

Algèbre Relationnelle et Introduction au Langage SQL

Sources du document :

- Livre bible Oracle 9i, J. Gabillaud, Editions Eni ;
- Support Oracle ;
- ISO Norme 2382:1999 parties 1 à 5, ISO Norme 9075 parties 1 à 14 corrigées en 2005.

Sources du document :

- Site sql.developpez.com ;
- Manuel SQL PostgreSQL (existe en plusieurs versions HTML, PDF...).



Olivier Mondet
<http://unidentified-one.net>

A. Les requêtes

Une requête est destinée à obtenir des informations précises et ordonnées sur la base de données. On peut également par le biais des requêtes :

- effectuer des calculs,
- obtenir des statistiques,
- modifier les tables,
- créer de nouvelles tables
- gérer des utilisateurs...

Trois façons de créer des requêtes :

- L'algèbre relationnelle qui permet de préparer une requête en utilisant un langage algébrique.
- Le QBE (Query By Exemple), qui est un mode graphique où l'on clique sur les éléments choisis pour construire la requête. C'est un moyen interactif et puissant utilisé par Access.
- Le SQL (Structured Query Language) qui est le langage des requêtes, qui est plus puissant que le QBE et plus universel.

B. L'algèbre relationnelle

B.1. Définition de l'algèbre relationnelle

Ensemble d'opérations permettant de manipuler des relations (schéma relationnel) et produisant comme résultat de nouvelles relations.

Basé sur le langage algébrique de Codd (Edgar Frank Codd décédé en 2003, père entre autre des « Forme Normales »).

B.2. Objectif de l'algèbre relationnelle

Description algorithmique ou symbolique préliminaire à l'utilisation d'un langage non procédural de type SQL.

B.3. Types d'opérations

Opérations spécifiques au modèle relationnel (Opérateurs relationnels) :

PROJECTION

SELECTION

JOINTURE

(RESTRICTION)

Les opérations spécifiques sont unitaires comme la Projection et la Sélection (une relation manipulée) ou binaire comme la Jointure.

Plusieurs types de Jointures existent : Equijointure (ou jointure naturelle), Jointure externe gauche ou droite (cela sera détaillé avec la clause JOIN dans le chapitre qui suit).

Opérations de la théorie des ensembles (Opérateurs ensemblistes) :

UNION

INTERSECTION

DIFFERENCE

PRODUIT

Degré d'une relation :

C'est le nombre de constituants (attributs) qui participent à cette relation.

Notion de t-uplet :

Pour une relation de degré t, chaque occurrence de cette relation est un t-uplet (ou tuple).

C. Opérateurs de base

C.1. La projection

La projection permet d'extraire des données d'une table, en ne conservant que les colonnes souhaitées.

Notation :

$R2 = \text{PROJECTION } R1(\text{Nom-Champ}, \text{Nom-Champ}...)$

R1 et R2 sont deux relations, entre parenthèses figurent les critères de projection.

Instructions SQL :

SELECT Nom-Table.Nom-Champ, Nom-Table.Nom-Champ...
FROM Nom-Table;

Exemple :

Soit la table FILM qui permet de gérer une vidéothèque.

FILM

no-film	titre	durée	production	code-catégorie
10	Camille Claudel	150	Gaumont	COMD
20	Fenêtre sur cour	120	Pathé	COMD
25	Sueurs froides	115	Pathé	COMD
50	Cendrillon	140	UGC	DESA
64	Super Mondet II	10	Universal	DOCU
65	La vie des coccinelles	60.	UGC	DOCU
71	La guerre des étoiles I	120	Paramounth	COMD
...

→ On veut faire une projection des colonnes *no-film* et *titre* de la table FILM :

En langage algébrique :

R1 = PROJECTION FILM(no-film, titre)

Traduction en SQL :

```
SELECT FILM.no-film, FILM.titre
FROM FILM;
```

RÉSULTAT REQUÊTE

no-film	titre
10	Camille Claudel
20	Fenêtre sur cour
25	Sueurs froides
50	Cendrillon
64	Super Mondet II
65	La vie des coccinelles
71	La guerre des étoiles I
...	...

