

SQL Opération Sur Plusieurs Classes

T.Le, M. Léonard

{thang/leonard}@cui.unige.ch

<http://cui.unige.ch/~thang/support>

SYSINF-SES-CUI
UNIVERSITE DE GENEVE



Opération Sur Plusieurs Classes

1

CONTENU

- **Sélection des données dans plusieurs tables**
- Types de jointures
- A propos des jointures
- Q&A



Opération Sur Plusieurs Classes

2

Jointure

- Une *jointure* combine deux ou plusieurs tables afin d'en extraire des données
 - A l'aide de colonnes communes contenant les *clés primaires*



CONTENU

- Sélection des données dans plusieurs tables
- **Types de jointures**
 - **Jointures d'égalité (EQUI JOIN)**
 - Jointures naturelles (NATURAL JOIN)
 - Jointures de non égalité (NON-EQUI JOIN)
 - Jointures externes (OUTER JOIN)
 - Jointures réflexives (SELF JOIN)
- A propos des jointures



Jointures d'égalité

- **Jointure d'égalité** sert à joindre deux tables à l'aide d'une colonne commune qui est habituellement la clé primaire

- **Syntaxe:**

SELECT Table1.Colonne1, Table2.Colonne2,...

FROM Table1, Table2

WHERE

Table1.Colonne_Commune = Table2.Colonne_Commune



Opération Sur Plusieurs Classes

5

Jointures d'égalité

- *Ex: Affichez tous les noms, prénoms, adresses des clients qui effectuent des commandes datées après le 31.12.99*

COMMANDE

NumCde	NumClient	DateCde
10	1	15.12.99
20	2	20.01.00
30	1	18.06.00
40	3	20.09.00
50	1	30.12.00
60	2	07.01.01

CLIENT

NumClient	Nom	Prénom
1	MENU	NIOLAS
2	LEE	ROBERTO
3	BLOND	ALEXANDRE

Numclient: la clé primaire

SQL>

```
1 SELECT DISTINCT Nom, Prenom, AdresseClient FROM Client, Commande
2*  WHERE (Client.NumClient = Commande.NumClient) AND (DateCde > '31-12-99')
```

NOM PRENOM ADRESSECLIENT

```
-----
BLOND              ALEXANDRE    RUE DE LAUSANNE, 01
LEE                ROBERTO      MIREMONT, 46
MENU              NIOLAS        GENERAL DUFOUR, 24
```



Opération Sur Plusieurs Classes

6

Jointures d'égalité

```
SELECT * FROM Client, Commande  
WHERE (Client.NumClient = Commande.NumClient)
```

⇒ *CLIENT* (NumClient // Nom, Prénom, AdresseClient)
COMMANDE (NumCde // NumClient, DateCde, Total, AdresseLivraison)

```
SELECT Client.* FROM Client, Commande  
WHERE (Client.NumClient = Commande.NumClient)
```

⇒ *CLIENT* (NumClient // Nom, Prénom, AdresseClient)



Opération Sur Plusieurs Classes

7

Jointures d'égalité

- Jointures d'égalité (EQUI JOIN)
- **Jointures naturelles (NATURAL JOIN)**
- Jointures de non égalité (NON-EQUI JOIN)
- Jointures externes (OUTER JOIN)
- Jointures réflexives (SELF JOIN)



Opération Sur Plusieurs Classes

8

Jointures naturelles

- **Jointure naturelle**: presque à une *jointure d'égalité*
 - *Permet d'éliminer les valeurs dupliquées dans des colonnes jointes*

```
SELECT Client.NumClient, Nom, Prenom, AdresseClient,  
Commande.NumClient, NumCde, DateCde  
FROM Client, Commande  
WHERE (Client.NumClient = Commande.NumClient)
```



Types de jointures

- Jointures d'égalité (EQUI JOIN)
- Jointures naturelles (NATURAL JOIN)
- **Jointures de non égalité (NON-EQUI JOIN)**
- Jointures externes (OUTER JOIN)
- Jointures réflexives (SELF JOIN)



Jointures de non égalité

- **Jointure de non égalité** sert à joindre deux tables ou plus à partir d'une valeur de colonne spécifiée **différente** de la valeur de colonne spécifiée dans l'autre table

- **Syntaxe:**

SELECT Table1.Colonne1, Table2.Colonne2,...

