Extraction de donn´ees
Les fonctions
Les sous int´errogations
Les jointures
Les op´erateurs ensemblistes

Ines BAKLOUTI

ines.baklouti@esprit.tn

Ecole Sup´erieure Priv´ee d'Ing´enierie et de

Extraction de donn´ees
Les fonctions
Les sous int´errogations
Les jointures
Les op´erateurs ensemblistes

Extraction de donn´ees

Instruction SELECT
Restriction de donn´ees
Tri de donn´ees

Les fonctions

Les fonctions mono-ligne
Les fonctions de caract`eres
Les fonctions num´eriques
Les fonctions de dates
Les fonctions de conversion
Autres fonctions
Les fonctions analytiques

Les fonctions multi-lignes
Les sous int´errogations
Les sous-int´errogations monoligne
Les sous-int´errogations multi-lignes
Les jointures
Jointure interne
Jointure externe
Equijointure / Non-´equijointure
Auto-jointure
Jointure naturelle
Produit cart´esien
Les op´erateurs ensemblistes
L'op´erateur UNION
L'op´erateur UNION ALL
L'op´erateur INTERSECT
L'op´erateur MINUS

Extraction de donn´ees Instruction SELECT
Restriction de donn´ees Tri de donn´ees

Les fonctions

Les sous int´errogations
Les jointures

Les op´erateurs ensemblistes

Les fonctions
Les sous int´errogations
Les jointures
Les op´erateurs ensemblistes

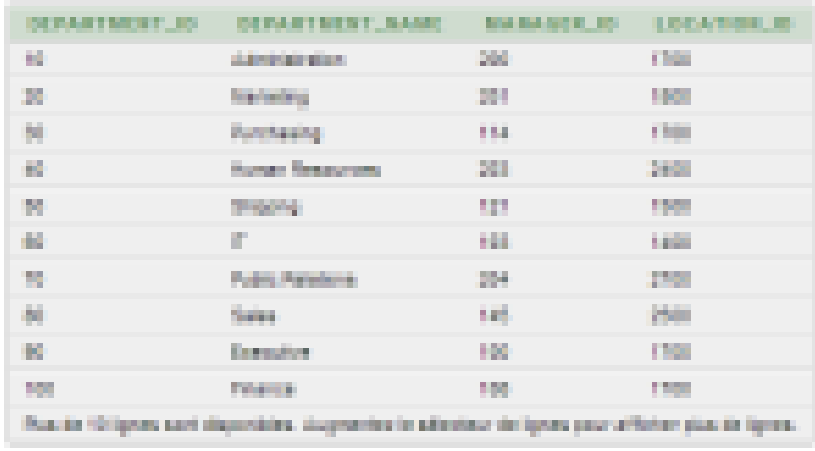
SELECT * | *{* [DISTINCT] <colonne> | <expression> [alias],...*}*
FROM <nom_table>;

SELECT : indique les colonnes `a afficher

DISTINCT : supprime les doublons

FROM : indique les tables contenant les colonnes

SELECT * FROM departments;



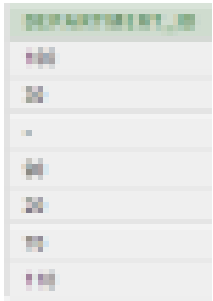
490

Les fonctions
Les sous int´errogations
Les jointures
Les op´erateurs ensemblistes

Restriction de donn´ees Tri de donn´ees

SELECT department_id, department_name FROM departments;

SELECT DISTINCT department_id FROM employees;

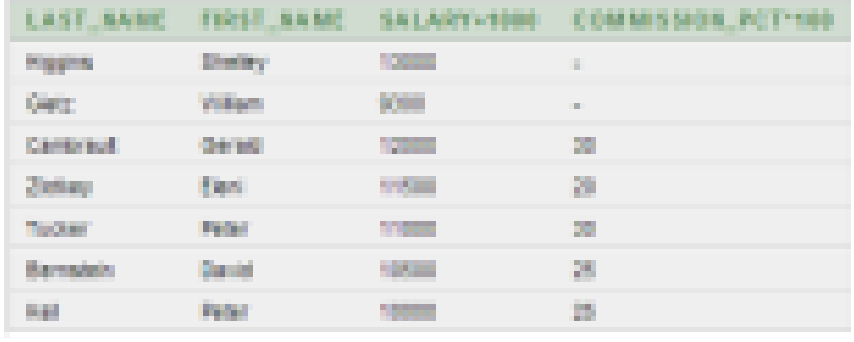


590

Expressions contenant des donn´ees de type NUMBER, DATE et des op´erateurs arithm´etiques

Opérateur	Description
+	Addition
-	Soustraction
*	Multiplication
/	Division

SELECT last_name, first_name, salary+1000, commission_pct*100



FROM employees;

NULL repr´esente une valeur non disponible, non affect´ee

NULL est diff´erente de z´ero, espace ou chaˆıne vide

SELECT last_name, first_name, commission_pct FROM employees;

LAST_NAME	FIRST_NAME	COMMISSION_PCT
Higgins	Shelley	-
Gietz	William	-
Cambrault	Gerald	,3
Zlotkey	Eleni	,2
Tucker	Peter	,3
Bernstein	David	,25
Hall	Peter	,25
Olsen	Christopher	,2
Cambrault	Nanette	,2

790

Les op´erateurs ensemblistes

Les fonctions

Les sous int´errogations Les jointures

Restriction de donn´ees Tri de donn´ees

Les expressions arithm´etiques comportant une valeur NULL

renvoient toujours une valeur NULL

SELECT last_name, salary, salary+commission_pct, salary*commission pct FROM employees;

LAST_NAME	SALARY	SALARY+COMMISSION_PCT	SALARY*COMMISSION_PCT
Higgins	12000	-	-
Gietz	8300	-	-
Cambrault	11000	11000,3	3300
Zlotkey	10500	10500,2	2100
Tucker	10000	10000,3	3000
Bernstein	9500	9500,25	2375
Hall	9000	9000,25	2250

Un alias de colonne :

Renomme un entˆete de colonne
Est utile avec les calculs
Suit imm´ediatement le nom d'une colonne (le mot cl´e facultatif AS peut ´egalement ˆetre utilis´e entre le nom de la colonne et l'alias)
N´ecessit´e des guillemets ("alias") s'il contient des espaces ou des caract`eres sp´eciaux (# $), ou s'il distingue les majuscules des minuscules

SELECT last_name nom, first_name AS pr´enom, salary*12 "revenu annuel" FROM employees;

NOM	PRÉNOM	Revenu Annuel
King	Steven	288000
Kochhar	Neena	204000
De Haan	Lex	204000
Hunold	Alexander	108000
Ernst	Bruce	72000
Austin	David	57600
Pataballa	Valli	57600
Lorentz	Diana	50400

Concat`ene des colonnes ou des chaines de caract`eres

Est repr´esent´e par le symbole ||

La colonne r´esultante est une expression de type carat`ere

SELECT department_id||' ** '||department_name AS "d´epartement"

from departments;

Département
10 [illegible] Administration
20 [illegible] Marketing
30 [illegible] Purchasing
40 [illegible] Human Resources
50 [illegible] Shipping
60 [illegible] IT
70 [illegible] Public Relations
80 [illegible] Sales

1090

Les op´erateurs ensemblistes

Les fonctions

Instruction SELECT

Les sous int´errogations Les jointures

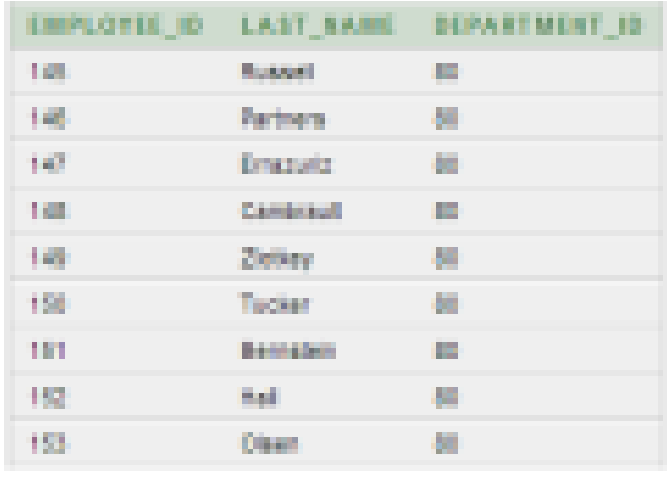
Tri de donn´ees

Restreindre les lignes renvoy´ees `a l'aide la clause WHERE

SELECT * | *{* [DISTINCT] <colonne> | <expression> [alias],...*}*

FROM <nom_table>
[WHERE <condition(s)>];

SELECT employee_id, last_name, department_id
FROM employees
WHERE department_id= 80;



1190

Les fonctions
Les sous int´errogations
Les jointures
Les op´erateurs ensemblistes

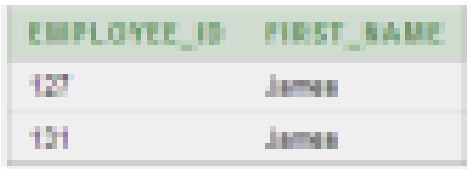
Les chaˆınes de caract`eres et les dates sont incluses entre apostrophes.