C.2. La sélection

Permet d'extraire les lignes d'une table qui vérifient la réalisation d'une certaine condition (on parle parfois de critère).

Notation :

R2 = **SELECTION** R1(Expression conditionnelle)

R1 et R2 sont deux relations, entre parenthèses figure le critère de sélection.

Instructions SQL :

SELECT Nom-Table.Nom-Champ, Nom-Table.Nom-Champ...

FROM Nom-Table

WHERE Critère de sélection;

Opérateurs de comparaison : = != <> > >= < <=

Exemple :

Soit la table FILM qui permet de gérer une vidéothèque.

no-film	titre	durée	production	code-catégorie
10	Camille Claudel	150	Gaumont	COMD
20	Fenêtre sur cour	120	Pathé	COMD
25	Sueurs froides	115	Pathé	COMD
50	Cendrillon	140	UGC	DESA
64	Super Mondet II	10	Universal	DOCU
65	La vie des coccinelles	60.	UGC	DOCU
71	La guerre des étoiles I	120	Paramounth	COMD
...

→ On aimerait avoir les titres des films produits par Pathé.

En langage algébrique :

R1 = SELECTION FILM(production = "Pathé")

R2 = PROJECTION R1(titre)

Traduction en SQL :

```
SELECT FILM.titre
```

```
FROM FILM
```

```
WHERE FILM.production = "Pathé";
```

RÉSULTAT REQUÊTE

titre
Fenêtre sur cour
Sueurs froides

C.3. La jointure

La jointure consiste donc à combiner deux tables ligne à ligne en vérifiant la concordance entre certaines colonnes des deux tables. Autrement dit, cela permet de relier deux tables ayant un champ commun et de faire correspondre les lignes qui ont une même valeur.

Notation :

R3 = R1 \bowtie R2 (Expression conditionnelle)

Ou

R3 = JOINTURE R1.R2 (Expression conditionnelle)

R1 et R2 sont deux relations, entre parenthèses figure le critère de jointure.

Instructions SQL :

```
SELECT Nom-Table.Nom-Champ, Nom-table.Nom-Champ...
```

```
FROM Nom-Table, Nom-Table
```

```
WHERE Condition de jointure;
```

Opérateurs de comparaison :

= != <> > >= < <=

Exemple :

Soient les tables FILM et CATÉGORIE qui permettent de gérer une vidéothèque.

no-film	titre	durée	production	code-catégorie
10	Camille Claudel	150	Gaumont	COMD
20	Fenêtre sur cour	120	Pathé	COMD
25	Sueurs froides	115	Pathé	COMD
50	Cendrillon	140	UGC	DESA
64	Super Mondet II	10	Universal	DOCU
65	La vie des coccinelles	60.	UGC	DOCU
71	La guerre des étoiles I	120	Paramounth	COMD
...

code-catégorie	libellé-catégorie
COMD	comédie dramatique
DESA	dessin animé
DOCU	documentaire
...	...

→ On souhaiterait avoir le titre des films ainsi que les libellés de leurs catégories (et non leur code...).

En langage algébrique :

R1 = FILM \bowtie CATÉGORIE (FILM.code-catégorie = CATÉGORIE.code-catégorie)

R2 = PROJECTION R1(titre, libellé-catégorie)

Traduction en SQL :

```
SELECT FILM.titre, CATÉGORIE.libellé-catégorie  
FROM FILM, CATÉGORIE  
WHERE FILM.code-catégorie = CATÉGORIE.code-catégorie;
```

titre	libellé-catégorie
Camille Claudel	comédie dramatique
Fenêtre sur cour	comédie dramatique
Sueurs froides	comédie dramatique
Cendrillon	dessin animé
Super Mondet II	documentaire
La vie des coccinelles	documentaire
La guerre des étoiles I	comédie dramatique

Remarque n°1 :

Il est bien évidemment possible de faire plusieurs jointures, autant qu'il y a de tables à relier entre elles.