FROM Table1, Table2

WHERE

Table1.Colonne_Commune *nonequi_opérateur*

Table2.Colonne_Commune

- NonEqui Opérateur: !=, <>, >, <, >=, <=, ...



Opération Sur Plusieurs Classes

11

Jointures de non égalité

- *Ex: Comparer les nombres de places passagers des voitures*

SQL>

```
1 SELECT U1.Nov, U1.Psg, U2.Nov, U2.Psg FROM Voiture U1, Voiture U2
2* WHERE U1.Psg < U2.Psg
```

NOU	PSG	NOU	PSG
12	4	11	5
12	4	13	5
11	5	14	6
12	4	14	6
13	5	14	6

- **Attention:** Utiliser des alias de table équivaut à renommer temporairement une table dans une instruction SQL



Opération Sur Plusieurs Classes

12

Types de jointures

- Jointures d'égalité (EQUI JOIN)
- Jointures naturelles (NATURAL JOIN)
- Jointures de non égalité (NON-EQUI JOIN)
- **Jointures externes (OUTER JOIN)**
- Jointures réflexives (SELF JOIN)



Jointures externes

- **Jointure externe** retourne toutes les lignes d'une table, même si celles-ci ne possèdent pas de correspondance dans la table jointe.
- **Syntaxe:**

```
SELECT Table1.Colonne1, Table2.Colonne2,...  
FROM Table1, Table2  
WHERE  
Table1.Colonne_Commune = Table2.Colonne_Commune (+)
```



Jointures externes

- Ex: Affichez tous les numéros des voitures (qui ont été réparées et n'ont pas été réparées) avec leur total des montants des réparations

VOITURE				REPARATION		
NOV	MV	KM	PSG	NOREP	NOV	NOG
11	Honda	5800	5	551	11	901
12	Ford	4000	4	552	11	902
13	Honda	290	5	553	11	901
14	BMW	100	6	554	12	902
				555	12	903

```

SQL>
1 SELECT Voiture.Nov, SUM(Px)
2 FROM Voiture, Reparation
3 WHERE Voiture.Nov = Reparation.Nov (+)
4* GROUP BY Voiture.Nov

```

NOV	SUM(PX)
11	12200
12	6000
13	
14	



Opération Sur Plusieurs Classes

15

Types de jointures

- Jointures d'égalité (EQUI JOIN)
- Jointures naturelles (NATURAL JOIN)
- Jointures de non égalité (NON-EQUI JOIN)
- Jointures externes (OUTER JOIN)
- Jointures réflexives (SELF JOIN)**



Opération Sur Plusieurs Classes

16

Jointures réflexive

- **Jointure réflexive** s'utilise pour joindre une table à elle même

- **Syntaxe:**

SELECT Alias1.Colonne1, Alias2.Colonne2,...

FROM Table1 **Alias1**, Table2 **Alias2**



Opération Sur Plusieurs Classes

17

Jointures réflexive

- Ex: Donnez la liste de toutes les numéros des commandes dont le total est supérieur ou égal au total de la commande n^{COMMANDE}

NumCde	NumClient	DateCde	Total	AdresseLivraison
10	1	15.12.99	32.500	Général Dufour, 24
20	2	20.01.00	24.000	Rue de Lyon, 03
30	1	18.06.00	2.500	Miremont, 29
40	3	20.09.00	6.000	Rue de Lausanne, 01
50	1	30.12.00	1.800	Général Dufour, 24
60	2	07.01.01	3.100	Rue de Genève, 120

SQL>

```
1 SELECT DISTINCT CDE1.NumCde,
2 FROM Commande CDE1, Commande CDE2
3 WHERE (CDE1.NumClient = CDE2.NumClient) AND
4 (CDE1.Total >= CDE2.Total) AND (CDE2.NumCde = 20)
```

NUMCDE

```
-----
10
20
```



Opération Sur Plusieurs Classes

18

CONTENU

- Sélection des données dans plusieurs tables
- Types de jointures
- **A propos des jointures**
- Q&A



A propos des jointures

Comment spécifier une jointure ?

