

OUAHEB Fawzi
MIMOUNI NADIR

1/

Description: la premiere requete nous permet de recuperer le nombre de client qui paye leurs abonnement en cb(carte bancaire).

requete en sql:

```
select type_de_paiement,COUNT(id_client)
from client
where type_de_paiement='cb' group by type_de_paiement;
```

requete en forme algebrique:

```
 $\pi$  type_de_paiement, COUNT (id_client)
 $\sigma$  (type_de_paiement = "cb" (client))
```

2/

Description : la deuxieme requete nous permet de recupere l'emplacement de tout les arrêts qui sont dans la ligne une(L1) et la ligne deux(L3).

requete sql:

```
select emplacement
from arret,contient
where arret.id_arret=contient.id_arret and id_ligne='L1'
union
select emplacement
from arret,contient
where arret.id_arret=contient.id_arret and id_ligne='L3';
```

requete en forme algebrique:

```
 $\pi$  emplacement
 $\sigma$  arret . id_arret = contient . id_arret AND id_ligne = "L1" (arret  $\bowtie$ 
contient) U
 $\pi$  emplacement
 $\sigma$  arret . id_arret = contient . id_arret AND id_ligne = "L3" (arret  $\bowtie$ 
contient)
```

3/

Description: la troisieme requete nous permet de recuperer les terminus de toutes les lignes qui ont l'arrêt "mairie".

requete en sql:

```
select terminus
from arret,contient,ligne
where ligne.id_ligne=contient.id_ligne and
arret.id_arret=contient.id_arret and emplacement='mairie';
```

requete en forme algebrique:

```
 $\pi$  terminus(
 $\sigma$  emplacement = "mairie" (arret  $\bowtie$  contient  $\bowtie$  ligne))
```

4/

Description: cette requete est une division qui va nous permettre de recuperer les arrêts qui sont dans toutes les lignes.

premiere

requete en forme algebrique:

C: la table contient

L: la table ligne

$C/L = \pi(\text{id_arret}(\text{contient})) - \pi(\text{id_arret})((\pi(\text{id_arret})(\text{contient}) * L) - C)$

$\pi \text{ emplacement } (\text{arret jointure } \pi(\text{id_arret}(\text{contient})) -$
 $\pi(\text{id_arret}(\pi \text{ id_arret, id_ligne}$
 $\quad (\pi(\text{id_arret}(\text{contient})) * (\pi(\text{id_ligne}(\text{ligne})) -$
 $\quad \pi(\text{id_ligne, id_arret}(\text{contient}))$

)

requetes sql:

methode 1; double negation:

select distinct emplacement

from contient c1, arret

where arret.id_arret=c1.id_arret and not exists

(select id_ligne from ligne where

not exists (select id_arret from

contient c2 where c1.id_arret=c2.id_arret and

c2.id_ligne=ligne.id_ligne));

methode 2: denombrement:

select emplacement

from arret

where id_arret=(select id_arret

from ligne, contient c2

where c2.id_ligne=ligne.id_ligne

group by c2.id_arret

having(count(c2.id_ligne)=(select count(id_ligne)

from ligne));

methode 3: operations ensemblistes

select emplacement

from arret

where id_arret=(select distinct id_arret from contient c2 where not exists (

(select id_ligne from ligne)

minus

(select contient.id_ligne from contient where

c2.id_arret=contient.id_arret)));

5/

Description: cette requetes est une division elle va nous permettre de recupereer le nom et le prenom du techniciens qui repare tout vehicule;
requete en algebrique:

R:la table reparer

V: la table vehicule

$R/V = \pi(\text{id_techniciens}(\text{reparer})) - \pi(\text{id_techniciens})((\pi(\text{id_techniciens})(\text{reparer}) * V) - R)$

$$\pi \text{ nom, prenom}(\text{personne} \bowtie \pi(\text{id_techniciens}(\text{reparer})) - \pi(\text{id_techniciens}(\pi \text{ id_techniciens, id_vec}(\pi(\text{id_techniciens}(\text{reparer})) * (\pi(\text{id_vec}(\text{vehicule})) - \pi(\text{id_vec, id_techniciens}(\text{reparer}))))))$$

)

requetes sql:

methode 1: double negation

select nom, prenom

from personne, techniciens

where personne.id_personne=techniciens.id_personne and

techniciens.id_techniciens=(

select distinct id_techniciens from reparer c1 where not exists (select id_vec from vehicule

where not exists

(select id_techniciens from

reparer c2 where c1.id_techniciens=c2.id_techniciens and

c2.id_vec=vehicule.id_vec));

methode 2:denombrement

select nom, prenom

from personne, techniciens

where personne.id_personne=techniciens.id_personne and

techniciens.id_techniciens=(

select id_techniciens

from reparer , vehicule

where reparer.id_vec=vehicule.id_vec

group by reparer.id_techniciens having count(reparer.id_vec)=(select count(id_vec) from vehicule));

methode 3: operations ensemblistes

select nom, prenom

from personne, techniciens

where personne.id_personne=techniciens.id_personne and

techniciens.id_techniciens=

(select distinct id_techniciens from reparer c2 where

not exists (

(select id_vec from vehicule)

minus

(select reparer.id_vec from reparer where

c2.id_techniciens=reparer.id_techniciens));

6/

Description: cette requete elle est composé d'une intersection de deux table qui va nous permerre d'afficher le nom et le prenom des

personne qui sont techniciens et clients en meme temps.

requete en algebrique:

$$\pi \text{ nom, prenom}(\text{personne} \bowtie \begin{matrix} \pi(\text{id_personne}(\text{techniciens})) \\ \cap \\ \pi(\text{id_personne}(\text{client})) \end{matrix})$$

requete en sql:

```
select nom, prenom
from personne
where id_personne=(select id_personne from techniciens
INTERSECT
select id_personne from client);
7/
```

Description: cette requete elle est composé d'une intersection de deux table qui va nous permettre d'afficher le nom et le prenom des personne qui sont techniciens et chauffeur en meme temps.

requete en algebrique:

$$\pi \text{ nom, prenom}(\text{personne} \bowtie \begin{matrix} \pi(\text{id_personne}(\text{techniciens})) \\ \cap \\ \pi(\text{id_personne}(\text{chauffeur})) \end{matrix})$$

requete en sql:

```
select nom, prenom
from personne
where id_personne=(select id_personne from techniciens
INTERSECT
select id_personne from chauffeur);
```