Equijointure : le pivot utilise l'opérateur = (jointure naturelle)

Thétajointure : le pivot utilise les opérateurs <, <=, >, >=, != ou <>

Remarque n°2 :

Il est possible d'utiliser des alias (pour simplifier) pour nommer une table différemment dans la requête.

Par exemple :

```
SELECT F.titre, C.libellé-catégorie  
FROM FILM F, CATÉGORIE C  
WHERE F.code-catégorie = C.code-catégorie;
```

C.4. Combinaison des opérateurs de base

Une requête d'extraction de données sur une base de données est la plupart du temps une combinaison astucieuse des trois opérations de base.

On peut donc selon les besoins combiner les instructions. L'opérateur AND permet d'avoir plusieurs conditions de sélection. L'instruction ORDER BY permet de trier selon l'ordre d'un des champs (ASCendant ou DESCendant).

Exemple de séquence d'opérations algébriques :

R3 = R1 \bowtie R2 (Condition de Jointure)
R4 = **SELECTION** R3 (Critère de sélection **ET** Critère de sélection)
R5 = **SELECTION** R4 (Critère de sélection)
R6 = **PROJECTION** R5 (Nom-Champ, Nom-Champ, ...)

Exemple de séquence d'instructions SQL :

```
SELECT Nom-Table.Nom-Champ, Nom-table.Nom-Champ...  
FROM Nom-Table, Nom-Table  
WHERE Condition de jointure  
AND Condition  
ORDER BY Nom-Table.Nom-Champ ASC;
```

Exemple :

Soient les tables FILM et CATÉGORIE qui permettent de gérer une vidéothèque.

no-film	titre	durée	production	code-catégorie
10	Camille Claudel	150	Gaumont	COMD
20	Fenêtre sur cour	120	Pathé	COMD
25	Sueurs froides	115	Pathé	COMD
50	Cendrillon	140	UGC	DESA
64	Super Mondet II	10	Universal	DOCU
65	La vie des coccinelles	60.	UGC	DOCU
71	La guerre des étoiles I	120	Paramounth	COMD
...

code-catégorie	libellé-catégorie
COMD	comédie dramatique
DESA	dessin animé
DOCU	documentaire
...	...

→ On souhaiterait avoir le numéro, le titre des films, les libellés des catégories auxquelles ils appartiennent uniquement pour les films de plus de 100 minutes. Le tout trié alphabétiquement (ordre ascendant) selon le titre du film.

En langage algébrique :

R1 = FILM \bowtie CATÉGORIE (FILM.code-catégorie = CATÉGORIE.code-catégorie)
R2 = **SELECTION** R1 (R1.Durée > 100)
R3 = **PROJECTION** R2 (no-film, titre, libellé-catégorie)

Traduction en SQL :

```
SELECT FILM.no-film, FILM.titre, CATÉGORIE.libellé-catégorie  
FROM FILM, CATÉGORIE  
WHERE FILM.code-catégorie = CATÉGORIE.code-catégorie;  
AND FILM.Durée > 100  
ORDER BY FILM.titre ASC;
```

RÉSULTAT REQUÊTE

no-film	titre	libellé-catégorie
10	Camille Claudel	comédie dramatique
50	Cendrillon	dessin animé
20	Fenêtre sur cour	comédie dramatique
71	La guerre des étoiles I	comédie dramatique
65	La vie des coccinelles	documentaire
25	Sueurs froides	comédie dramatique
64	Super Mondet II	documentaire

Si l'on avait demandé un classement dans l'ordre inverse (descendant) :

