Algèbre relationnelle

Algèbre relationnelle

Exemple d'instance de la Base de Données Cinéma :

FILM	*TITRE*	*METTEUR EN SCENE*	*ACTEUR*
	Speed 2	Jan de Bont	S. Bullock
	Speed 2	Jan de Bont	J. Patrick
	Speed 2	Jan de Bont	W. Dafoe
	Marion	M. Poirier	C. Tetard
	Marion	M. Poirier	MF Pisier
	Marion	M. Poirier	M. Poirier

CINE	*NOM*	*ADRESSE*	*TELEPHONE*
	Français	9, rue Montesquieu	05 56 44 11 87
	Gaumont	9, Cours Clemenceau	05 56 52 03 54
	Trianon	6, rue Franklin	05 56 44 35 17
	UGC Ariel	20, rue Judaique	05 56 44 31 17

PROGRAMME	*NOM-CINE*	*TITRE*	*HORAIRE*
	Français	Speed 2	18h00
	Français	Speed 2	20h00
	Français	Speed 2	22h00
	Français	Marion	16h00
	Trianon	Marion	18h00

Projection

- Notation : Projection(Relation / Attr)

Extraire tous les titres de tous les films :

R1 = Projection(FILM / TITRE)

FILM	*TITRE*	*METTEUR EN SCENE*	*ACTEUR*
	Speed 2	Jan de Bont	S. Bullock
	Speed 2	Jan de Bont	J. Patrick
	Speed 2	Jan de Bont	W. Dafoe
	Marion	M. Poirier	C. Tetard
	Marion	M. Poirier	MF Pisier
	Marion	M. Poirier	M. Poirier

R1	*TITRE*
	Speed 2
	Marion

Extraire les titres des films à l'affiche :

R2 = Projection(PROGRAMME / TITRE)

PROGRAMME	*NOM-CINE*	*TITRE*	*HORAIRE*
	Français	Speed 2	18h00
	Français	Speed 2	20h00
	Français	Speed 2	22h00
	Français	Marion	16h00
	Trianon	Marion	18h00

R2	*TITRE*
	Speed 2
	Marion

Sélection

- Notations :
  - Sélection(Relation / Attr *op Valeur*) où *op* est un opérateur de comparaison, *Valeur* une constante.
  - Sélection(Relation / Attr *op Attr'*) où *op* est un opérateur de comparaison, *Attr'* un autre attribut de la relation.

Extraire les informations concernant les films dont le titre est 'Speed 2' :

R3 = Sélection(FILM / TITRE = 'Speed 2')

FILM	*TITRE*	*METTEUR EN SCENE*	*ACTEUR*
	Speed 2	Jan de Bont	S. Bullock
	Speed 2	Jan de Bont	J. Patrick
	Speed 2	Jan de Bont	W. Dafoe
	Marion	M. Poirier	C. Tetard
	Marion	M. Poirier	MF Pisier
	Marion	M. Poirier	M. Poirier

R3	*TITRE*	*METTEUR EN SCENE*	*ACTEUR*
	Speed 2	Jan de Bont	S. Bullock
	Speed 2	Jan de Bont	J. Patrick
	Speed 2	Jan de Bont	W. Dafoe

Extraire les films dont un acteur est le metteur en scène de ce film :

R4 = Sélection(FILM / METTEUR EN SCENE = ACTEUR)

FILM	*TITRE*	*METTEUR EN SCENE*	*ACTEUR*
	Speed 2	Jan de Bont	S. Bullock
	Speed 2	Jan de Bont	J. Patrick
	Speed 2	Jan de Bont	W. Dafoe
	Marion	M. Poirier	C. Tetard
	Marion	M. Poirier	MF Pisier
	Marion	M. Poirier	M. Poirier

R4	*TITRE*	*METTEUR EN SCENE*	*ACTEUR*
	Marion	M. Poirier	M. Poirier

Sélection (2)

Extraire les informations concernant la programmation après 18h00 du film 'Marion' :

R5 = Sélection(PROGRAMME / HORAIRE >= 18h00 ET TITRE = 'Marion')

Ou

R5_1 = Sélection(PROGRAMME / HORAIRE >= 18h00)
R5 = Sélection(R5_1 / TITRE = 'Marion')