Les valeurs de type caract`ere distinguent les majuscules des minuscules

Les valeurs de type date sont sensibles au format

Le format de date par d´efaut est DD-MM-RR

SELECT employee_id,first_name FROM employees WHERE first name='James';



EMPLOYEE_ID	FIRST_NAME
127	James
131	James

SELECT employee_id,first_name FROM employees WHERE first name='JAMES'; => aucune ligne s´electionn´ee

Les fonctions
Les sous int´errogations
Les jointures
Les op´erateurs ensemblistes

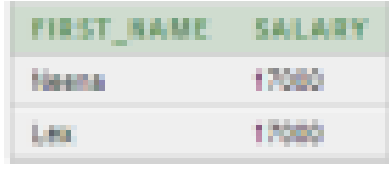
Instruction SELECT

Tri de donn´ees

Opérateur	Description
=	Egal à
<	Inférieur à
<=	Inférieur à ou égal
>	Supérieur à
>=	Supérieur à ou égale à
<> ou !=	Différent
BETWEEN val1 AND val2	Valeur comprise entre val1 et val2
In (val1,val2,…,valN)	Appartient à une liste de valeurs
Like	Correspond à un modèle de chaînes de caractères
IS NULL	Correspond à une valeur NULL

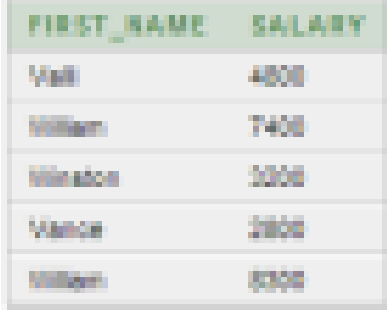
Les fonctions
Les sous int´errogations
Les jointures

SELECT first_name, salary FROM employees WHERE salary BETWEEN 15000 AND 20000;



FIRST_NAME	SALARY
Neena	17000
Lex	17000

SELECT first_name, salary FROM employees WHERE first_name BETWEEN ’V’ AND ’X’;



FIRST_NAME	SALARY
Vail	4800
William	7400
Winston	3200
Vance	2800
William	8300

Instruction SELECT

Tri de donn´ees

1490

Les fonctions
Les sous int´errogations

Les jointures
Les op´erateurs ensemblistes

SELECT first_name, department_id FROM employees WHERE department_id IN (10,20);

SELECT first_name, last_name FROM employees WHERE first_name

IN (’James’,’David’,’Diana’);
1590

Les fonctions
Les sous int´errogations
Les jointures
Les op´erateurs ensemblistes

Instruction SELECT
donn´ees

Tri de

L’op´erateur LIKE permet de rechercher des cha ınes de caract`eres `a l’aide de caract`eres g´en´eriques

Les conditions de recherche peuvent contenir des caract`eres ou des nombres litt´eraux

% repr´esente Z´ero ou plusieurs caract`eres
repr´esente un caract`ere

```
SELECT first_name FROM employees
WHERE first_name LIKE 'S_e%' ;
```

1690

Les fonctions
Les sous int´errogations
Les jointures
Les op´erateurs ensemblistes

Instruction SELECT

Tri de donn´ees

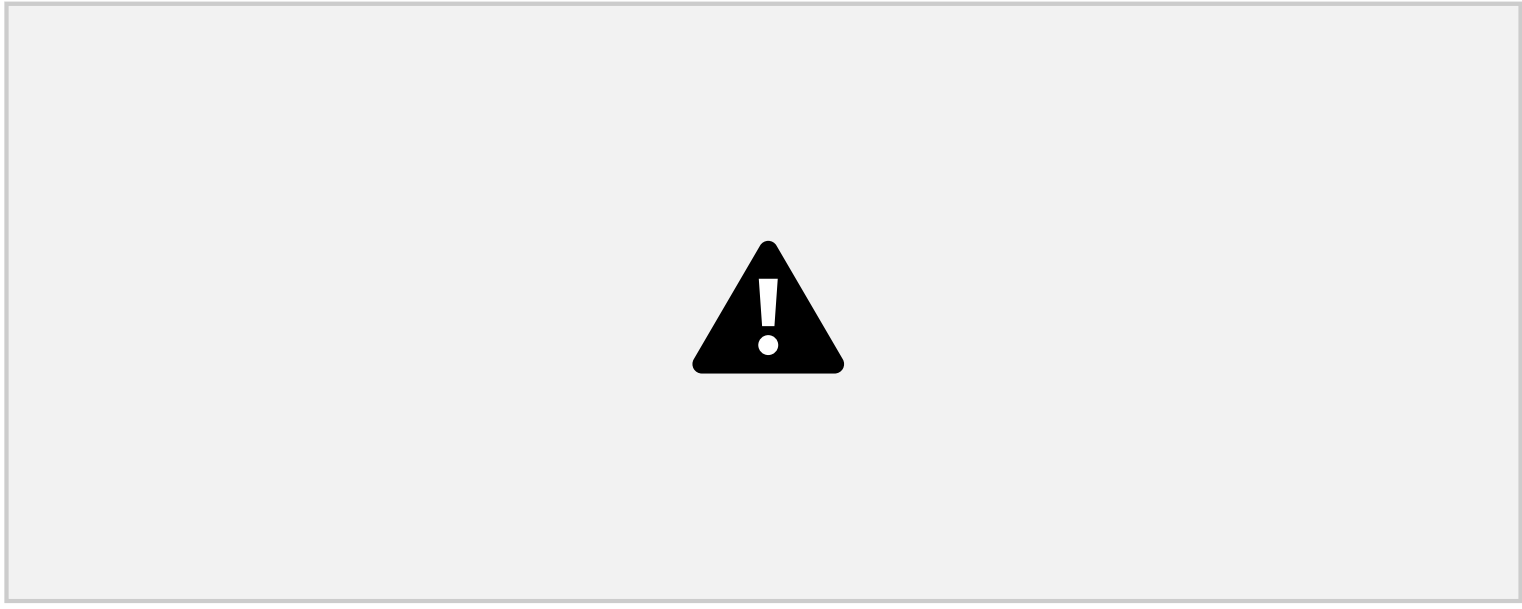
L’op´erateur IS NULL permet de tester la pr´esence de valeurs NULL.

```
SELECT first_name, manager_id FROM employees
WHERE manager_id IS NULL;
```


1790

Instruction SELECT
donn´ees

Tri de



1890

```
SELECT last_name, job_id, salary
FROM employees
```



```
WHERE salary >=10000 AND job_id LIKE '%MAN%';
```

1990

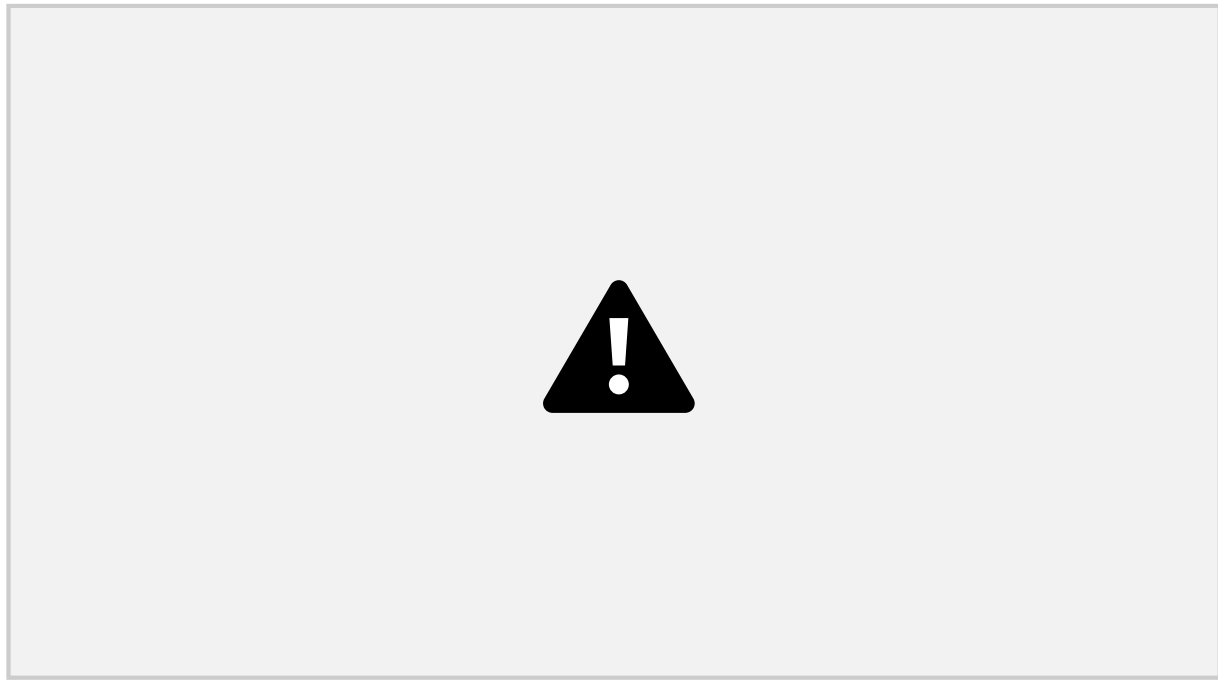
Les fonctions
Les sous int´errogations
Les jointures
Les op´erateurs ensemblistes

