

Exploitation d'une base de données

Département Informatique

IUT2 de Grenoble

BUT1 - Ressource 2.06

Introduction à l'exploration et l'administration des BD

1. Révisions : BD relationnelle et SQL simple
2. SQL pour explorer et analyser
3. SQL augmenté : *triggers*
4. Administration de 1^{er} niveau

Introduction à l'exploration et l'administration des BD

1. Révisions : BD relationnelle et SQL simple
2. SQL pour explorer et analyser
3. SQL augmenté : *triggers*
4. Administration de 1^{er} niveau

1. Révisions : BD relationnelle et SQL simple

1.1. Base de données relationnelle

1.2. Répartir l'information entre tables (dépendances fonctionnelles)

1.3. La jointure pour réunir les informations

1. Révisions : BD relationnelle et SQL simple

1.1. Base de données relationnelle

1.2. Répartir l'information entre tables (dépendances fonctionnelles)

1.3. La jointure pour réunir les informations

1.1. Base de données relationnelle

Donnée = 1 tuple

valeur pour un ensemble d'attributs défini

Attribut : définit une information élémentaire

un nom + un domaine (type et/ou ensemble de valeurs)

Schéma d'une relation : ensemble d'attributs *en relation*

Relation : ensemble de données (tuples)

= un tableau (**ligne** = donnée) (**colonne** = valeurs pour un attribut)

1.1. Base de données relationnelle

Avantages

- la structure de données est la même quelle que soit l'application
- une description logique indépendante de l'implantation physique
- le langage SQL permet une interrogation performante
- bien adaptée pour assurer une cohérence statique et dynamique

1.1. Base de données relationnelle

Exemple : le *Select Club*

prenom	nom	annee_inscr	num_salon	localisation
Sherlock	Holmes	1990	1	N
Sherlock	Holmes	1990	5	W
Hercule	Poirot	2004	3	S
Harry	Potter	2020	null	null
null	null	null	2	E
null	null	null	4	S
null	null	null	6	W

Il s'agit d'une base pour gérer les membres du *Select Club* dont certains peuvent avoir le privilège d'avoir des salons privés qui leur sont réservés. Chaque salon a une localisation géographique dans le club. Un salon privé ne peut être réservé qu'à un seul membre privilégié.

1. Révisions : BD relationnelle et SQL simple

1.1. Base de données relationnelle

1.2. Répartir l'information entre tables (dépendances fonctionnelles)

1.3. La jointure pour réunir les informations

1.2. Répartir l'information entre tables (dép. fonct.)

Pourquoi ne pas utiliser qu'une seule table ?

- il existe des dépendances fonctionnelles entre les attributs ;
- des attributs qualifient naturellement un même *objet* (entité) ;
- cela permet de minimiser la redondance d'information :
 - gain de place physique
 - facilite la maintenance de la cohérence des informations

Conséquence théorique de la définition d'une clé primaire :

une relation est un **ensemble** de tuples

1.2. Répartir l'information entre tables (dép. fonct.)

prenom	nom	annee_inscr	num_salon	localisation
Sherlock	Holmes	1990	1	N
Sherlock	Holmes	1990	5	W
Hercule	Poirot	2004	3	S
Harry	Potter	2020	null	null
null	null	null	2	E

...

- 1 Quels sont les *objets* (entités) sous-jacents ?
- 2 Quelles sont les dépendances fonctionnelles ?
- 3 Quels changements sur la 1ère ligne peuvent poser problème ?
- 4 Quel changement pour remplacer `prénom` et `nom` par leur initiale ?

1.2. Répartir l'information entre tables (dép. fonct.)

Exemple : le *Select Club*

idm	prenom	nom	annee_inscr	num_salon	loc.
1	S	H	1990	1	N
1	S	H	1990	5	W
2	H	P	2004	3	S
3	H	P	2020	null	null
null	null	null	null	2	E
null	null	null	null	4	S
null	null	null	null	6	W

⑤ Répartir l'information en deux tables.

1. Révisions : BD relationnelle et SQL simple

1.1. Base de données relationnelle

1.2. Répartir l'information entre tables (dépendances fonctionnelles)

1.3. La jointure pour réunir les informations

1.3. La jointure pour réunir les informations

Contrainte référentielle

un attribut fait référence à la clé primaire d'une autre relation

Contraintes de clé primaire + référentielles

modéliser la base par un schéma entité-association

Exemple : le *Select Club*

- 6 Ecrire le schéma relationnel de cette base.

1.3. La jointure pour réunir les informations

Pour reconstituer l'information globale de façon cohérente

- **jointure interne :**

- produit cartésien ...
- ...suivi d'une sélection sur l'égalité entre valeurs d'attributs liés par une contrainte référentielle.

- **! jointure externe :**

- jointure interne ...
- ...suivie de l'ajout de tuples complétés par des `NULL` pour que tous les tuples de chacune des deux tables soient présents.