Ou

R5_1 = Sélection(PROGRAMME / HORAIRE >= 18h00)
R5_2 = Sélection(PROGRAMME / TITRE = 'Marion')
R5 = Intersection(R5_1, R5_2)

Sélection (3)

PROGRAMME	*NOM-CINE*	*TITRE*	*HORAIRE*
	Français	Speed 2	18h00
	Français	Speed 2	20h00
	Français	Speed 2	22h00
	Français	Marion	16h00
	Trianon	Marion	18h00

R5_1	*NOM-CINE*	*TITRE*	*HORAIRE*
	Français	Speed 2	18h00
	Français	Speed 2	20h00
	Français	Speed 2	22h00
	Trianon	Marion	18h00

R5	*NOM-CINE*	*TITRE*	*HORAIRE*
	Trianon	Marion	18h00

Sélection (4)

Extraire la programmation des films dont le titre est 'Marion' ou qui passent à l'UGC :

R6 = Sélection(PROGRAMME / TITRE = 'Marion' OU NOM-CINE = 'UGC')

Ou

R6_1 = Sélection(PROGRAMME / TITRE = 'Marion')
R6_2 = Sélection(PROGRAMME / NOM-CINE = 'UGC')
R6 = Union(R6_1, R6_2)

Sélection (5)

PROGRAMME	*NOM-CINE*	*TITRE*	*HORAIRE*
	Français	Speed 2	18h00
	Français	Speed 2	20h00
	Français	Speed 2	22h00
	Français	Marion	16h00
	Trianon	Marion	18h00

R6_1	*NOM-CINE*	*TITRE*	*HORAIRE*
	Français	Marion	16h00
	Trianon	Marion	18h00

R6_2	*NOM-CINE*	*TITRE*	*HORAIRE*

R6	*NOM-CINE*	*TITRE*	*HORAIRE*
	Français	Marion	16h00
	Trianon	Marion	18h00

Sélection (6)

Extraire les cinémas qui projettent le film 'Marion' après 18h00 :

R7_1 = Sélection(PROGRAMME / TITRE = 'Marion')
R7_2 = Sélection(R7_1 / HORAIRE >= 18h00)
R7 = Projection(R7_2 / NOM-CINE, HORAIRE)

PROGRAMME	*NOM-CINE*	*TITRE*	*HORAIRE*
	Français	Speed 2	18h00
	Français	Speed 2	20h00
	Français	Speed 2	22h00
	Français	Marion	16h00
	Trianon	Marion	18h00

R7_1	*NOM-CINE*	*TITRE*	*HORAIRE*
	Français	Marion	16h00
	Trianon	Marion	18h00

R7_2	*NOM-CINE*	*TITRE*	*HORAIRE*
	Trianon	Marion	18h00

R7	*NOM-CINE*	*HORAIRE*
	Trianon	18h00

Jointure

- Notation : Jointure(Relation1, Relation2 / Attr1 *op* Attr2) où *op* est un opérateur de comparaison, Attr1 est un attribut de Relation1 et Attr2 est un attribut de Relation2.

Exemple:

R_A	A1	A2
	*	a_1
	?	a_2
	+	a_3

R_B	B1	B2
	*	b_1
	-	b_2
	*	b_3
	+	b_4

Jointure(R_A, R_B / A1 = B1)	A1	A2	B1	B2
	*	a_1	*	b_1
	*	a_1	*	b_3
	+	a_3	+	b_4

Produit cartésien (R_A x R_B)	A1	A2	B1	B2
	*	a_1	*	b_1
	*	a_1	-	b_2
	*	a_1	*	b_3
	*	a_1	+	b_4
	?	a_2	*	b_1
	?	a_2	-	b_2
	?	a_2	*	b_3
	?	a_2	+	b_4
	+	a_3	*	b_1
	+	a_3	-	b_2
	+	a_3	*	b_3
	+	a_3	+	b_4

ATTENTION:

Jointure(R_A, R_B / A1 <> B1)	A1	A2	B1	B2
	*	a_1	-	b_2
	*	a_1	+	b_4
	?	a_2	*	b_1
	?	a_2	-	b_2
	?	a_2	*	b_3
	?	a_2	+	b_4
	+	a_3	*	b_1
	+	a_3	-	b_2
	+	a_3	*	b_3