SELECT last_name, job_id, salary
FROM employees

Instruction SELECT

Tri de donn´ees

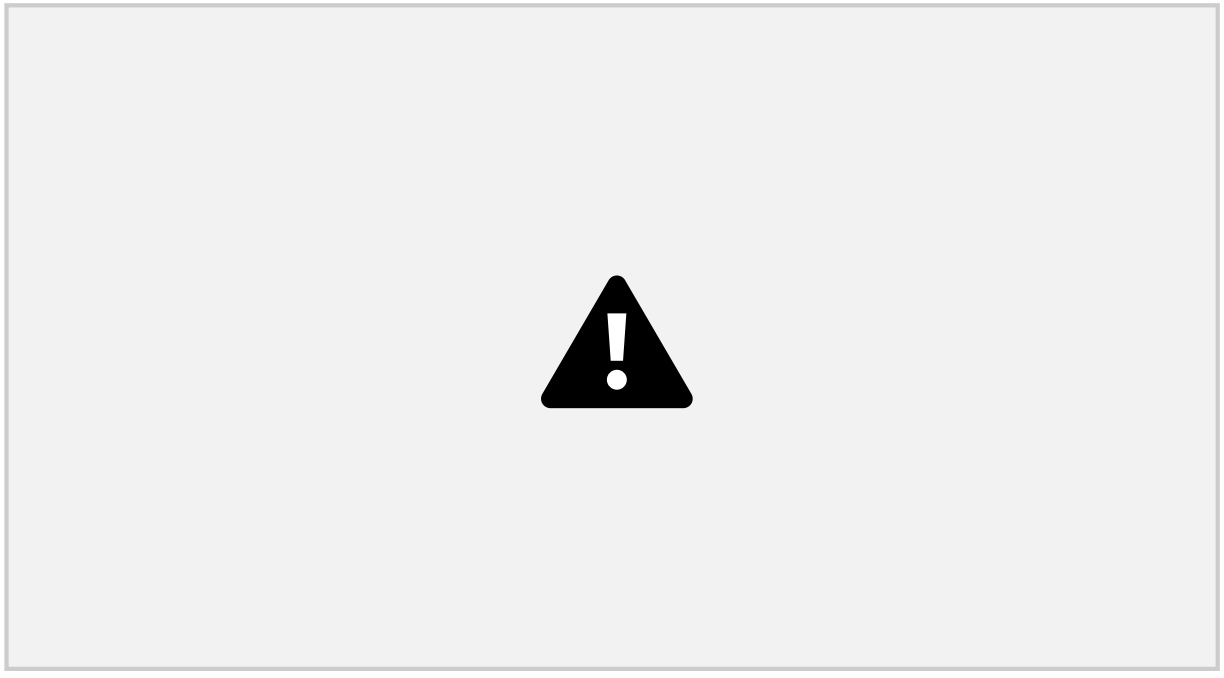
WHERE salary >=10000 OR job_id LIKE ’%MAN%’;



2090

SELECT last_name, job_id FROM employees

WHERE job_id NOT IN (’IT_PROG’, ’ST_CLERK’, ’SA_REP’) ;

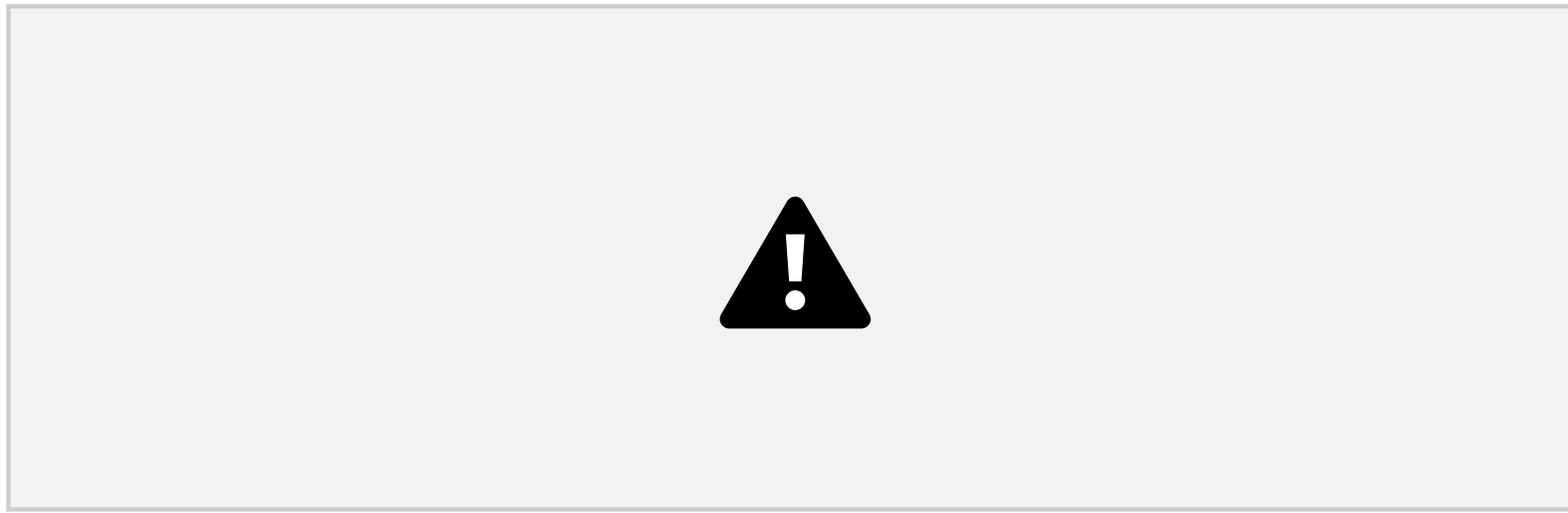


2190

Les fonctions
Les sous int´errogations
Les jointures
Les op´erateurs ensemblistes

Instruction SELECT

Tri de donn´ees



2290

Les fonctions
Les sous int´errogations
Les jointures
Les op´erateurs ensemblistes

SELECT last_name, job_id, salary

FROM employees

donn´ees

Instruction SELECT

Tri de

WHERE job_id= ’SA_MAN’ OR job_id= ’AD_VP’ AND salary > 12000;

SELECT last_name, job_id, salary FROM employees
WHERE (job_id= ’SA_MAN’ OR job_id= ’AD_VP’) AND salary > 12000;

La clause ORDER BY:

permet de triez les lignes extraites

ASC : ordre croissant (par d´efaut)
DESC : ordre d´ecroissant

toujours la derni`ere clause dans l'instruction SELECT

SELECT * | *{* [DISTINCT] <colonne> | <expression> [alias],...*}*
FROM <nom_table>;
[WHERE condition(s)]
[ORDER BY *{*<colonne>, <expression>, <alias>*}* [ASC | DESC]] ;

SELECT last_name, hire_date FROM employees
ORDER BY hire_date DESC ; –ou bien ORDER By 2 DESC

2490

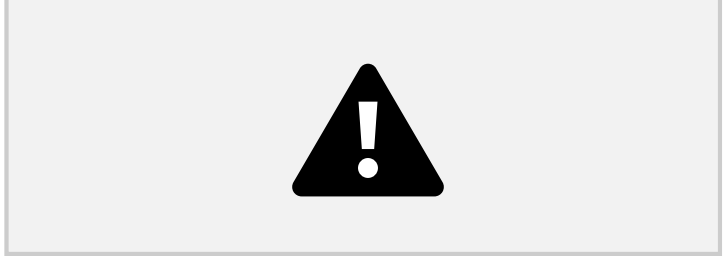
Les op´erateurs ensemblistes

Les fonctions

Les sous int´errogations Les jointures

Instruction SELECT
Restriction de donn´ees

SELECT employee_id, last_name, salary*12 ”Salaire Annuel”
FROM employees
ORDER BY ”Salaire Annuel”; –ou bien ORDER By 3



```
SELECT employee_id, last_name, salary*12 ”Salaire Annuel”
FROM employees
ORDER BY ”Salaire Annuel”, last_name DESC; —ou bien ORDER By
```

```
3, 2 DESC
```

2590

Les fonctions mono-ligne
Les fonctions analytiques
Les fonctions multi-lignes

Extraction de donn´ees
Les sous int´errogations Les jointures
Les op´erateurs ensemblistes

Les fonctions
Les fonctions mono-ligne Les fonctions analytiques Les fonctions multi-lignes

Les sous int´errogations
Les jointures

Les op´erateurs ensemblistes

Les fonctions mono-ligne
Les fonctions analytiques
Les fonctions multi-lignes

Il existe 3 types de fonctions dans le language SQL

Fonctions mono-ligne: manipulent une seule ligne et ram`enent un seul r´esultat

Fonctions analytiques: manipulent plusieurs lignes et ram`enent un plusieurs r´esultats

Fonctions multi-lignes: manipulent plusieurs lignes et ram`enent un seul r´esultat

