

Cours de bases de données, ,
<http://sql.bdpedia.fr>

SQL, quantificateurs et nœud d'algation

SQL, quantificateurs et négation

Cette session présente les quantifications **existentielle** et **universelle**.

Les quantificateurs permettent l'expression de la négation : “je veux tous ces nuplets **sauf** ceux-là”.

Dans cette session :

- Le quantificateur `exists` et l'expression du “pour tout”
- La négation
- Equivalence des requêtes : plusieurs syntaxes, une seule signification.

Ces diapositives correspondent au support en ligne disponible sur le site <http://sql.bdpedia.fr/>

Le quantificateur “exists”

Reprenons la requête “les logements où l’on peut faire du ski”.

```
select distinct l.nom
from Logement as l, Activité as a
where l.code = a.codeLogement
and a.codeActivité = 'Ski'
```

“a” n’intervient pas dans le nuplet-résultat. On peut la remplacer par une variable liée.

```
select distinct l.nom
from Logement as l
where exists (select ''
              from Activité as a
              where l.code = a.codeLogement
              and a.codeActivité = 'Ski')
```

Légère reformulation : maintenant on cherche les logements tels **qu’il existe** une activité “Ski”.

Construction de formules complexes

On peut construire des formules imbriquées (sous-requêtes SQL) sans limitation de profondeur.

Qui est allé dans les Alpes ?

```
select distinct v.prénom, v.nom
from Voyageur as v, Séjour as s, Logement as l
where v.idVoyageur=s.idVoyageur
and s.codeLogement = l.code
and l.lieu = 'Alpes'
```

Ni s ni l ne sont utilisées dans la construction du nuplet-résultat

Qui est allé dans les Alpes

Avec quantificateur existentiel : "Les voyageurs tels **qu'il existe** un de leurs séjours dans les Alpes".

```
select distinct v.prénom, v.nom
from Voyageur as v
where exists (select ''
              from Séjour as s, Logement as l
              where v.idVoyageur=s.idVoyageur
                 and s.codeLogement = l.code
                 and l.lieu = 'Alpes')
```

Plus clair ? Moins clair ?

Imbrication d'imbrication

Avec quantificateur existentiel : "Les voyageurs tels qu'il **existe** un de leurs séjours tel que son logement **existe** dans les Alpes".

```
select distinct v.prénom, v.nom
from Voyageur as v
where exists (select ''
              from Séjour as s
              where v. idVoyageur=s.idVoyageur
              and exists (select ''
                          from Logement as l
                          where s.codeLogement = l .code
                          and l.lieu = 'Alpes'))
              )
```

Pas très naturel.

Quantificateurs et négation

Les logements qui **ne proposent pas** de Ski.

```
select distinct l.nom
from Logement as l
where not exists (select ''
                  from Activité as a
                  where l.code = a.codeLogement
                  and a.codeActivité = 'Ski')
```

Correspond à la formulation : “Les logements tels **qu’il n’existe pas** d’activité Ski”.

Quantificateur universel

Les voyageurs qui sont allés dans **tous** les logements

```
select distinct v.prénom, v.nom
from Voyageur as v
where not exists (select ''
                  from Logement as l
                  where not exists (select ''
                                    from Séjour as s
                                    where l.code = s.codeLogement
                                    and   v.idVoyageur = s.idVoyageur)
                  )
```

Reformulation avec double négation : on cherche les voyageurs tels **qu'il n'existe pas** de logement où **ils ne sont pas** allés.

À retenir

SQL = un langage à la définition très précise.

- **Tout ce qui peut s'exprimer par une formule logique est exprimable en SQL.** Ni plus, ni moins.
- **Inversement**, tout ce qui ne s'exprime pas par une formule (boucles, incrémentations, etc.) ne s'exprime pas en SQL.

Maîtriser SQL = savoir exprimer sa requête de manière rigoureuse.

Vous savez maintenant exprimer **tout** ce qui est exprimable en SQL (sauf agrégations)