

Exemple d'applications

Une bibliothèque d'images

Nous allons construire une bibliothèque d'images qui permet à plusieurs utilisateurs de partager des images et de les annoter en associant des à chaque image, un ou plusieurs mot-clés. Le dialecte SQL utilisé dans cette exemple est celui de PostgreSQL (<http://www.postgresql.org>), pour des raisons de simplicité. Les types de données varient d'un SGBD à l'autre mais les concepts resteront les mêmes.

Modèle de données

Créons tout d'abord une représentation d'un utilisateur.

```
create table utilisateur
(
  sid serial primary key not null,
  prenom varchar(256) not null,
  nom varchar(256) not null,
  email varchar(256) not null,
  creation timestamp not null
);
```

sid est un identifiant dont la particularité est d'être unique et strictement monotone grâce au fait qu'il est basé sur une séquence qui possède les mêmes propriétés. Une séquence est une fonction strictement monotone qui renvoie une liste d'entiers distincts à chaque fois qu'elle est appelée. sid se prête donc particulièrement à une utilisation comme clé primaire. Utiliser des clés primaires arbitraires, c'est-à-dire créer un identifiant spécifiquement à cet usage, plutôt que d'utiliser des attributs uniques existants (comme dans ce cas, l'adresse email) facilite les jointures.

Pour s'assurer de l'unicité de l'adresse email dans cette table, ajoutons une contrainte d'unicité sur l'attribut email

```
alter table utilisateur add constraint email_unique unique(email);
```

Créons à présent la table qui comportera toutes les images. Comme nous construisons une bibliothèque qui pourra contenir un grand nombre d'images, nous stockerons ces images en dehors du SGBD et ne ferons figurer que le chemin qui mène au fichier qui contient l'image. Il est possible de stocker les images elles-mêmes dans la base mais cela conduira à limiter la capacité du SGBD, qui, sous une charge importante, passera un certain temps à extraire les images de ses tables et limitera le trafic maximum.

```
create table image
(
  sid serial primary key not null,
  nom varchar(256) not null,
  chemin varchar(256) not null,
  creation timestamp not null
);
```

Associions maintenant images et utilisateurs.

```
create table image_utilisateur
(
```

```

    sid serial primary key not null,
    image_sid integer not null,
    utilisateur_sid integer not null
);

```

Pour être sûr que image_sid et utilisateur_sid pointent vers des objets existants, ajoutons 2 contraintes d'intégrité référentielle.

```

alter table image_utilisateur add constraint image_utilisateur_utilisateur_fk
foreign key (utilisateur_sid) references utilisateur(sid);

alter table image_utilisateur add constraint image_utilisateur_image_fk
foreign key (image_sid) references image(sid);

```

Ajoutons des mots-clés :

```

create table mot_cle
(
    sid serial primary key not null,
    nom varchar(256) not null,
    creation timestamp not null
);

```

Et associons-les aux images :

```

create table image_mot_cle
(
    sid serial primary key not null,
    image_sid integer not null,
    mot_cle_sid integer not null
);

```

Même traitement pour éviter les mots-clés inexistantes attachés à des images inexistantes :

```

alter table image_mot_cle add constraint image_mot_cle_mot_cle_fk
foreign key (mot_cle_sid) references mot_cle(sid);

alter table image_mot_cle add constraint image_mot_cle_image_fk
foreign key (image_sid) references image(sid);

```

Dans cet exemple, nous omettons de créer des index puisque ceux-ci, quoique qu'indispensables pour obtenir une performance correcte lorsque la cardinalité des relations dépassent la dizaine ou la centaine, ne modifient en rien le résultat.

Exemples de requête

Nous avons besoin de données pour ces exemples.

```

insert into utilisateur (prenom, nom, email, creation) values ('Dupont', 'Dupond',
'dupont@dupond.fr', now());
insert into utilisateur (prenom, nom, email, creation) values ('Durant', 'Durand',
'durant@durand.fr', now());
insert into image (chemin, nom, creation) values ('/mes/images/', 'image1.jpg', now());
insert into image (chemin, nom, creation) values ('/mes/images/', 'image2.jpg', now());
insert into image (chemin, nom, creation) values ('/mes/images/', 'image3.jpg', now());
insert into image_utilisateur (image_sid, utilisateur_sid) values (1, 1);
insert into image_utilisateur (image_sid, utilisateur_sid) values (3, 1);

```

```

insert into image_utilisateur (image_sid, utilisateur_sid) values (3, 2);
insert into image_utilisateur (image_sid, utilisateur_sid) values (2, 2);
insert into mot_cle (nom, creation) values ('wiki', now());
insert into mot_cle (nom, creation) values ('populaire', now());
insert into image_mot_cle (image_sid, mot_cle_sid) values (3, 1);

```

Combien d'images appartiennent-elles à l'utilisateur dont l'email est "dupont@dupond.fr" ?

```

select count(*)
  from utilisateur u
 join image_utilisateur iu on (u.sid = iu.utilisateur_sid)
 where u.email = 'dupont@dupond.fr';

```

Quelles sont les 10 images les plus partagées ?

```

select i.chemin, i.nom, count(*) as partage
  from image i
 join image_utilisateur ui on (i.sid = ui.image_sid)
 group by i.chemin, i.nom
 order by count(*) desc
 limit 10;

```

Quel est l'utilisateur avec le plus d'images ?

```

select u.prenom, u.nom
  from utilisateur u
 join image_utilisateur iu on (u.sid = iu.utilisateur_sid)
 group by u.prenom, u.nom
 order by count(*) desc
 limit 1;

```

Quelles sont les images qui n'appartiennent à personne ?

```

select i.chemin, i.nom
  from image i
 left outer join image_utilisateur iu on (i.sid = iu.image_sid)
 where iu.utilisateur_sid is null;

```

Ajoutons le mot-clé "wiki" à toutes les images de "dupont@dupond.fr".

```

insert into image_mot_cle (image_sid, mot_cle_sid)
select iu.image_sid, mc2.sid
  from utilisateur u
 join image_utilisateur iu on (u.sid = iu.utilisateur_sid)
 left outer join image_mot_cle imc on (iu.image_sid = imc.image_sid)
 left outer join mot_cle mc on (imc.mot_cle_sid = mc.sid),
  mot_cle mc2
 where u.email = 'dupont@dupond.fr'
   and (imc.sid is null or mc.sid is null or mc.nom <> 'wiki')
   and mc2.nom = 'wiki'
 group by mc2.sid, iu.image_sid;

```

Dans cette dernière requête, nous voulons éviter les images qui sont déjà associées au mot-clé "wiki", tout en conservant les images sans mot-clé ou associées à d'autres mot-clés, le tout pour l'utilisateur en question.

Comme vous pouvez le voir, SQL permet d'exprimer de manière succincte des conditions relativement précises.

Récupérée de « https://fr.wikibooks.org/w/index.php?title=Programmation_SQL/Exemples&oldid=609671 »

La dernière modification de cette page a été faite le 14 décembre 2018 à 15:30.

Les textes sont disponibles sous [licence Creative Commons attribution partage à l'identique](#) ; d'autres termes peuvent s'appliquer.

Voyez les [termes d'utilisation](#) pour plus de détails.