Les fonctions analytiques
Les fonctions multi-lignes

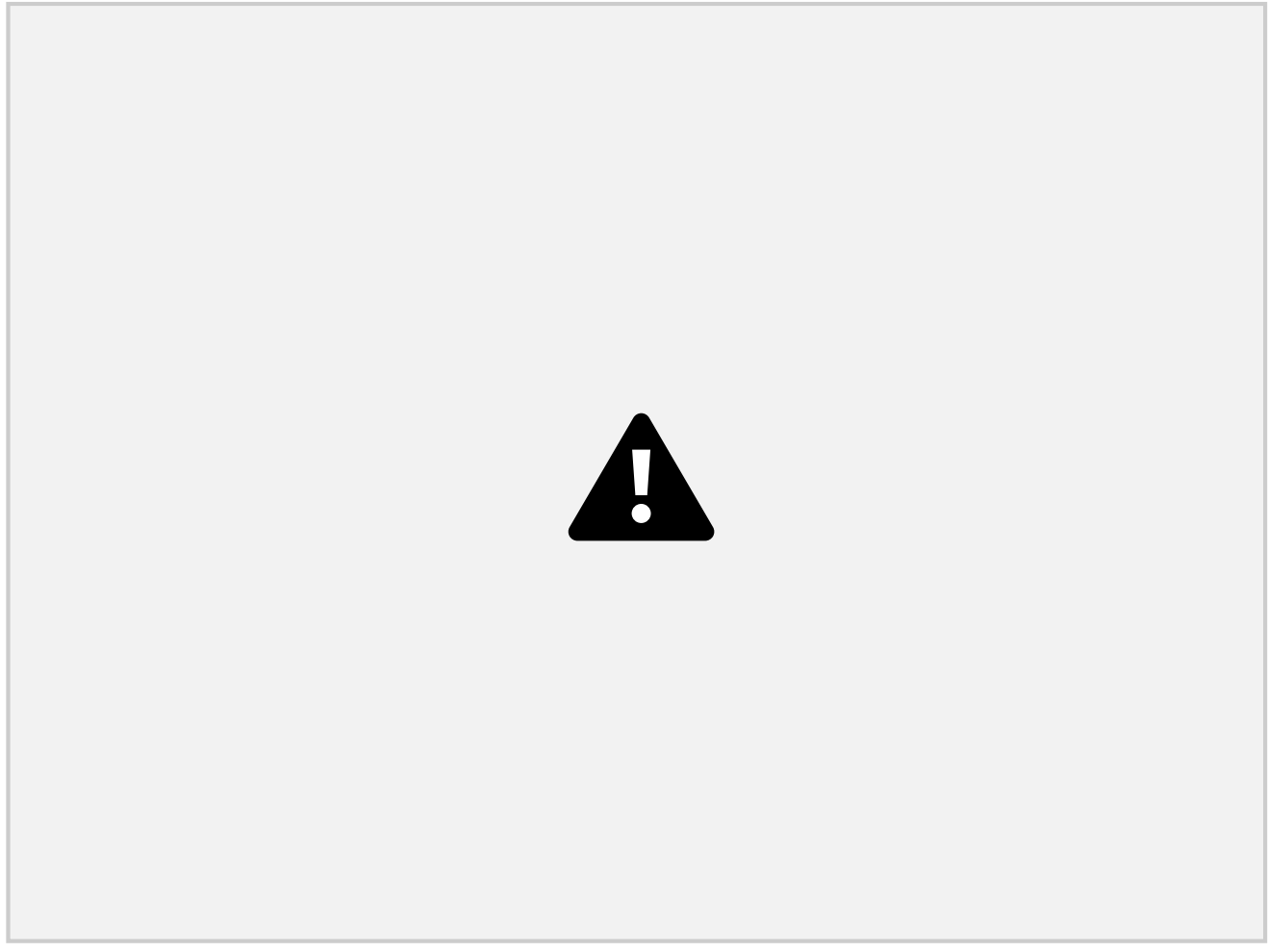
SELECT first_name, lower(first_name) , upper(first_name), initcap(first name) FROM employees;

2890

Extraction de donn´ees

Les sous int´errogations
Les jointures
Les op´erateurs ensemblistes

Les fonctions analytiques
Les fonctions multi-lignes



Extraction de donn´ees

Les op´erateurs ensemblistes

Les sous int´errogations
Les jointures

```
CONCAT(first_name,last_name) ”Nom
et pr´enom”, LENGTH (last_name)
”longueur nom”,
INSTR(last_name,’a’) ”position a”,
LPAD(last_name,10,’*’),
RPAD(last_name,10,’*’)
FROM employees
WHERE SUBSTR(job_id, 4) = ’REP’;
```

Les fonctions analytiques
Les fonctions multi-lignes

```
SELECT first_name, last_name, job_id,
```

3090

Extraction de donn´ees

Les sous int´errogations
Les jointures
Les op´erateurs ensemblistes

```
SELECT commission_pct+0.2,
ROUND(commission_pct+0.2),
TRUNC(commission_pct+0.2),
FLOOR(commission_pct+0.2),
CEIL(commission_pct+0.2)
FROM employees where department
id=80;
```

3290

Extraction de donn´ees

Les sous int´errogations
Les jointures
Les op´erateurs ensemblistes

Dans la base de donn´ees Oracle, les dates sont stock´ees dans un format num´eriques interne: si`ecle, ann´ee, mois, jour, heures, minutes et secondes

Le format de date par d´efaut est ’DD-MON-YY’

SYSDATE est une fonction qui renvoie :

La date
L’heure

Extraction de donn´ees

Les sous int´errogations
Les jointures
Les op´erateurs ensemblistes

Calcul arithm´etique sur des dates:

Les fonctions analytiques
Les fonctions multi-lignes

ajout ou soustraction d'un nombre de jour `a une date afin d'obtenir une date
r´esultante
Ajout ou soustraction d'un nombre d'heures `a une date en divisant le nombre d'heures par 24
soustraction d'une date d'une autre afin de d´eterminer le nombre de jours entre les deux dates

```
SELECT first_name,
(SYSDATE-hire_date) AS jours,
(SYSDATE-hire_date)/7 AS semaines
FROM employees;
```

3490

Extraction de donn´ees

Les sous int´errogations
Les jointures
Les op´erateurs ensemblistes

Les fonctions analytiques
Les fonctions multi-lignes

Extraction de donn´ees

Les sous int´errogations
Les jointures

Les op´erateurs ensemblistes

Ann´ee, extract(month from systimestamp) as Mois, extract(day from systimestamp) as Jour, extract(hour from systimestamp) as Hour, extract(minute from systimestamp) as Minutes, extract(second from systimestamp) as Secondes FROM dual;

SELECT systimestamp,
extract(year from systimestamp) as

3690

Extraction de donn´ees

Les sous int´errogations
Les jointures
Les op´erateurs ensemblistes

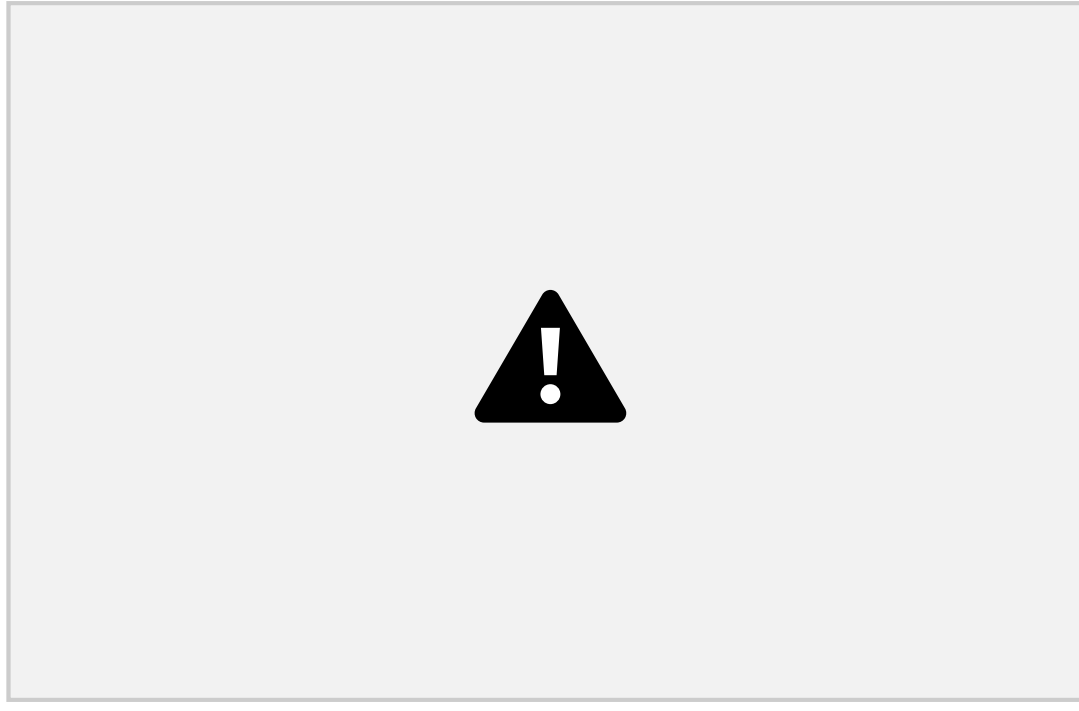
Pour les affectations, le serveur Oracle peut convertir
automatiquement les types de donn´ees suivants:

L’expression salary=‘2000’ entraˆıne la conversion implicite de la chaˆıne ‘2000’ en valeur num´erique 2000