A propos:

- Spécifiez la liste des tables (FROM ...)
- Spécifiez le prédicat de la jointure (WHERE ...)
- Spécifiez la liste des attributs (SELECT ...)
- Spécifiez le reste (WHERE, GROUPBY, HAVING,...)



A propos des jointures

Ex: Affichez tous les numéros, noms, prénoms, adresses des clients qui effectuent des commandes datées après le 31.12.99

1. Spécifiez la liste des tables (FROM ...)

PRODUIT (NumProd // LibelléProd, PrixUnitaireProd)

→ CLIENT (NumClient // Nom, Prénom, AdresseClient)

→ COMMANDE (NumCde // NumClient, DateCde, Total, AdresseLivraison)

LIGNECDE (NumCde, NumProd // QuantiteProdCde)

LIVRCDE (NumLiv, NumCde // DateLivr, EtatLivr)

LIVRCDEPRODUIT (NumLiv, NumCde, NumProd // QteLivrée)

RENOI (NumLiv, NumCde // DateRenvoi)

```
SELECT
FROM Client, Commande
WHERE
```



Opération Sur Plusieurs Classes

21

A propos des jointures

Ex: Affichez tous les numéros, noms, prénoms, adresses des clients qui effectuent des commandes datées après le 31.12.99

2. Spécifiez le prédicat de la jointure (WHERE ...)

CLIENT (NumClient // Nom, Prénom, AdresseClient)

COMMANDE (NumCde // NumClient, DateCde, Total, AdresseLivraison)

```
SELECT
FROM Client, Commande
WHERE Numclient = Numclient → Ambigu
```

→ Utiliser les alias

```
SELECT
FROM Client CL, Commande CDE
WHERE CL.Numclient = CDE.Numclient
```



Opération Sur Plusieurs Classes

22

A propos des jointures

Ex: Affichez tous les numéros, noms, prénoms, adresses des clients qui effectuent des commandes datées après le 31.12.99

3. Spécifiez la liste des attributs (SELECT ...)

CLIENT (NumClient // Nom, Prénom, AdresseClient)

COMMANDE (NumCde // NumClient, DateCde, Total, AdresseLivraison)

```
SELECT DISTINCT CL.Numclient, Nom, Prénom, AdresseClient  
FROM Client CL , Commande CDE  
WHERE CL.Numclient = CDE.Numclient
```



Opération Sur Plusieurs Classes

23

A propos des jointures

Ex: Affichez tous les numéros, noms, prénoms, adresses des clients qui effectuent des commandes datées après le 31.12.99

4. Spécifiez le reste (WHERE, GROUPBY, HAVING,...)

CLIENT (NumClient // Nom, Prénom, AdresseClient)

COMMANDE (NumCde // NumClient, DateCde, Total, AdresseLivraison)

```
SELECT DISTINCT CL.Numclient, Nom, Prénom, AdresseClient  
FROM Client CL , Commande CDE  
WHERE ( CL.Numclient = CDE.Numclient )  
AND (DateCde > '31/12/99')
```



Opération Sur Plusieurs Classes

24

CONTENU

- Sélection des données dans plusieurs tables
- Types de jointures
- A propos des jointures
- **Q&A**



CONTENU

- **Q&A**
 - *Restrictions des requêtes de regroupement*
 - *L'effet secondaire en comparant les chaînes de caractères*
 - *NOT NULL pour les attributs de clefs primaires*
 - *Type de donnée pour les attributs de clefs primaires*



Q&A 4

- *Restrictions des requêtes de regroupement*

- *Pourquoi l'expression suivant ne marche pas*

```
SQL>
1 SELECT reparation.nov, mv
2     FROM voiture, reparation
3     WHERE voiture.nov = reparation.nov
4     GROUP BY reparation.nov
5*    HAVING avg (px) BETWEEN 3000 AND 4000
SELECT reparation.nov, mv
*
```

```
ERROR at line 1:
ORA-00979: not a GROUP BY expression
```

→ *Parce que la colonne MV n'est pas explicitement spécifiée comme colonne de regroupement*