Jointure(2)

- Avec
  - FILM(Titre, MetteurEnScene, Acteur)
  - PROGRAMME(NomCine,Titre, Horaire)
  - CINEMA(Nom, Adresse, Téléphone)
- Titres des films programmés à l'UGC :

  R1 = JOINTURE(CINEMA, PROGRAMME / Nom=NomCinema)
  R2 = SELECTION(R1/Nom='UGC')
  R3 = PROJECTION(R2 / Titre)

Opérateurs ensemblistes

- UNION : opère sur deux relations de même nature. Il permet d'obtenir une nouvelle relation qui contient tous les n-tuples des 2 relations initiales.
- Exemple : deux relations :
  - ACTIVITES_SPORTIVES(NumAdh, NomAdh,DateNaissance, DateAdhésion)
  - ACTIVITES_CULTURELLES(NumAdh, NomAdh, DateNaissance, DateAdhésion)
- Si on veut l'ensemble de tous ceux qui pratiquent au moins une activité :
  - ACTIVITES= UNION (ACTIVITES_SPORTIVES, ACTIVITES_CULTURELLES)

- INTERSECTION : opère sur deux relations de même nature. Il permet d'obtenir une nouvelle relation qui contient les n-tuples qui apparaissent à la fois dans les 2 relations initiales.

- Exemple : deux relations :
  - ACTIVITES_SPORTIVES(NumAdh, NomAdh,DateNaissance, DateAdhésion)
  - ACTIVITES_CULTURELLES(NumAdh, NomAdh, DateNaissance, DateAdhésion)

- Si on veut l'ensemble des adhérents qui pratiquent les deux types d'activités :

  - ACTIVITES_COMMUNES = INTERSECTION (ACTIVITES_SPORTIVES,ACTIVITES_CULTURELLES)

- DIFFERENCE : opère sur deux relations de même nature. Il permet d'obtenir une nouvelle relation qui contient les n-tuples qui apparaissent dans le première relation mais pas dans la deuxième.

- Exemple : deux relations :
  - ACTIVITES_SPORTIVES(NumAdh, NomAdh,DateNaissance, DateAdhésion)
  - ACTIVITES_CULTURELLES(NumAdh, NomAdh, DateNaissance, DateAdhésion)

- Si on veut l'ensemble des adhérents qui pratiquent des activités sportives mais pas les activités culturelles :

  - ACTIVITES_COMMUNES = DIFFERENCE (ACTIVITES_SPORTIVES ,ACTIVITES_CULTURELLES)

Modèle Relationnel
- Exercices -

- Avec
  - FILM(Titre, MetteurEnScene, Acteur)
  - PROGRAMME(NomCine,Titre, Horaire)
  - CINEMA(Nom, Adresse, Téléphone)

1. La liste de tous les metteurs en scène.
2. Les titres des films réalisés par Poirier.
3. Titres des films programmés à l'UGC.
4. Téléphones des cinémas qui programment un film avec Sandra Bullock

Structured Query Language (SQL)

- Langage de Définition de Données (LDD)
  création et modification de la structure des Bases de Données
- Langage de Manipulation de Données (LMD)
  insertion et modification des données des Bases de Données
- Langage de Contrôle des Données (LCD)
  gestion de la sécurité, confidentialité et Contraintes d'Intégrité
- Petit lexique entre le modèle relationnel et SQL :

Modèle Relationnel	SQL
Relation	Table
Attribut	Colonne
Tuple	Ligne

SQL : création/modification de la structure des Bases de Données

- La notion de domaine n'est pas très présente dans les SGBD. Il nous faut donc nous limiter à la définition des types syntaxiques suivants dans la majorité des cas :
  - VARCHAR2(n) Chaîne de caractères de longueur variable (maximum n)
  - CHAR(n) Chaîne de caractères de longueur fixe (n caractères)
  - NUMBERNombre entier (40 chiffres maximum)
  - NUMBER(n, m)Nombre de longueur totale n avec m décimales
  - DATE Date (DD-MON-YY est le format par défaut)
  - LONG Flot de caractères