Traduction en SQL :

```
SELECT FILM.no-film, FILM.titre, CATÉGORIE.Libellé-catégorie
FROM FILM, CATÉGORIE
WHERE FILM.code-catégorie = CATÉGORIE.code-catégorie;
AND FILM.Durée > 100
ORDER BY FILM.titre DESC;
```

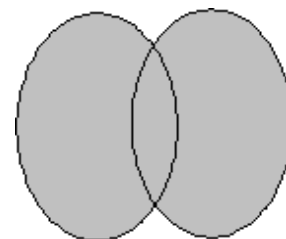
RÉSULTAT REQUÊTE

no-film	titre	libellé-catégorie
64	Super Mondet II	documentaire
25	Sueurs froides	comédie dramatique
65	La vie des coccinelles	documentaire
71	La guerre des étoiles I	comédie dramatique
20	Fenêtre sur cour	comédie dramatique
50	Cendrillon	dessin animé
10	Camille Claudel	comédie dramatique

D. Autres opérateurs (ensemblistes)

D.1. L'union (ou / or)

L'union consiste à combiner deux relations (compatibles) pour créer une troisième relation qui contient toutes les occurrences appartenant à l'une ou à l'autre des relations de départ.



Notation :

$R3 = R1 \cup R2$

Ou

$R3 = \mathbf{UNION} (R1, R2)$

Instructions SQL :

SELECT Nom-Champ, Nom-Champ, ...

FROM Nom-Table1

UNION

SELECT Nom-Champ, Nom-Champ, ...

FROM Nom-Table2;

On peut utiliser **UNION [ALL]** pour avoir toutes les lignes communes aux deux tables (y compris celles en double), sans cela les doublons sont éliminés.

Exemple :

Soient les tables ANGLAIS et ESPAGN qui permettent de gérer les candidats à des épreuves de langue.

nocandidat	nomcandidat	prénomcandidat	numétablissement
10123	DUPONT	Patrick	94010
10216	GRANGETTE	Didier	75100
10309	HERISSON	Patrick	94010
10405	MAGNAN	Hélène	75250
10505	SERRES	Sylvia	94010

nocandidat	nomcandidat	prénomcandidat	numétablissement
10025	ACQUINO	Patricia	91260
10216	GRANGETTE	Didier	75100
10505	SERRES	Sylvia	94010

→ On souhaiterait obtenir la liste des candidats passant les épreuves d'anglais ou d'espagnol (l'union des deux tables) (sans doublons).

En langage algébrique :

$$R1 = \text{ANGLAIS} \cup \text{ESPAGN}$$

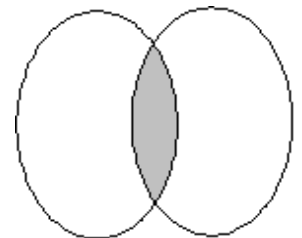
Traduction en SQL :

```
SELECT *  
FROM ANGLAIS  
UNION  
SELECT *  
FROM ESPAGN;
```

nocandidat	nomcandidat	prénomcandidat	numétablissement
10123	DUPONT	Patrick	94010
10216	GRANGETTE	Didier	75100
10309	HERISSON	Patrick	94010
10405	MAGNAN	Hélène	75250
10505	SERRES	Sylvia	94010
10025	ACQUINO	Patricia	91260

D.2. L'intersection (et / and)

L'intersection consiste à combiner deux relations (compatibles) pour créer une troisième relation qui contient toutes les occurrences appartenant à l'une et à l'autre des relations de départ.



Notation :

$$R3 = R1 \cap R2$$

Ou

$$R3 = \text{INTERSECTION} (R1, R2)$$

Instructions SQL :

```
SELECT Nom-Champ, Nom-Champ, ...  
FROM Nom-Table1  
INTERSECT  
SELECT Nom-Champ, Nom-Champ, ...  
FROM Nom-Table2;
```


Exemple :

Soient les tables ANGLAIS et ESPAGN qui permettent de gérer les candidats à des épreuves de langue.

