

INFO-H-303 - Projet 09/10 - Requêtes

Requête 1

Les bleus : Tous les utilisateurs qui ont au plus 2 amis.

Algèbre

$$\begin{aligned} EstAmi'(u1, u2) &\leftarrow \pi_{utilisateur1, utilisateur2}(\sigma_{date \neq NULL}(EstAmi)) \\ EstAmi'' &\leftarrow \alpha_{u1:u2, u2:u1}(EstAmi') \cup EstAmi' \\ 3Amis &\leftarrow \alpha_{u1:u1, u2:ami1}(EstAmi'') * \alpha_{u1:u1, u2:ami2}(EstAmi'') * \\ &\quad \alpha_{u1:u1, u2:ami3}(EstAmi'') \\ 3AmisDif &\leftarrow \sigma_{ami1! = ami2 \wedge ami1! = ami3 \wedge ami3! = ami2}(3Amis) \\ Reponse &\leftarrow \pi_{email}(Utilisateur) - \pi_{u1}(3AmisDif) \end{aligned}$$

Calcul tuple

$$\begin{aligned} EstAmi'(u1, u2) &\leftarrow \{u1.email, u2.email \mid Utilisateur(u1) \wedge Utilisateur(u2) \wedge \\ &\quad \exists a(a.date \neq null \wedge \\ &\quad ((a.utilisateur1 = u1.email \wedge a.utilisateur2 = u2.email) \vee \\ &\quad (a.utilisateur1 = u2.email \wedge a.utilisateur2 = u1.email)))\} \\ Reponse &\leftarrow \{u.email \mid Utilisateur(u) \wedge \nexists a1, a2, a3(\\ &\quad EstAmi'(a1) \wedge EstAmi'(a2) \wedge EstAmi'(a3) \\ &\quad a1.u1 = u.email \wedge a2.u1 = u.email \wedge a3.u1 = u.email \wedge \\ &\quad a1.u2 \neq a2.u2 \wedge a1.u2 \neq a3.u2 \wedge a3.u2 \neq a2.u2)\} \end{aligned}$$

SQL

```
SELECT u.email
FROM Utilisateur u, EstAmi a
WHERE (u.email = a.utilisateur1 OR
```

```

        u.email = a.utilisateur2)
    AND a.date != NULL
GROUP BY u.email
HAVING COUNT(*) <= 2

```

Requête 2

Suggestion de flux : La liste des flux auxquels a souscrit au moins un utilisateur qui a souscrit à au moins deux flux auxquels X a souscrit.

Algèbre

$$\begin{aligned}
 FluxDeX &\leftarrow \pi_{flux}(\sigma_{utilisateur='X'}(Suit)) \\
 2FluxDif &\leftarrow \sigma_{flux1 \neq flux2}(\alpha_{flux:flux1}(Suit * FluxDeX) * \alpha_{flux:flux2}(Suit * FluxDeX)) \\
 Reponse &\leftarrow \pi_{flux}(\pi_{utilisateur}(2FluxDif) * Suit) - FluxDeX
 \end{aligned}$$

Calcul tuple

$$\{s.flux \mid Suit(s) \wedge \exists s1, s2, s3, s4(Suit(s1) \wedge Suit(s2) \wedge Suit(s3) \wedge Suit(s4) \\
 s1.utilisateur = s.utilisateur \wedge s2.utilisateur = s.utilisateur \wedge \\
 s1.flux = s3.flux \wedge s2.flux = s4.flux \wedge \\
 s3.utilisateur = 'X' \wedge s4.utilisateur = 'X')\}$$

SQL

```

SELECT DISTINCT s.flux
FROM   Suit s
WHERE  s.utilisateur != 'X'
      AND EXISTS (SELECT s2.utilisateur
                  FROM   Suit s1, Suit s2
                  WHERE  s1.utilisateur = 'X'
                        AND s2.utilisateur = s.utilisateur
                        AND s1.flux = s2.flux
                  GROUP BY s2.utilisateur
                  HAVING COUNT(*) >=2
        )

```

Requête 3

Nettoyage de flux : La liste des flux auxquels X a souscrit, auxquels aucun de ses amis n'a souscrit et duquel il n'a partagé aucune publication.

Algèbre

$$\begin{aligned} EstAmi'(u1, u2) &\leftarrow \pi_{utilisateur1, utilisateur2}(\sigma_{date \neq NULL}(EstAmi)) \\ EstAmi'' &\leftarrow \alpha_{u1:u2, u2:u1}(EstAmi') \cup EstAmi' \\ FluxDeX &\leftarrow \pi_{flux}(\sigma_{utilisateur='X'}(Suit)) \\ FluxdAmi &\leftarrow \pi_{flux}(EstAmi'' *_{u1=utilisateur} Suit) \\ FluxPartage &\leftarrow \pi_{flux}(\sigma_{utilisateur='X'}(Partage) * Compose) \\ Reponse &\leftarrow (FluxDeX - FluxdAmi) - FluxPartage \end{aligned}$$

Calcul tuple

$$\begin{aligned} Reponse &\leftarrow \{s.flux \mid Suit(s) \wedge s.utilisateur = 'X' \wedge \\ &\quad \nexists s1, a (Suit(s1) \wedge EstAmi(a) \wedge \\ &\quad ((s1.utilisateur = a.utilisateur1 \wedge a.utilisateur2 = 'X') \vee \\ &\quad (s1.utilisateur = a.utilisateur2 \wedge a.utilisateur1 = 'X')) \wedge \\ &\quad a.date \neq null \wedge s1.flux = s.flux) \wedge \\ &\quad \nexists p, c (Compose(c) \wedge Partage(p) \\ &\quad \wedge c.publication = p.publication \wedge p.utilisateur = 'X' \\ &\quad \wedge c.flux = s.flux)\} \end{aligned}$$