L’expression hire_date>’01-Jan-90’ entraˆıne la

conversion implicite de la chaˆıne ’01-Jan-90’ en date

3790



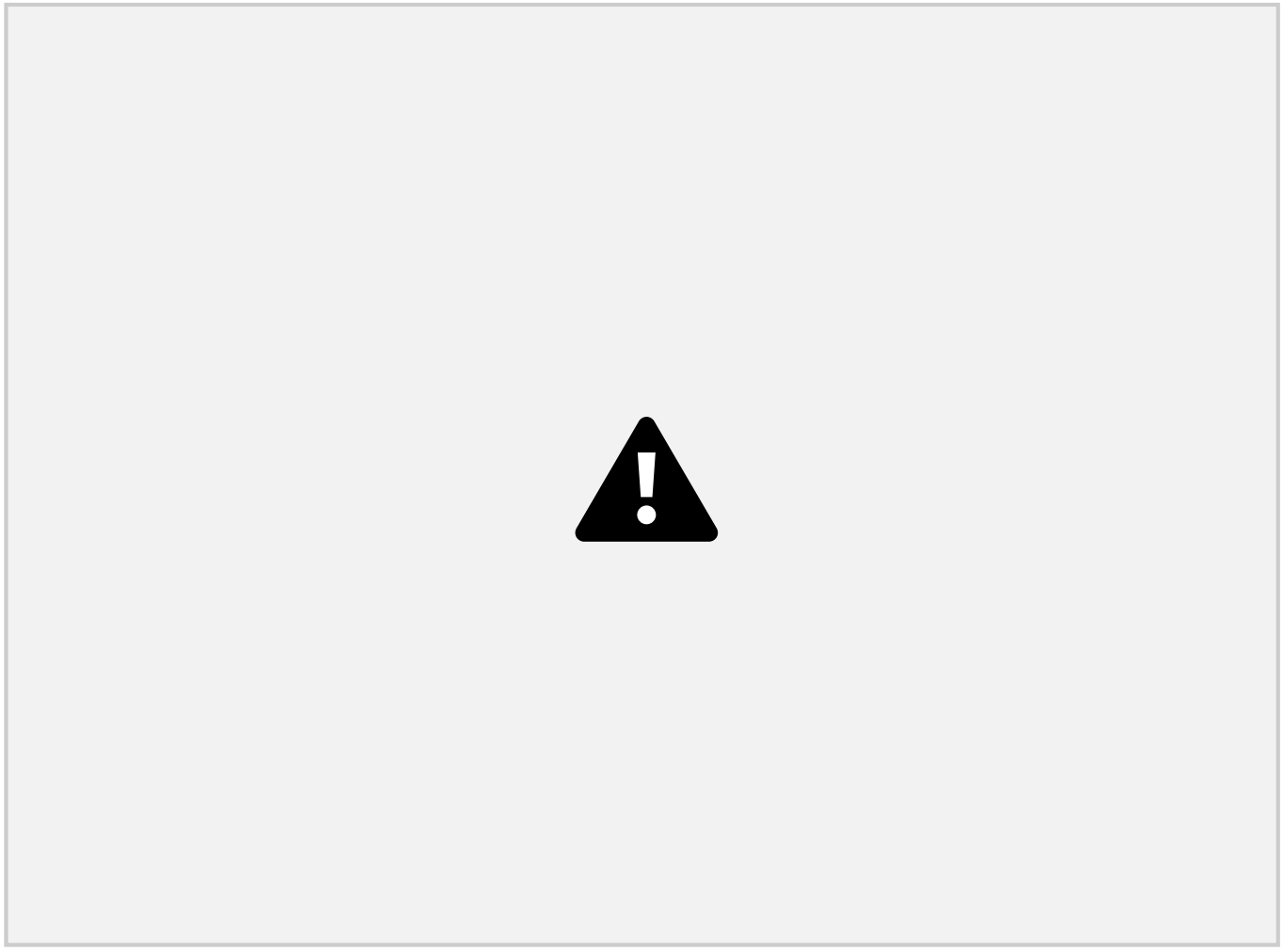
3890

Extraction de donn´ees

Les sous int´errogations
Les jointures
Les op´erateurs ensemblistes

Les fonctions analytiques

Les fonctions multi-lignes



Extraction de donn´ees

Les sous int´errogations

Les jointures

Les op´erateurs ensemblistes

SELECT last_name, TO_CHAR(hire_date, 'fm DD Month YYYY') AS HIREDATE FROM employees;

fm permet de supprimer les espaces de remplissage ou les z´eros de d´ebut

SELECT first_name, TO_CHAR(salary, '$99,000.00') SALARY FROM

employees;

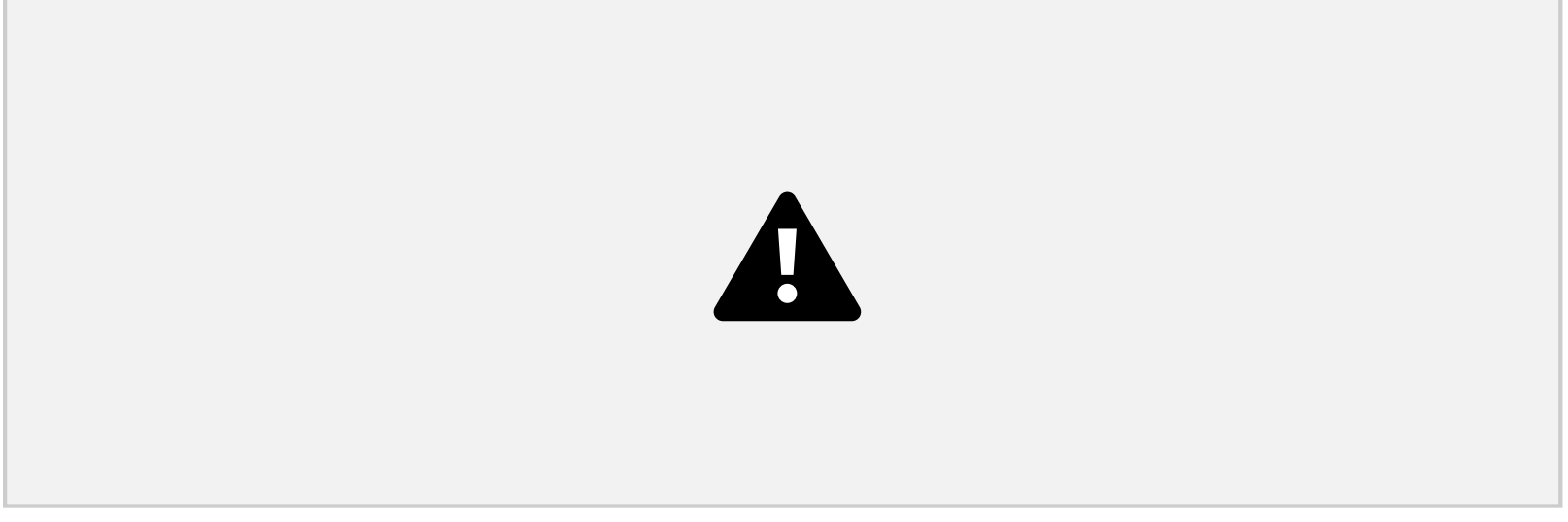
4090

Extraction de donn´ees

Les sous int´errogations
Les jointures
Les op´erateurs ensemblistes

Les fonctions analytiques

Les fonctions multi-lignes



select first_name, hire_date from emp where hire_date>to_date('01/01/1982','DD MM-YYYY');

Select first_name, salary from employees where salary>=to number('15000');

4190

La fonction decodepermet de faire un traitement conditionnel sur les donn´ees :

Decode (expr, val1, res1, val2, res2, ... ValN, resN, default)

retourne res1 si expr = val1, res2 si expr=val2,...,resN si expr=valN sinon default

SELECT first_name,department_id, decode(department_id,10, ’ACCOUNTING’, 20, ’RESEARCH’, ’DEP. INCONNU ’) AS ”NOM DEPARTEMENT” FROM employees;

Les fonctions analytiques
Les fonctions multi-lignes

Extraction de donn´ees
Les sous int´errogations Les jointures
Les op´erateurs ensemblistes

NVL(expr,val): retourne val si expr est NULL

NVL2(expr,val1,val2): retourne val1 si expr est NOT NULL et val2 si expr est NULL

expr peut ˆetre de type date, les caract`ere et valeur num´erique. Les types de donn´ees de expr et val doivent correspondre.

