

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

Vues

# Vues

Toute requête produit une relation. Nommer cette requête c'est nommer la relation résultat et pouvoir l'interroger. **Une vue est une requête nommée.**

C'est aussi une table **calculée** au moment où on l'interroge.

Dans cette session :

- La commande `create view`
- Interrogation d'une vue
- Mise à jour d'une vue

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

# La commande create view

Une vue est le résultat d'une requête

```
create view Koudalou as
  select nom, adresse, count(*) as nb_apparts
  from Immeuble as i join Apart as a on (i.id=a.idImmeuble)
  where i.id=1
  group by i.id, nom, adresse
```

Le résultat de la requête est réévalué à chaque fois que l'on accède à la vue.

# Interrogation d'une vue

On interroge une vue comme n'importe quelle table.

```
select * from Koudalou
```

nom	adresse	nb_apparts
Koudalou	3 rue des Martyrs	5

Une vue peut répondre à des objectifs de **simplification** et/ou de **confidentialité**.

# Vue dénormalisée

Une vue peut présenter une base dénormalisée.

```
create view ApartKoudalou as
  select no, surface, niveau, i.nom as immeuble, adresse,
         concat(p.prénom, ' ', p.nom) as occupant
  from Immeuble as i, Apart as a, Personne as p
  where i.id=a.idImmeuble
  and   a.id=p.idApart
  and   i.id=1
```

no	surf.	niv.	immeuble	adresse	occupant
1	150	14	Koudalou	3 rue des Martyrs	Léonie Atchoum
51	200	2	Koudalou	3 rue des Martyrs	Barnabé Simplet
52	50	5	Koudalou	3 rue des Martyrs	Alice Grincheux
43	75	3	Koudalou	3 rue des Martyrs	Brandon Timide

# Insertion dans une vue

Beaucoup de restrictions.

- la vue doit être basée sur une seule table ;
- toute colonne non référencée dans la vue doit pouvoir être mise à `null` ou disposer d'une valeur par défaut ;
- on ne peut pas mettre à jour un attribut qui résulte d'un calcul ou d'une opération.

```
create view PropriétaireAlice
  as select * from Propriétaire
  where idPersonne=2
```

```
insert into PropriétaireAlice values (2, 100, 20)
```

```
insert into PropriétaireAlice values (3, 100, 20)
```

# La clause check option

Sur la vue précédente, la requête :

```
select * from PropriétaireAlice
```

ne montre pas le propriétaire 3 que l'on vient d'insérer !

La clause `check option` permet de n'insérer que des nuplets que l'on peut ensuite sélectionner.

```
create view PropriétaireAlice  
as select * from Propriétaire  
where idPersonne=2  
with check option
```

# À retenir

Les **vues** sont des requêtes nommées que l'on peut traiter comme des tables.

Elles permettent de restructurer “virtuellement” une base.

- Pour faciliter l'accès (jointures pré-définies)
- Pour restreindre la visibilité des données (on ne donne accès qu'à la vue)
- Les insertions dans les vues offrent peu d'intérêt.