SQL : Création de table

- **CREATE TABLE** <nom de la table> (<nom de colonne> <type> [**NOT NULL**][, <nom de colonne> <type>]…, [<contrainte>]…);
  où <contrainte> représente la liste des contraintes d'intégrité structurelles concernant les colonnes de la table crée. Elle s'exprime sous la forme suivante :
  - **CONSTRAINT** <nom de contrainte> <sorte de contrainte>
    où <sorte de contrainte> est :
    - **PRIMARY KEY** (attribut1, [attribut2…])
    - **FOREIGN KEY** (attribut1, [attribut2…]) **REFERENCES** <nom de table associée>(attribut1, [attribut2…])
    - **CHECK** (attribut <condition>) avec <condition> qui peut être une expression booléenne "simple" ou de la forme **IN** (liste de valeurs) ou **BETWEEN** <borne inférieure> **AND** <borne supérieure>

SQL : Modification de la structure d'une table

- Ajout :
  - **ALTER TABLE** <nom de la table> **ADD** (<nom de colonne> <type>
    [, <contrainte>]…);
  - **ALTER TABLE** <nom de la table> **ADD** (<contrainte> [, <contrainte>]…);
- Modification :
  - **ALTER TABLE** <nom de la table> **MODIFY**([<nom de colonne> <nouveau type>] [,<nom de colonne> <nouveau type>]…);

SQL : Manipulation de Données

- **Interrogation**

SELECT [**DISTINCT**] <nom de colonne>[, <nom de colonne>]…
FROM <nom de table>[, <nom de table>]…
[WHERE <condition>**]**
[GROUP BY <nom de colonne>[, <nom de colonne>]…
[HAVING <condition avec calcul verticaux>**]]**
[ORDER BY <nom de colonne>[, <nom de colonne>]…**]**

SQL : Manipulation de Données

- **Insertion de données**

INSERT INTO <nom de table> [(colonne, …)] **VALUES** (valeur, …)
INSERT INTO <nom de table> [(colonne, …)] **SELECT**…

- **Modification de données**

UPDATE <nom de table> **SET** colonne = valeur, …
[**WHERE** condition]
UPDATE <nom de table> **SET** colonne = **SELECT**…

- **Suppression de données**

DELETE FROM <nom de table> [**WHERE** condition]

La clause SELECT

- Base utilisée :
  - PILOTE(NUMPIL, NOMPIL, PRENOMPIL, ADRESSE, SALAIRE, PRIME)
  - AVION(NUMAV, NOMAV, CAPACITE, LOCALISATION)
  - VOL(NUMVOL, ***NUMPIL***, ***NUMAV***, DATE_VOL, HEURE_DEP, HEURE_ARR, VILLE_DEP, VILLE_ARR).
- On suppose qu'un vol, référencé par son numéro NUMVOL, est effectué par un unique pilote, de numéro NUMPIL, sur un avion identifié par son numéro NUMAV.

La clause SELECT : Expression des projections

- **SELECT** *
  FROM *table*
  où *table* est le nom de la relation à consulter. Le caractère * signifie qu'aucune projection n'est réalisée, i.e. que tous les attributs de la relation font partie du résultat.
  - Exemple : donner toutes les informations sur les pilotes.
    SELECT *
    FROM PILOTE
- La liste d'attributs sur laquelle la projection est effectuée doit être précisée après la clause **SELECT**.
  - Exemple : donner le nom et l'adresse des pilotes.
    SELECT NOMPIL, ADRESSE
    FROM PILOTE

La clause SELECT : Expression des sélections

- Clause WHERE
  - Exemple : donner le nom des pilotes qui habitent à Bordeaux.

```
SELECT   NOMPIL
FROM     PILOTE
WHERE        ADRESSE = 'BORDEAUX'
```

  - Exemple : donner le nom et l'adresse des pilotes qui gagnent plus de 3.000 €.

```
SELECT   NOMPIL
FROM     PILOTE
WHERE        SALAIRE > 3000
```


- Clause WHERE (Autres opérateurs spécifiques)
  - **IS NULL** : teste si la valeur d'une colonne est une valeur nulle (inconnue).
  - Exemple : rechercher le nom des pilotes dont l'adresse est inconnue.

```
SELECT NOMPIL
FROM   PILOTE
WHERE  ADRESSE IS  NULL
```

