

Rapports et requêtes 3

A la suite des formations Rapports et requêtes 1 et 2, traitons d'autres exemples:

Accueil, ETP, observance et communauté.

La version Santia 6/5/2010 ou ultérieure est requise pour les travaux pratiques

formation 5/7/2010

Rapport Accueil 1/3

Je voudrais connaître le nombre de patients passés à l'accueil, en fonction du motif (consultation médicale, etc.).

Ok. D'abord la partie formulaire pour la période (voir formation 2):

```
#### saisie ####
Début de la période|debut_période|date|
Fin de la période|fin_période|date|
```

Ensuite, cherchons la table qui stocke les données de l'accueil.

La documentation indique que c'est la table **Entr**.

Oui et comme mentionné dans la doc, les colonnes qui nous intéressent dans la table sont:

ArriHoro: date et heure d'arrivée du patient

Moti: motif de son arrivée

Pour obtenir le nombre de patients arrivés pour consultation médicale:

```
#### sql ####
nb_accueil_med<-SELECT COUNT(*) FROM Entr WHERE Moti="Consultation méd." AND
ArriHoro BETWEEN debut_période AND fin_période
#### affichage ####
Période|debut_période|fin_période
Nb de patients arrivés pour consultation médicale:
nb_accueil_med
```

On voit maintenant qu'on peut mettre une condition sur une désignation:**WHERE**

Moti="Consultation méd.". Auparavant, nous n'avions mis des conditions que sur des dates ou des nombres.

Bien sûr, il faut saisir exactement la même désignation que dans les écrans Santia pour que cela fonctionne.

! Copier/coller le texte de paramétrage et supprimer le saut de ligne éventuel dans la requête SQL.

Rapport Accueil 2/3

Il reste donc à faire la même chose pour les autres motifs (Consultation sociale, Dépistage, etc).

Pourrais-je avoir aussi une répartition par âge de ce nombre de patients ?

Oui, ça nous oblige à aller chercher l'information dans la table **Doss** des dossiers, car évidemment l'âge n'est pas stocké dans la table des arrivées.

On a vu à la fin de la formation 2 comment aller chercher simultanément une information dans une autre table. On voulait savoir si le paiement concerne un ARV ou un médicament courant. Et on est allé chercher l'information dans la table des produits. Ici, on va demander à SQL, pour chaque arrivée à l'accueil, de relever le n° de dossier et d'aller chercher l'âge du patient dans la table des dossiers. Voici pour les enfants de moins de 15 ans:

```
nb_accueil_med_15<-SELECT COUNT( DISTINCT Entr.Doss ) FROM Entr,Doss WHERE
Moti="Consultation méd." AND Entr.Doss=Doss.Nume AND Doss.RensAge_ BETWEEN 1
AND 15 AND ArriHoro BETWEEN debut_pperiode AND fin_pperiode
```

On a ajouté la table **Doss** après FROM et une condition sur l'âge compris entre 1 et 15.

Oui, car SQL doit lire aussi la table des dossiers.

L'âge est la colonne **RensAge_** de la table **Doss**.

Doss.RensAge_ veut dire "colonne RensAge_ de la table Doss".

Notons que la tranche d'âge suivante est "BETWEEN 16 AND 19" (par ex.) pour éviter de compter 2 fois les enfants de 15 ans.

On a ajouté également la condition **Entr.Doss=Doss.Nume** pour que SQL fasse le lien entre l'arrivée d'un patient et son dossier qui contient l'âge.

Entr.Doss: identifiant du dossier renseigné dans l'arrivée à l'accueil

Doss.Nume: identifiant de ce dossier dans la table des dossiers

On a vu à la fin de la formation 2 que chaque ligne d'une table a un identifiant (comme les numéros de ligne dans une feuille Excel). On utilise cet identifiant pour chercher une information relative dans une autre table.

On peut comparer cela à un marque-page dans un livre. Le patient 2436 est arrivé à l'accueil pour une consultation médicale, on va chercher dans les dossiers au marque-page 2436 quel est son âge.

Et que signifie **DISTINCT Entr.Doss** ?

DISTINCT est un mot SQL qui permet de ne compter que les valeurs distinctes d'une colonne, c'est-à-dire de ne pas compter les doublons. Ici, il s'applique à l'identifiant du dossier renseigné dans l'arrivée à l'accueil.

Rapport Accueil 3/3

Mais avec cette requête, on n'obtient que l'âge que de ceux qui ont un dossier. Comment compter la répartition par âge de ceux qui arrivent à l'accueil et n'ont pas de dossier ?

Bonne remarque.

Voici pour ceux qui n'ont pas de dossier:

```
nb_accueil_med_15_sans_doss<-SELECT COUNT(*) FROM Entr WHERE
Moti="Consultation méd." AND Age_ BETWEEN 1 AND 15 AND ArriHoro BETWEEN
debut_pperiode AND fin_pperiode
```

La table **Entr** contient une colonne **Age_** qui est utilisée uniquement pour les patients sans dossier.

5/10



Lors d'une requête sur une table A (ex. les arrivées), SQL peut chercher en plus une information relative dans une table B (ex. l'âge)

6/10

Rapport ETP et observance 1/2

Comment compter le nombre de séances d'éducation thérapeutiques effectuées sur une période ?

