERE etab.Createur = admin.Identifiant
             AND NOT EXISTS (SELECT *
                            FROM commentaire com
                            WHERE admin.Identifiant = com.Identifiant
                            AND com.Nom = etab.Nom))
```

Requête 5

La liste des établissements ayant au minimum trois commentaires, classée selon la moyenne des scores attribués.

SQL

```
SELECT c.Nom, AVG(c.Score) as moyenne
FROM Etablissement e, Commentaire c
WHERE c.Nom = e.Nom
GROUP BY c.Nom
HAVING COUNT(*) > 2
ORDER BY moyenne DESC
```

Requête 6

La liste des labels étant appliqués à au moins 5 établissements, classée selon la moyenne des scores des établissements ayant ce label.

SQL

```
SELECT Intitule
FROM (SELECT DISTINCT Intitule, Nom
      FROM estLabellise) as labels,
      (SELECT Nom, AVG(Score) as t1
      FROM Commentaire
      GROUP BY Nom) as t2
WHERE t1.Nom = t2.Nom
GROUP BY Intitule
HAVING COUNT(*) >=5
ORDER BY AVG(moyenne) DESC
```