  - **IN** (liste) : teste si la valeur d'une colonne coïncide avec l'une des valeurs de la liste.
  - Exemple : rechercher les avions de nom A310, A320, A330 et A340.

```
SELECT  *
FROM    AVION
WHERE   NOMAV IN ('A310', 'A320', 'A330', 'A340')
```

Calculs horizontaux

- Des expressions arithmétiques peuvent être utilisées dans les clauses WHERE et SELECT.
  - Exemple : donner le revenu mensuel des pilotes Toulousain.

```
SELECT  NUMPIL, NOMPIL, SALAIRE + PRIME
FROM    PILOTE
WHERE   ADRESSE = 'TOULOUSE'
```

  - Exemple : quels sont les pilotes qui avec une augmentation de 10% de leur prime gagnent moins de 5.000 € ? Donner leur numéro, leurs revenus actuel et simulé.

```
SELECT  NUMPIL, SALAIRE+PRIME, SALAIRE + (PRIME*1.1)
FROM    PILOTE
WHERE   SALAIRE + (PRIME*1.1) < 5000
```


Calculs verticaux (fonctions agrégatives)

- Les fonctions agrégatives s'appliquent à un ensemble de valeurs d'un attribut de type numérique sauf pour la fonction de comptage **COUNT** pour laquelle le type de l'attribut argument est indifférent.

- Les fonctions agrégatives disponibles sont généralement les suivantes :

SUM	somme,
AVG	moyenne arithmétique,
COUNT	nombre ou cardinalité,
MAX	valeur maximale,
MIN	valeur minimale,
STDDEV	écart type,
VARIANCE	variance.

- Exemple : quel est le salaire moyen des pilotes Bordelais.

```
SELECT AVG(SALAIRE)
FROM   PILOTE
WHERE  ADRESSE = 'BORDEAUX'
```

- Exemple : trouver le nombre de vols au départ de Bordeaux.

```
SELECT COUNT(VOLNUM)
FROM   VOL
WHERE  VILLE_DEP = 'BORDEAUX'
```

- Dans cette requête, **COUNT**(*) peut être utiliser à la place de **COUNT**(VOLNUM)
- La colonne ou l'expression à laquelle est appliquée une fonction agrégative peut avoir des valeurs redondantes. Pour indiquer qu'il faut considérer seulement les valeurs distinctes, il faut faire précéder la colonne ou l'expression de **DISTINCT**.

- Exemple : combien de destinations sont desservies au départ de Toulouse ?

```
SELECT COUNT(DISTINCT VILLE_ARR)
FROM   VOL
WHERE  VILLE_DEP = 'TOULOUSE'
```


Tri des résultats

- Il est possible avec SQL d'ordonner les résultats. Cet ordre peut être croissant (**ASC**) ou décroissant (**DESC**) sur une ou plusieurs colonnes ou expressions.

ORDER BY *expression* [**ASC** | **DESC**],…

- L'argument de **ORDER BY** peut être un nom de colonne ou une expression basée sur une ou plusieurs colonnes mentionnées dans la clause **SELECT**.

  - Exemple : donner la liste des pilotes Bordelais par ordre de salaire décroissant, puis par ordre alphabétique des noms.

```
SELECT NOMPIL, SALAIRE
FROM   PILOTE
WHERE  ADRESSE = 'BORDEAUX'
ORDER BY SALAIRE DESC, NOMPIL
```

Groupement

- La clause GROUP BY permet de regrouper les enregistrement ayant la même valeur pour une ou plusieurs colonnes :
  - Exemple : quel est le nombre de vols effectués par chaque pilote ?

    SELECT NUMPIL, COUNT(NUMVOL)
    FROM VOL
    GROUP BY NUMPIL
    - Dans ce cas, le résultat de la requête comporte une ligne par numéro de pilote présent dans la relation VOL.
  - Exemple : combien de fois chaque pilote conduit-il chaque avion ?

    SELECT NUMPIL, NUMAV, COUNT(NUMVOL)
    FROM VOL
    GROUP BY NUMPIL, NUMAV
    - Nous obtenons ici autant de lignes par pilote qu'il y a d'avions distincts conduits le pilote considéré. Chaque classe d'équivalence créée par le **GROUP BY** regroupe tous les vols ayant même pilote et même avion.