Opération Sur Plusieurs Classes

27

Q&A 4

- *Restrictions des requêtes de regroupement*

- **Proposition**

- Utiliser sous-requête

- Inclure simplement la colonne MV comme deuxième colonne de regroupement

```
SQL>
1 SELECT voiture.nov, mv
2     FROM voiture
3     WHERE voiture.nov IN
4     (SELECT reparation.nov
5     FROM reparation
6     GROUP BY reparation.nov
7*    HAVING avg (px) BETWEEN 3000 AND 4000)
no rows selected
```

```
SQL>
1 SELECT reparation.nov, mv
2     FROM voiture, reparation
3     WHERE voiture.nov = reparation.nov
4     GROUP BY reparation.nov, mv
5*    HAVING avg (px) BETWEEN 3000 AND 4000
no rows selected
```



Opération Sur Plusieurs Classes

28

Q&A 5

- *L'effet secondaire en comparant les chaînes de caractères*
 - *Pourquoi il faut utiliser la fonction de RTRIM() ou TRIM()*

```
SQL>
Wrote file afiedt.buf

      1* SELECT * FROM ch WHERE chauffeur LIKE 'L%E'
SQL> /

no rows selected

SQL>
Wrote file afiedt.buf

      1* SELECT * FROM ch WHERE TRIM(chauffeur) LIKE 'L%E'
SQL> /
```

NCH	CHAUFFEUR
4	LEE



Opération Sur Plusieurs Classes

29

Q&A 5

- *L'effet secondaire en comparant les chaînes de caractères*
 - *Pourquoi il faut utiliser la fonction de `RTRIM()` ou `TRIM()`*

chauffeur

⊕

L	E	E															
---	---	---	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--

TRIM(chauffeur)

L	E	E
---	---	---



Opération Sur Plusieurs Classes

30

Q&A 6

- *NOT NULL pour les attributs de clefs primaires*

- Est ce-que il faut déclarer NOT NULL pour les attributs de clefs primaires

→ Oracle assignera automatiquement NOT NULL sur les attributs de clefs primaires

```
SQL>
Wrote file afiedt.buf
```

```
1 CREATE TABLE ch2 (
2   nch NUMBER(3,0) CONSTRAINT pk_ch2 PRIMARY KEY,
3* chauffeur CHAR(40))
SQL> /
```

Table created.

```
SQL> DESCRIBE ch2;
```

Name	Null?	Type
NCH	NOT NULL	NUMBER(3)
CHAUFFEUR		CHAR(40)



Opération Sur Plusieurs Classes

31

Q&A 7

- *Type de donnée pour les attributs de clefs primaires*

- l'attribut NOV (table VOITURE) comme CHAR(12), pourquoi pas comme NUMBER(12,0) ?

→ Profiter les opérateurs et fonctions pour les chaînes de caractères

```
SQL>
1 SELECT voiture.*,ch.*
2 FROM voiture, v_ch, ch
3 WHERE
4* voiture.nov || ch.nch = v_ch.nov || v_ch.nch
```

NOU	MU	KM	PSG	NCH
CHAUFFEUR				
11 BOUBOU	HONDA	5600	5	1
11 LOULOU	HONDA	5600	5	2
12 BOUBOU	FORD	4000	4	1



Opération Sur Plusieurs Classes

32

Q&A 7

- *Type de donnée pour les attributs de clefs primaires*

– l'attribut NOV (table VOITURE) comme CHAR(12), pourquoi pas comme NUMBER(12,0) ?

→ *Profiter les opérateurs et fonctions pour les chaînes de caractères*

```
SQL>
1 SELECT voiture.*,ch.*
2 FROM voiture, v_ch, ch
3 WHERE
4*   voiture.nov || ch.nch = v_ch.nov || v_ch.nch
```

NOV	HV	KM	PSG	NCH
CHAUFFEUR				
11 BOUBOU	HONDA	5600	5	1
11 LOULOU	HONDA	5600	5	2
12 BOUBOU	FORD	4000	4	1



Opération Sur Plusieurs Classes

33

Q&A 8

- *Travailler à la distance?*
- *Stella - relational database navigator*
<http://cui.unige.ch/~thang/References.htm>

Stella - relational database navigator

Student:	Roberto Lopez Llamazares
Student:	Reuf Mustabasic
Assistant:	Slim Turki
Professor:	Michel Leonard