SQL

```
SELECT s.flux
FROM Suit s
WHERE s.utilisateur = 'X'
AND NOT EXISTS (SELECT *
                 FROM EstAmi a, Suit s1
                 WHERE (a.utilisateur1 = 'X' OR
```

```

        a.utilisateur2 = 'X')
    AND a.date != NULL
    AND s1.flux = s.flux
    AND s1.utilisateur != 'X'
    AND (s1.utilisateur = a.utilisateur1 OR
        s1.utilisateur = a.utilisateur2)
    )
AND NOT EXISTS (SELECT *
    FROM Partage p, Compose c
    WHERE p.publication = c.publication
        AND p.utilisateur = 'X'
        AND s.flux = c.flux
    )

```

Requête 4

Suggestion d'ami : La liste des utilisateurs qui ont partagé au moins 3 publications que X a partagé.

Algèbre

$$\begin{aligned}
 \text{PubliX} &\leftarrow \pi_{\text{publication}}(\sigma_{\text{utilisateur} \neq 'X'}(\text{Partage})) \\
 3\text{PartagesCommuns} &\leftarrow \alpha_{\text{publication:publi1}}(\text{Partage} * \text{PubliX}) * \\
 &\quad \alpha_{\text{publication:publi2}}(\text{Partage} * \text{PubliX}) * \\
 &\quad \alpha_{\text{publication:publi3}}(\text{Partage} * \text{PubliX}) \\
 3\text{PartagesCommunsDif} &\leftarrow \sigma_{\text{publi1} \neq \text{publi2} \wedge \text{publi1} \neq \text{publi3} \wedge \text{publi3} \neq \text{publi2}}(3\text{PartagesCommuns}) \\
 \text{Reponse} &\leftarrow \pi_{\text{utilisateur}}(3\text{PartagesCommunsDif})
 \end{aligned}$$

Calcul tuple

$$\{u.\text{email} \mid \text{Utilisateur}(u) \wedge \exists p1, p2, p3, pX1, pX2, pX3(
 \text{Partage}(p1) \wedge \text{Partage}(p2) \wedge \text{Partage}(p3) \wedge \text{Partage}(pX1) \wedge
 \text{Partage}(pX2) \wedge \text{Partage}(pX3) \wedge p1.\text{utilisateur} = u.\text{email} \wedge$$

$$\begin{aligned}
& p2.utilisateur = u.email \wedge p3.utilisateur = u.email \wedge \\
& pX1.utilisateur = 'X' \wedge pX2.utilisateur = 'X' \wedge pX3.utilisateur = 'X' \wedge \\
& p1.publication = pX1.publication \wedge p2.publication = pX2.publication \wedge \\
& p3.publication = pX3.publication \}
\end{aligned}$$

SQL

```

SELECT u.email
FROM Utilisateur u
WHERE u.email != 'X'
      AND (SELECT COUNT(*)
           FROM Partage p1, Partage p2
           WHERE p1.utilisateur = 'X'
                AND p2.utilisateur = u.email
                AND p1.publication = p2.publication
           ) >= 3

```

Requête 5

Statistiques des souscriptions : La liste des flux auquel un utilisateur est inscrit avec le nombre de publications lues, le nombre de publications partagées, le pourcentage de ces dernières par rapport aux premières, cela pour les 30 derniers jours et ordonnée par le nombre de publications partagées.

SQL

Les flux pour lesquels X n'a ni lu ni partagé une publication sont volontairement oubliés.

```

SELECT s.flux, COUNT(l.publication), COUNT(p.publication),
       (COUNT(p.publication)/COUNT(l.publication))*100
FROM Suit s, Lit l, Partage p, Compose c
WHERE s.utilisateur = 'X'
      AND p.utilisateur = 'X'
      AND l.utilisateur = 'X'
      AND s.flux = c.flux
      AND c.publication = l.publication
      AND c.publication = p.publication

```

```

    AND INTERVAL(DAY, l.date, NOW()) < 30
GROUP BY s.flux
ORDER BY COUNT(p.publication)

```

Requête 6

Statistiques des amis : La liste des amis d'un utilisateur avec pour chacun le nombre de publications lues par jour et le nombre d'amis, ordonnée par la moyenne des lectures par jour depuis la date d'inscription de cet ami.

SQL

```

SELECT u.email, COUNT(l.publication)/INTERVAL(DAY,u.date,NOW()),
       COUNT(SELECT *
              FROM EstAmi a2
              WHERE a2.date != NULL
                    AND (a2.utilisateur1 = u.email OR
                        a2.utilisateur2 = u.email)
              )
FROM   Utilisateur u, EstAmi a, Utilisateur x, Lit l
WHERE  x.email = 'X'
       AND u.email != 'X'
       AND (u.email = a.utilisateur1 OR
            u.email = a.utilisateur2)
       AND a.date != NULL
       AND (x.email = a.utilisateur1 OR
            x.email = a.utilisateur2)
       AND u.email = l.utilisateur
GROUP BY u.email
ORDER BY COUNT(l.publication)/INTERVAL(DAY,u.date,NOW())

```