SELECT first_name, salary, commission_pct,
NVL(commission_pct,0),
NVL(to_char(commission_pct), 'Pas de commission') AS "commission ?",
NVL2(commission_pct,commission_pct*salary,0) AS "commision",
to_char(NVL2(commission_pct,commission_pct*100/salary,0)) || '%' AS "pourcentage commission" FROM employees;

4390

Les fonctions analytiques
Les fonctions multi-lignes

Extraction de donn´ees
Les sous int´errogations Les jointures
Les op´erateurs ensemblistes

NULLIF (expr1, expr2) : retourne NULL si expr1= expr2, sinon retourne expr1

```
SELECT first_name, LENGTH(first_name) nbr1, last_name,
LENGTH(last_name) nbr2, NULLIF(LENGTH(first_name),
LENGTH(last_name)) result FROM employees;
```

Coalesce (exp1,expr2,expr3,. . .) : retourne la premi`ere valeur non nulle

select Coalesce(NULL,1,NULL,7) from dual;
=> retourne 1

La fonction case ´evalue une liste de conditions et retourne un r´esultat parmi les cas possibles

case <expression>
when <valeur1> then <resultat1>
..
.

when <valeurN> then <resultatN>
else resultat
end

case
when <condition1> then <resultat1>
..
.
when <conditionN> then <resultatN>
else resultat
end

Les fonctions analytiques
Les fonctions multi-lignes

Extraction de donn´ees
Les sous int´errogations Les jointures
Les op´erateurs ensemblistes

```
SELECT first_name, department_id,
case department_id
when 10 then 'Accounting'
when 20 then 'RESEARCH'
else 'INCONNU'
end as departement
FROM employees;
```

Les fonctions analytiques calculent une valeur globale bas´ee sur un groupe de lignes. Ils diff`erent des fonctions de groupe (ou d'agr´egation) en ce qu'ils renvoient plusieurs lignes pour chaque groupe.

Les fonctions analytiques sont la derni`ere s´erie d'op´erations effectu´ees dans une requ^ete `a l'exception de la clause finale ORDER BY. Par cons´equent, elles analytiques ne peuvent appara^ıtre que dans la liste de s´election ou clause ORDER BY.

fonction_analytique(expression) OVER([clause_partitionnement] [clause_ordre])

clause_partitionnement: sous forme PARTITION BY expression1,expression2,...,expressionN : d´efinit les groupes de partitionnement

clause_ordre: sous forme ORDER BY expression1,expression2,...,expressionN [NULLSFIRST|LAST] : d´efinit l'ordre `a l'int´erieur de chaque partition

NULLSFIRST/LAST: indique si les valeurs nulles seront en premier

ordre/dernier ordre

Extraction de donn´ees

Les sous int´errogations
Les jointures
Les op´erateurs ensemblistes

Les fonctions mono-ligne
Les fonctions multi-lignes

La fonction row_number() retourne le num´ero s´equentiel d'une ligne dans une partition de r´esultats, en commen¸cant `a 1 pour la premi`ere ligne de chaque partition.

SELECT employee_id,department_id, salary, row_number() over(order by salary DESC) "N° Salaire" FROM employees ;

4990

Extraction de donn´ees

Les fonctions mono-ligne

Les sous int´errogations

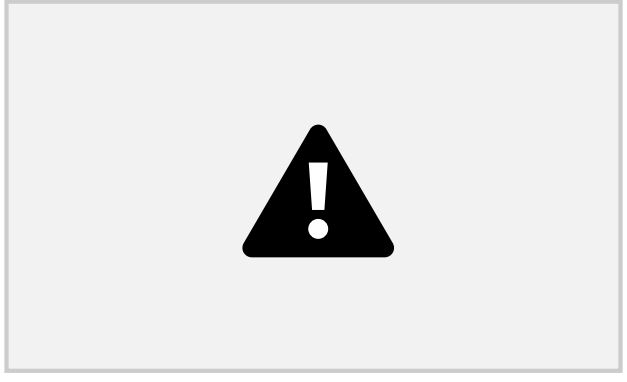
Les jointures

Les fonctions multi-lignes

Les op´erateurs ensemblistes

```
SELECT employee_id,department_id,
 salary, row_number() over(partition by
department_id order by salary DESC)
”Rang Salaire” FROM employees;
```

```
SELECT employee_id,department_id,job_id,
salary, row_number() over(partition by
department_id,job_id order by salary DESC)
”Rang Salaire” FROM employees;
```



5090

Extraction de donn´ees

Les sous int´errogations
Les jointures
Les op´erateurs ensemblistes

Les fonctions mono-ligne
Les fonctions multi-lignes

La fonction rank() retourne le rang dechaque ligne au sein de la partition d’un ensemble de r´esultats

SELECT employee_id,department_id, salary, rank() over(partition by department_id order by salary DESC) ”Rang Salaire” FROM employees;

5190

La fonction dense_rank() retourne le rang des lignes `a l'int´erieur de la partition d'un ensemble de r´esultats,sans aucun vide dans le classement

```
SELECT employee_id,department_id, salary, dense_rank()
over(partition by department_id order by salary DESC) ”Rang Salaire”
FROM employees;
```

5290

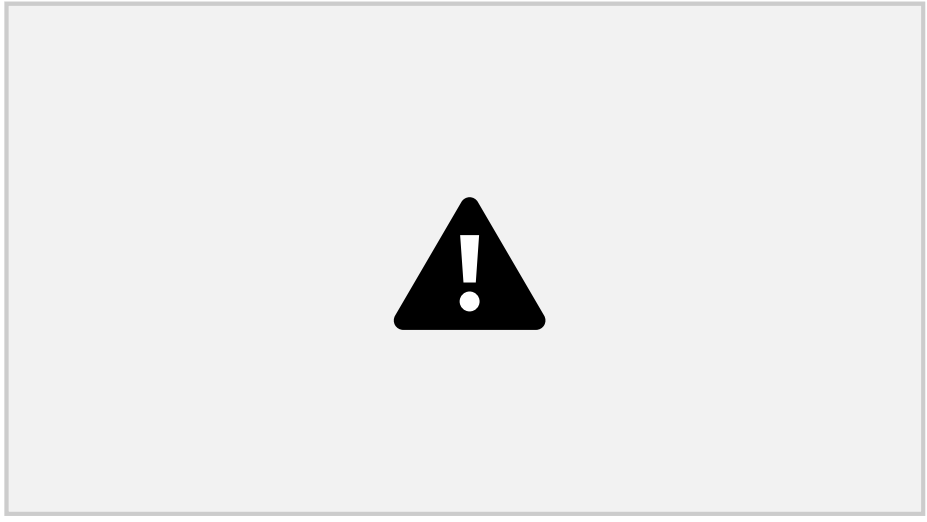
Extraction de donn´ees

Les sous int´errogations
Les jointures
Les op´erateurs ensemblistes

Les fonctions mono-ligne
Les fonctions multi-lignes

La fonction first_value() retourne la premi`ere valeur d’une partition

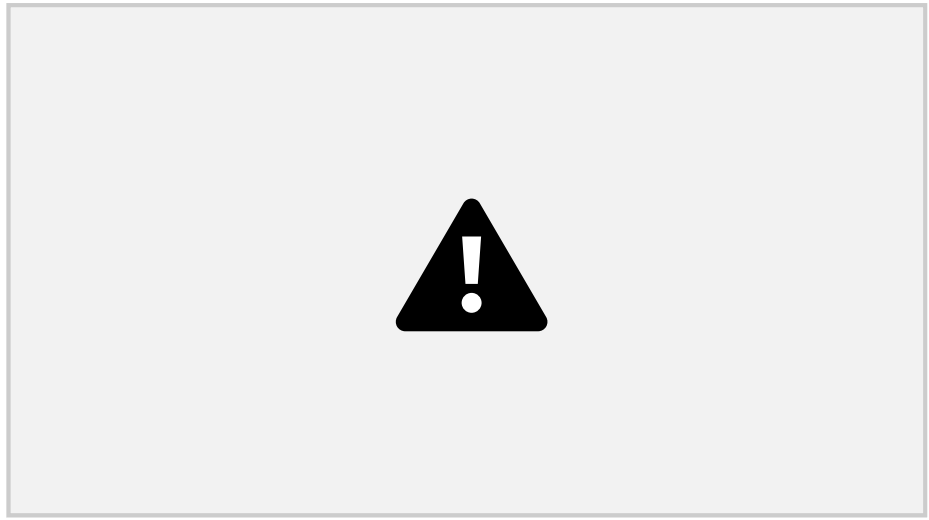
SELECT employee_id,department_id, salary, first_value(salary) over(partition by department_id order by salary) as first_valeur from employees;



5390

La fonction last_value() retourne la derni`ere valeur d’une partition

SELECT employee_id,department_id, salary, last_value(salary) over(partition by department_id order by salary) as last_valeur from employees;



5490

Les fonctions multi-lignes (appel´ees aussi de groupe ou d'agr´egation) op`erent sur des ensembles de lignes afin de renvoyer un seul r´esultat par groupe.