Welcome to Stella. Please choose your database :

☐ Oracle
☒ MySQL



Opération Sur Plusieurs Classes

34

Q&A 8

– Stella

Please, insert your connection settings :

Type of Data Base	Oracle
Driver	oracle.jdbc.OracleDriver
URL of the DB	cuisune.unige.ch
Connection port	1521
SID	CUIPRD
User name	msi2004_00
Password	*****



Opération Sur Plusieurs Classes

35

Q&A 8

– Stella

The information of the connection to the Data Base

Choose your schema :

Schemas: msi2004_00 Schemas found : (215)

User name :	msi2004_00
Selected schema :	msi2004_00
DataBase	Oracle
Version	Oracle8i Enterprise Edition Release 8.1.7.0.0 - Production JServer Release 8.1.7.0.0
JDBC Driver	Oracle JDBC driver
Version of the JDBC driver	10.1.0.2.0

For navigation :

To disconnect :



Opération Sur Plusieurs Classes

36

Q&A 8

Connection menu

Home

Change Schema

SQL Consoles

Logout

User: msi2004_00

Schema: msi2004_00

The tables

CLIENT

COMMANDE

LIGNECDE

LIVRCDE

LIVRCDEPRO

PRODUIT

RENOI

INFORMATION ABOUT THE SELECTED TABLE

Name of the Table :	Type of the Table :	Primary key :
CLIENT	TABLE	NUMCLIENT

See all relations
 Edit row
 Delete row

You have chosen the table (CLIENT)
Row(s) found : 3

Options	NUMCLIENT	NOM	PRENOM	ADRESSECLIENT
	1	MENU	NIOLAS	GENERAL DUFOUR, 24
	2	LEE	ROBERTO	MIREMONT, 46
	3	BLOND	ALEXANDRE	RUE DE LAUSANNE, 01

Insert values and clic on for validation

Opération Sur Plusieurs Classes

37

Q&A 8

Connection menu

Home

Change Schema

SQL Consoles

Logout

User: msi2004_00

Schema: msi2004_00

The tables

CLIENT

COMMANDE

LIGNECDE

LIVRCDE

LIVRCDEPRO

PRODUIT

RENOI

Today's date: dimanche, 5. décembre 2004

Put your SELECT query HERE

SELECT * FROM client WHERE nom LIKE 'L%'

EXECUTE SHOW EXAMPLE

Put your UPDATE/DELETE command HERE

EXECUTE SHOW EXAMPLE

Opération Sur Plusieurs Classes

38

Q&A 9

- **Syntaxe?**

- **BNF Index of SQL for Oracle**

<http://cui.unige.ch/~thang/References.htm>

index of rules

1. [alias](#)
2. [allocate_spec](#)
3. [alter_cluster](#)
4. [alter_cluster_spec](#)
5. [alter_command](#)
6. [alter_cost](#)
7. [alter_database](#)
8. [alter_index](#)
9. [alter_index_spec](#)
10. [alter_profile](#)
11. [alter_role](#)
12. [alter_rollback](#)
13. [alter_sequence](#)
14. [alter_sequence_spec](#)
15. [alter_session](#)
16. [alter_snapshot](#)
17. [alter_snapshot_log](#)
18. [alter_system](#)
19. [alter_table](#)



Opération Sur Plusieurs Classes

39

Q&A 10

- **Erreur?**

- **Oracle8 Server Error Messages 8.0**

<http://cui.unige.ch/~thang/References.htm>

01400-01489: SQL Execution Messages

This section lists messages generated during SQL execution.

ORA-01400: cannot insert NULL into ("USER"."TABLE"."COLUMN")

Cause: An attempt was made to insert a NULL into the column "USER"."TABLE"."COLUMN".

For example, if you enter:

```
connect scott/tiger

create table a (a1 number not null);

insert into a values (null);
```

Oracle returns:

```
ORA-01400 cannot insert NULL into ("SCOTT"."A"."A1") : which means you
cannot insert NULL into "SCOTT"."A"."A1".
```

Action: Retry the operation with a value other than NULL.



Opération Sur Plusieurs Classes

40