1.3. La jointure pour réunir les informations

Le *Select Club* - Produit cartésien

Membre

Salon

idm	prenom	nom	an._inscr	num_sal.	loc.	idm
1	S	H	1990	1	N	1
2	H	P	2004	1	N	1
3	H	P	2020	1	N	1
1	S	H	1990	2	E	null
2	H	P	2004	2	E	null
3	H	P	2020	2	E	null
1	S	H	1990	3	S	2
2	H	P	2004	3	S	2
3	H	P	2020	3	S	2
1	S	H	1990	4	S	null
2	H	P	2004	4	S	null
3	H	P	2020	4	S	null
1	S	H	1990	5	W	1

1.3. La jointure pour réunir les informations

Le *Select Club* - Produit cartésien

Membre

Salon

idm	prenom	nom	an._inscr	num_sal.	loc.	idm
1	S	H	1990	1	N	1
2	H	P	2004	1	N	1
3	H	P	2020	1	N	1
1	S	H	1990	2	E	null
2	H	P	2004	2	E	null
3	H	P	2020	2	E	null
1	S	H	1990	3	S	2
2	H	P	2004	3	S	2
3	H	P	2020	3	S	2
1	S	H	1990	4	S	null
2	H	P	2004	4	S	null
3	H	P	2020	4	S	null
1	S	H	1990	5	W	1

1.3. La jointure pour réunir les informations

Le *Select Club* - Jointure interne sur `idm`

Membre

Salon

idm	prenom	nom	an._inscr	num_sal.	loc.	idm
1	S	H	1990	1	N	1
2	H	P	2004	3	S	2
1	S	H	1990	5	W	1

1.3. La jointure pour réunir les informations

Le *Select Club* - Jointure externe à gauche (Membre) sur `idm`

Membre

Salon

idm	prenom	nom	an._inscr	num_sal.	loc.	idm
1	S	H	1990	1	N	1
2	H	P	2004	3	S	2
1	S	H	1990	5	W	1
3	H	P	2020	null	null	null

1.3. La jointure pour réunir les informations

Le *Select Club* - Jointure externe à droite (Salon) sur *idm*

Membre

Salon

idm	prenom	nom	an._inscr	num_sal.	loc.	idm
1	S	H	1990	1	N	1
2	H	P	2004	3	S	2
1	S	H	1990	5	W	1
null	null	null	null	2	E	null
null	null	null	null	4	S	null
null	null	null	null	6	W	null

1.3. La jointure pour réunir les informations

Le *Select Club* - Jointure externe complète sur *idm*

Membre

Salon

idm	prenom	nom	an._inscr	num_sal.	loc.	idm
1	S	H	1990	1	N	1
2	H	P	2004	3	S	2
1	S	H	1990	5	W	1
3	H	P	2020	null	null	null
null	null	null	null	2	E	null
null	null	null	null	4	S	null
null	null	null	null	6	W	null

1.3. La jointure pour réunir les informations

SQL - Cas 1 : Le schéma résultat est l'union des attributs de T1 et T2

Produit cartésien

```
SELECT *  
FROM T1, T2;
```

Jointure interne

```
SELECT *  
FROM T1, T2  
WHERE T1.a = T2.b;
```

ou !

```
SELECT *  
FROM T1 JOIN T2 ON T1.a = T2.b;
```

1.3. La jointure pour réunir les informations

SQL - Cas 1 bis : jointures externes

Jointure externe complète (*pour les deux tables*)

```
SELECT *  
FROM T1 FULL JOIN T2 ON T1.a = T2.b;
```

Jointure externe (*complète pour la table*) à gauche

```
SELECT *  
FROM T1 LEFT JOIN T2 ON T1.a = T2.b;
```

Jointure externe (*complète pour la table*) à droite

```
SELECT *  
FROM T1 RIGHT JOIN T2 ON T1.a = T2.b;
```

1.3. La jointure pour réunir les informations

SQL - Cas 2 :

- jointure sur **tous les attributs de même nom** dans T1 et T2
- ces attributs ne sont **inclus qu'une seule fois** dans le schéma résultat

Jointure interne

```
SELECT *  
FROM T1 NATURAL JOIN T2;
```

Jointure externe complète

```
SELECT *  
FROM T1 NATURAL FULL JOIN T2;
```

Jointure externe (*complète pour la table*) à gauche ou à droite

```
SELECT *                               SELECT *  
FROM T1 NATURAL LEFT JOIN T2;          FROM T1 NATURAL RIGHT JOIN T2;
```


1.3. La jointure pour réunir les informations

Le *Select Club* - Jointure interne sur `idm`

Membre

Salon

idm	prenom	nom	an._inscr	num_sal.	loc.	idm
1	S	H	1990	1	N	1
2	H	P	2004	3	S	2
1	S	H	1990	5	W	1

1.3. La jointure pour réunir les informations

Le *Select Club* - Jointure interne naturelle

Membre

Salon

idm	prenom	nom	an._inscr	num_sal.	loc.
1	S	H	1990	1	N
2	H	P	2004	3	S
1	S	H	1990	5	W