Les fonctions de groupe les plus utilis´ees:

AVG([distinct | all] expr) : valeur moyenne en ignorant les valeurs NULL COUNT ([* | distinct | all] expr) : nombre de lignes o`u expr est diff´erente de NULL. Le caract`ere * comptabilise toutes les lignes s´electionn´ees MAX ([distinct | all] expr) : valeur maximale en ignorant les valeurs NULL MIN([distinct | all] expr) : valeur minimale en ignorant les valeurs NULL STDDEV([distinct | all] expr) : ecart-type en ignorant les valeurs NULL SUM([distinct | all] expr) : somme en ignorant les valeurs NULL

VARIANCE([distinct | all] expr) : variance en ignorant les

valeurs NULL 5590

```
SELECT [colonne,] fonction_groupe(colonne), ...
FROM <nom_table>
[WHERE <condition>]
[GROUP BY colonne]
[ORDER BY colonne];
```

```
SELECT trunc(AVG(salary),3), SUM(salary), MAX(hire_date), MIN(hire
date)
FROM employees
WHERE department_id in(80,90);
```

5690

```
SELECT COUNT(*)
FROM employees
WHERE department_id in(80,90);
```


Les fonctions mono-ligne Les fonctions analytiques

=>retourne le nombre de lignes qui v´erifient la condition de la clause WHERE

```
SELECT COUNT(commission_pct) ”count”, COUNT(all commission
pct) ”all”, COUNT(DISTINCT commission_pct) ”distinct”
```

```
FROM employees
WHERE department_id in(80,90);
```


=>COUNT(expr) /COUNT(all expr): retourne le nombre de ligne ayant des valeurs non NULL de expr
=>COUNT(distinct expr): retourne le nombre de ligne ayant des valeurs non NULL et DISTINCT de expr

5790

Extraction de donn´ees

Les sous int´errogations
Les jointures
Les op´erateurs ensemblistes

Les fonctions mono-ligne Les fonctions analytiques

Les fonctions de groupe ignorent les valeurs NULL de la colonne

Exemple:

```
SELECT trunc(AVG(commission_pct) ,3) FROM
employees;
```


La fonction NVL force les fonctions de groupe `a inclure les valeurs NULL

Exemple:

```
SELECT trunc(AVG( NVL(commission_pct,0) ) ,3)
FROM employees;
```

Extraction de donn´ees

Les sous int´errogations
Les jointures
Les op´erateurs ensemblistes

```
SELECT department_id,
trunc(AVG(salary),3) FROM
employees WHERE department_id
in(80,90) GROUP BY department_id;
```


Notez Bien

Les fonctions mono-ligne Les fonctions analytiques

Toute colonne ou expression de la liste SELECT qui ne constitue pas une fonction d'agr´egation doit figurer dans la clause GROUP BY

Exemple:

```
SELECT department_id, job_id, trunc(AVG(salary),3)
FROM employees WHERE department_id in(80,90)
GROUP BY department_id, job_id
```


5990

La clause HAVING permet de restreindre l'affichage des r´esultats de groupes `a ceux qui v´erifient la condition dans cette clause

Les lignes sont regroup´ees
La fonction de groupe est appliqu´ee
Les groupes qui correspondent `a la clause HAVING sont affich´es

SELECT [colonne,] fonction_groupe(colonne), ...
FROM <nom_table>
[WHERE <condition>]

[GROUP BY colonne]
[HAVING <condition_groupe>]
[ORDER BY colonne];

Extraction de donn´ees

Les sous int´errogations

Les jointures
Les op´erateurs ensemblistes

ORDER BY department_id;

Les fonctions mono-ligne
Les fonctions analytiques

SELECT department_id, MAX(salary)
FROM employees
WHERE job_id LIKE '%REP'
GROUP BY department_id

SELECT department_id, MAX(salary) FROM employees
WHERE job_id LIKE '%REP' GROUP BY department_id
HAVING MAX(salary)>=10000

ORDER BY department_id;

6190

L’op´erateur ROLLUP : calcule des agr´egats (SUM, COUNT, MAX, MIN, AVG) `a tous les niveaux de totalisation sur une hi´erarchie de dimensions et calcule le total g´en´eral selon

l’ordre de gauche `a droite dans la clause GROUP BY

S’il y a n colonnes de regroupements, GROUP BY ROLLUP g´en`ere n+1 niveaux de totalisation

ROLLUP (a, b, c)

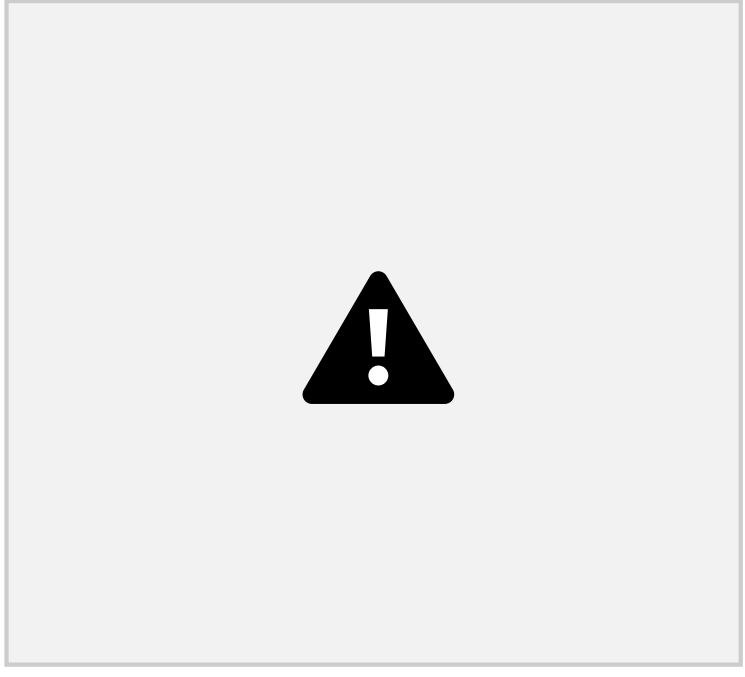
(a, b, c)
(a, b)
(a)
()

Extraction de donn´ees

Les sous int´errogations
Les jointures
Les op´erateurs ensemblistes

Les fonctions mono-ligne Les fonctions analytiques

```
SELECT Department_id, JOB_id,manager_id, SUM (SALARY)
FROM EMPLOYEES
WHERE DEPARTMENT_ID in (10,20,30)
GROUP BY Rollup(Department_id, JOB_id,manager_id);
```



Extraction de donn´ees

Les op´erateurs ensemblistes

Les sous int´errogations
Les jointures

L’op´erateur CUBE : calcule des agr´egats (SUM, COUNT, MAX, MIN, AVG) `a diff´erents niveaux d’agr´egation comme ROLLUP mais de plus permet de calculer toutes les combinaisons d’agr´egations :

L’op´erateur CUBE: calcule des sous-totaux pour toutes les combinaisons possibles d’un ensemble de colonnes de regroupement

Si la clause CUBE contient n colonnes, CUBE calcule 2^n

combinaisons de totaux CUBE (a, b, c)

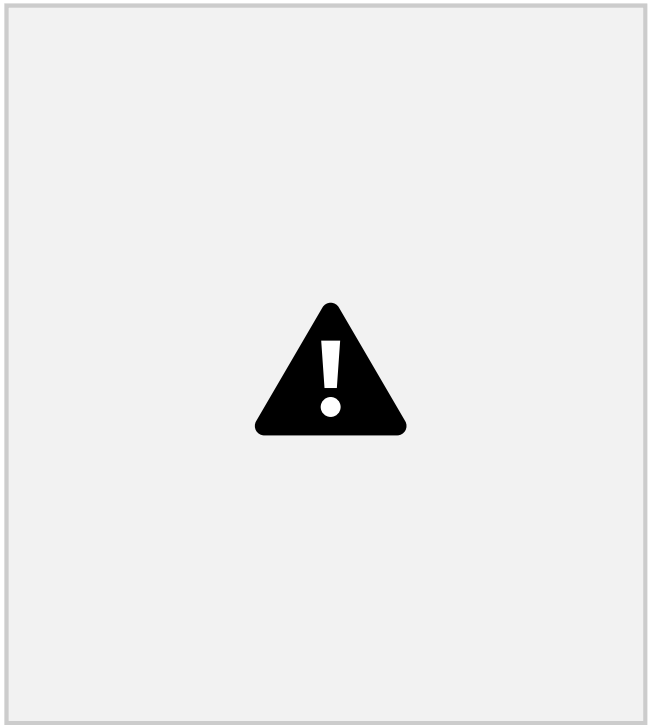
(a, b, c)
(a, b)
(a, c)
(a)
(b, c)

(b)
(c)
()

6490
Extraction de donn´ees

Les sous int´errogations
Les jointures
Les op´erateurs ensemblistes

```
SELECT Department_id, JOB_id, SUM (SALARY) FROM
EMPLOYEES
WHERE DEPARTMENT_ID in (10,20,30)
GROUP BY Cube(Department_id, JOB_id);
```



Les fonctions mono-ligne
Les fonctions analytiques

Extraction de donn´ees

Les sous int´errogations

Les jointures

Les op´erateurs ensemblistes

Les lignes de totaux correspondent g´enaralement aux lignes ayant des valeurs NULL

=>possibilit´e de confusion si les lignes contiennent d´eja des valeurs NULL !