ANGLAIS			
nocandidat	nomcandidat	prénomcandidat	numétablissement
10123	DUPONT	Patrick	94010
10216	GRANGETTE	Didier	75100
10309	HERISSON	Patrick	94010
10405	MAGNAN	Hélène	75250
10505	SERRES	Sylvia	94010

ESPAGN			
nocandidat	nomcandidat	prénomcandidat	numétablissement
10025	ACQUINO	Patricia	91260
10216	GRANGETTE	Didier	75100
10505	SERRES	Sylvia	94010

→ On souhaiterait obtenir la liste des candidats passant les épreuves d'anglais et d'espagnol (l'intersection des deux tables).

En langage algébrique :

$$R1 = \text{ANGLAIS} \cap \text{ESPAGN}$$

Traduction en SQL :

```

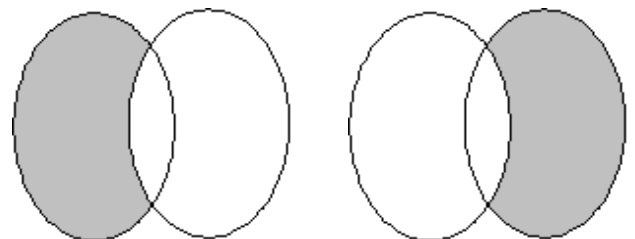
SELECT *
FROM ANGLAIS
INTERSECT
SELECT *
FROM ESPAGN;

```

RÉSULTAT REQUÊTE			
nocandidat	nomcandidat	prénomcandidat	numétablissement
10216	GRANGETTE	Didier	75100
10505	SERRES	Sylvia	94010

D.3. La différence (non / not)

La différence consiste à combiner deux relations (compatibles) pour créer une troisième relation qui contient toutes les occurrences appartenant à l'une des relations et non contenues dans l'autre des relations de départ. Deux différences sont possibles.



Notation :

$$R3 = R1 - R2 \quad R3 = R2 - R1$$

Ou

$$R3 = \text{DIFFERENCE} (R1, R2) \quad R3 = \text{DIFFERENCE} (R1, R2)$$

Instructions SQL :

```

SELECT Nom-Champ, Nom-Champ, ...
FROM Nom-Table2
MINUS
SELECT Nom-Champ, Nom-Champ, ...
FROM Nom-Table1;

```

```

SELECT Nom-Champ, Nom-Champ, ...
FROM Nom-Table1
MINUS
SELECT Nom-Champ, Nom-Champ, ...
FROM Nom-Table2;

```

Exemple :

Soient les tables ANGLAIS et ESPAGN qui permettent de gérer les candidats à des épreuves de langue.

ANGLAIS			
nocandidat	nomcandidat	prénomcandidat	numétablissement
10123	DUPONT	Patrick	94010
10216	GRANGETTE	Didier	75100
10309	HERISSON	Patrick	94010
10405	MAGNAN	Hélène	75250
10505	SERRES	Sylvia	94010

ESPAGN			
nocandidat	nomcandidat	prénomcandidat	numétablissement
10025	ACQUINO	Patricia	91260
10216	GRANGETTE	Didier	75100
10505	SERRES	Sylvia	94010

→ On souhaiterait obtenir la liste des candidats passant les épreuves d'anglais seulement.

En langage algébrique :

R1 = ANGLAIS - ESPAGN

Traduction en SQL :

```
SELECT *  
FROM ANGLAIS  
MINUS  
SELECT *  
FROM ESPAGN;
```

Il est possible de combiner les opérateurs UNION, INTERSECT et MINUS, il faut pour cela utiliser des parenthèses pour déterminer les ordres de priorité de ces opérateurs.

ANGLAIS			
nocandidat	nomcandidat	Prénomcandidat	numétablissement
10123	DUPONT	Patrick	94010
10309	HERISSON	Patrick	94010
10405	MAGNAN	Hélène	75250



Olivier Mondet
<http://unidentified-one.net>