

Bases de données

TD n° 3 : Requêtes SQL

Exercice 1 : modélisation (reprise TD2)

On veut construire le schéma entité/association pour une agence immobilière.

- Pour chaque bien, on doit connaître son adresse.
- On doit en connaître le propriétaire : nom, prénom, adresse mail, téléphone.
- Un bien est proposé soit en vente, soit en location, dans une ou plusieurs filiales de l'agence. Les prix peuvent être différents dans des agences différentes.
- Les clients visitent les biens et font éventuellement une offre d'achat ou déposent un dossier de location. On connaît la date de ces transactions et on note également la date de vente ou la date de début de contrat de location quand celle-ci est connue.
- Chaque filiale a un directeur dont on connaît le nom, le prénom et le numéro de portable. Un directeur peut s'occuper de plusieurs agences.
- On connaît également l'adresse et le numéro de téléphone de chaque filiale.

Donner un schéma entité/association qui modélise le fonctionnement de l'agence.

On considère maintenant les tables suivantes concernant l'appréciation de spectateurs sur des films :

- Film(numFilm, titre)
- Client(Login, trancheAge, sexe)
- Age(Tranche, min, max)
- Note(numFilm, Login, Note)

Film		Client		
Num Film	Titre	Login	Tranche Age	Sexe
1	Les enfants loup	u1	1	H
2	Kirikou	u2	2	
3	Princesse Mononoke	u3	4	H
4	Mon voisin Totoro	u4		F
		u5	2	

Age			Note		
Tranche	Min	Max	Num Film	Login	Note
1		8			
2	9	16	2	u1	3
3	17	24	2	u4	4
4	25	44	3	u1	5
5	45	64	4	u2	4
6	65		2	u2	3

Exercice 2 : jointures

Donner le résultat des requêtes suivantes.

```
SELECT Titre, Login, Note
  FROM Film, Note
 WHERE Film.numFilm = Note.numFilm;
```

```
SELECT Titre, Login, Note
  FROM Film JOIN Note ON Film.numFilm = Note.numFilm;
```

```
SELECT Titre, Login, Note
  FROM Film JOIN Note;
```

```
SELECT Login, TrancheAge, Age.Tranche, Min, Max
  FROM Client, Age
 WHERE Client.TrancheAge = Age.Tranche;
```

```
SELECT Login, TrancheAge, Age.Tranche, Min, Max
  FROM Client JOIN Age ON Client.TrancheAge = Age.Tranche;
```

Exercice 3 : requêtes

Donner des requêtes pour les questions suivantes.

1. Numéro des films qui ont une note égale à 3.
2. Titre de ces films.
3. Login des clients qui ont certainement moins de 40 ans.
4. Notes des films dont le titre commence par 'K'.

Exercice 4 : ORDER BY, GROUP BY

1. Donner une requête qui présente l'ensemble des clients par ordre d'âge croissant.
2. Donner une requête qui présente l'ensemble des clients par ordre d'âge décroissant.
3. Donner le résultat des requêtes suivantes.

```
SELECT Note, COUNT(*)
  FROM Note
 GROUP BY Note;
```

```
SELECT Note, COUNT(DISTINCT NumFilm)
  FROM Note
 GROUP BY Note;
```

```
SELECT Note, COUNT(DISTINCT NumFilm)
  FROM Note
 GROUP BY Note
 HAVING COUNT(DISTINCT NumFilm) = 2;
```

- Donner le résultat de la requête suivante.

```
SELECT Min, Max, COUNT(TrancheAge) as Nombre
  FROM Client, Age
 WHERE Client.TrancheAge = Age.Tranche
 GROUP BY Tranche, Min, Max;
```

Combien y a-t-il de groupes ? Pourquoi inclure les attributs `Min`, `Max` dans le `GROUP BY` ?

- Modifier la requête pour qu'elle affiche les tranches par ordre croissant.
- Modifier la requête pour qu'elle inclue aussi les tranches d'âge non représentées dans la clientèle.
- Écrire une requête qui retourne la note moyenne du film 2.
- Écrire une requête qui retourne la liste de tous les films avec leur note moyenne. Donner le résultat de votre requête.
- Dans la requête précédente, les films non notés apparaissent-ils parmi les résultats ? Expliquer pourquoi.
- Donner une requête qui trouve la liste des utilisateurs avec le nombre de films que chacun a noté.
- Donner une requête qui trouve la liste des utilisateurs ayant noté au moins 2 films.

Exercice 5 : sous-requêtes (pour les M1)

- Expliquer les requêtes suivantes.

```
SELECT Titre
  FROM Film
 WHERE NumFilm IN (1,3,4);
```

```
SELECT Titre
  FROM Film
 WHERE NumFilm IN (
    SELECT numfilm
      FROM note
    );
```

- Donner une requête qui liste tous les films n'ayant aucune note.
- Donner une requête listant, pour chaque film, toutes les notes au-dessus de la note moyenne de ce film.
- Donner une requête qui donne la liste des films ayant un écart d'au moins deux entre deux de ses notes.