La table des séances ou consultations d'ETP est **ObseCons** d'après la [doc](#):

```
nb_cons_etp<-SELECT COUNT(*) FROM ObseCons WHERE Dat_ BETWEEN debut_période AND fin_période
```

En fait, ce qui m'intéresse plutôt est le nombre de patients ayant suivi une séance d'ETP. C'est différent si un patient fait plusieurs séances.

Oui, on va demander à SQL de compter seulement les séances où le dossier est distinct:

```
nb_doss_etp<-SELECT COUNT( DISTINCT Doss ) FROM ObseCons WHERE Dat_ BETWEEN debut_période AND fin_période
```

COUNT(*) compte tout

COUNT(DISTINCT Doss) ne compte qu'une seule fois la valeur de la colonne Doss. Une seule séance est comptée pour un patient qui a fait plusieurs séances. Donc on obtient bien le nombre de patients ayant suivi une séance d'ETP.

7/10

Rapport ETP et observance 2/2

J'ai vu que dans la consultation médicale, le médecin peut saisir des informations sur l'observance du patient. Le pharmacien peut aussi saisir ces informations lors d'une dispensation.

Oui, c'est un petit questionnaire d'observance. On peut compter le nombre de questionnaires saisis:

```
nb_obse<-SELECT COUNT(*) FROM MediCons WHERE Obse<>"" AND Dat_ BETWEEN debut_periode AND fin_periode
```

MediCons: l'observance est stockée dans la table des consultations médicales
Obse: c'est la colonne de la table où la conclusion du questionnaire est saisie (écran Dossier > Médical > Suivi > Observance > Conclusion)
Dat_ est la date de la consultation médicale

En fait, on compte le nombre de consultations médicales où une conclusion d'observance a été saisie, c'est-à-dire n'est pas vide: **Obse<>""**. "<>" signifie "n'est pas égal à".

Ok. Mais comme pour l'ETP, je voudrais plutôt le nombre de patients qui ont répondu à ce questionnaire d'observance et non pas le nombre de questionnaires.

Même réponse que précédemment. La table **MediCons** contient une colonne **Doss**, donc on peut demander à SQL de ne compter que les dossiers distincts:

```
nb_obse<-SELECT COUNT( DISTINCT Doss ) FROM MediCons WHERE Obse<>"" AND Dat_ BETWEEN debut_periode AND fin_periode
```

Si un patient a répondu plusieurs fois au questionnaire d'observance, il n'est compté qu'une fois. **DISTINCT** évite de compter les doublons de la colonne **Doss**.

8/10



*En SQL, le mot **DISTINCT** permet de ne pas compter les doublons contenus dans une colonne*

9/10

Rapport Communauté 1/2

Comment compter le nombre d'activités communautaires réalisées sur une période, en fonction de leur type ?

On lit dans la **doc** que la table des activités communautaires est **Comm**:

```
nb_comm<-SELECT COUNT(*) FROM Comm WHERE Dat_ BETWEEN debut_periode AND fin_periode
```

Pour un type d'activité seulement:

```
nb_comm_parole<-SELECT COUNT(*) FROM Comm WHERE Acti="Groupe de parole" AND Dat_ BETWEEN debut_periode AND fin_periode
```

Attention, la désignation de l'activité est tronquée à 20 caractères. Par exemple, pour la désignation "Education nutritionnelle":

```
nb_comm_nutri<-SELECT COUNT(*) FROM Comm WHERE Acti="Education nutritionn"  
AND Dat_ BETWEEN debut_période AND fin_période
```

Ok, la colonne **Acti** contient la désignation du type d'activité.

10/10

Rapport Communauté 2/2

Et pour le nombre de personnes qui ont participé à ces activités ?

La liste des participants à une activité communautaire est stockée dans la table **CommDoss** d'après la [doc](#).

D'un côté la date et le type de l'activité sont dans la table **Comm**, de l'autre les participants sont dans une autre table. C'est logique en terme de stockage pour ne pas répéter la date et le type d'activité pour chaque participant. Mais ça oblige SQL à chercher l'information des participants dans l'autre table:

```
nb_doss_parole<-SELECT COUNT( DISTINCT CommDoss.Doss ) FROM Comm,CommDoss  
WHERE Comm.Nume=CommDoss.Comm AND Acti="Groupe de parole" AND Dat_  
BETWEEN debut_période AND fin_période
```

Donc on demande à SQL de lire les tables **Comm** et **CommDoss**, à la suite du mot **FROM**.

Oui, et de ne pas compter les doublons de la colonne **Doss** de la table **CommDoss**. Cette colonne contient les identifiants des dossiers des participants. On ne compte qu'une fois les participants à plusieurs activités du même type.

Et **Comm.Nume=CommDoss.Comm** ?

C'est le lien entre la table des activités et celle des participants. Chaque participant est rattaché à une activité. Chaque activité a un identifiant (**Comm.Nume**). On demande à SQL de chercher cet identifiant dans la colonne activité de la table des participants (**CommDoss.Comm**). D'où le signe "=" entre les colonnes de ces deux tables, comme condition.

Voilà, c'est tout pour aujourd'hui,

Merci de votre attention !