La fonction GROUPING permet d'´eliminer cette ambiguit´e. Elle accepte une seule colonne comme param`etre et retourne:

1 si la colonne contient une valeur null g´en´er´e dans le cadre d'un sous-total par un ROLLUP ou CUBE

0 pour une autre valeur, y compris les valeurs NULL

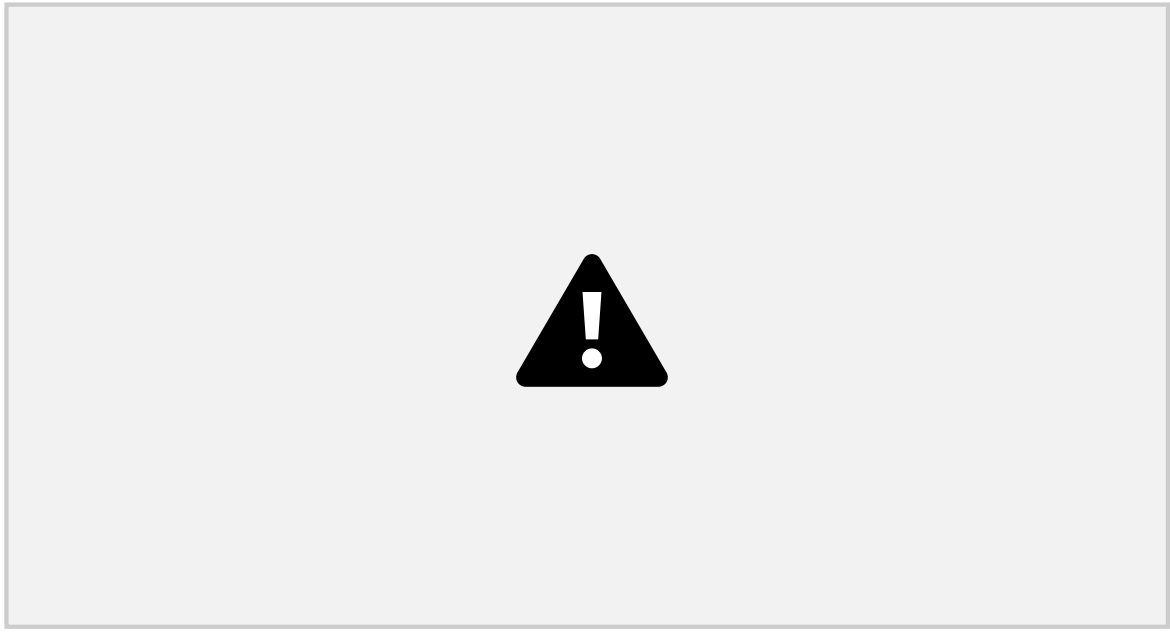
stock´ees 6690

Extraction de donn´ees

Les sous int´errogations
Les jointures
Les op´erateurs ensemblistes

```
SELECT Department_id, JOB_id, SUM
(SALARY), GROUPING(Department
id), GROUPING( JOB_id) FROM
EMPLOYEES
WHERE DEPARTMENT_ID in
(10,20,30)
GROUP BY Cube(Department_id, JOB
id);
```

Les fonctions mono-ligne Les fonctions analytiques



6790

Les sous-int´errogations monoligne
Les sous-int´errogations multi-lignes

Extraction de donn´ees

Les fonctions

Les sous int´errogations
Les sous-int´errogations monoligne Les sous-int´errogations multi-lignes
Les jointures

Les op´erateurs ensemblistes

6890

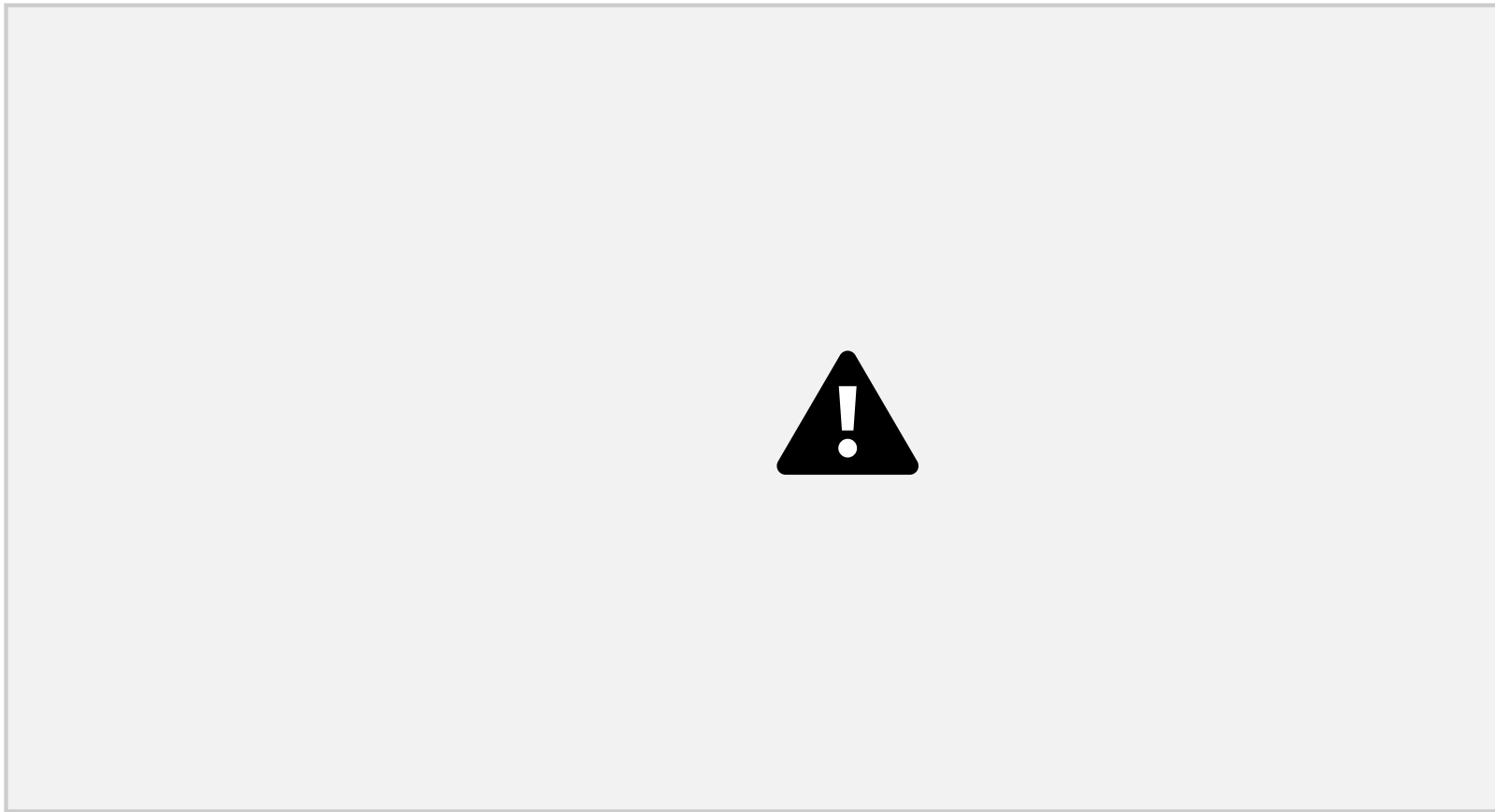
Extraction de donn´ees

Les fonctions

Les sous-int´errogations monoligne Les sous-int´errogations multi-lignes

Les jointures

Les op´erateurs ensemblistes



6990

Extraction de donn´ees

Les fonctions

Les jointures
Les op´erateurs ensemblistes

SELECT <colonne1>[, <colonne2>,..., <colonneN>]
FROM <nom_table>
WHERE <expression> OPERATEUR (SELECT SELECT <colonne1>[, <colonne2>,..., <colonneN>] FROM <nom_table>)

Mettre les sous-interrogations entre parenth`eses

La clause order by de la sous-interrogation n'est pas n´ecessaire

Utilisez des op´erateurs de comparaison monolignes avec les sous-interrogations monolignes, et des op´erateurs de comparaison multilignes avec les sous interrogations multilignes

op´erateurs mono-ligne (>,>= ,<, <=, . . .)
op´erateurs multi-lignes (IN, ALL, ANY)

La sous-interrogation (requˆete interne) est ex´ecut´ee une seule fois avant la requˆete principale Le r´esultat de la sous-interrogation est utilis´e par la requˆete principale (requˆete externe) Une sous-interrogation est utilis´ee dans les clauses suivantes :

WHERE
HAVING
FROM

Renvoient une seule ligne

Utilisent des operateurs de comparaison monolignes (= , >, >= , < , <= , <>)

```
SELECT last_name, job_id, salary
FROM employees
WHERE job_id =( SELECT job_id FROM employees WHERE
employee_id=124 );
```


```
SELECT last_name, job_id, salary
FROM employees
WHERE salary =(SELECT max(salary) FROM employees )
```

```
SELECT department_id, min(salary)
FROM employees
GROUP BY department_id
HAVING min(salary) > (SELECT MIN(salary) FROM employees
WHERE department_id= 20)
```

Renvoient plusieurs lignes

Utilisent des op´erateurs de comparaison multiligne (IN, ANY, ALL)

```
SELECT employee_id, last_name, job_id, salary
FROM employees
WHERE salary > ANY (SELECT salary FROM employees WHERE job
id = 'SA_MAN') AND job_id <> 'SA_MAN';
```

```
SELECT employee_id, last_name, job_id, salary
FROM employees
WHERE salary > ALL (SELECT salary FROM employees WHERE job
id = 'SA_MAN') AND job_id <> 'SA_MAN';
```

Jointure interne
Jointure externe
Equijointure / Non-´equijointure
Auto-jointure
Jointure naturelle
Produit cart´esien

Les jointures

Jointure interne
Jointure externe
Equijointure / Non-´equijointure
Auto-jointure
Jointure naturelle
Produit cart´esien

Jointure interne
Jointure externe
Equijointure / Non-´equijointure
Auto-jointure
Jointure naturelle
Produit cart´esien

Extraction de donn´ees Les fonctions
Les sous int´errogations
Les op´erateurs ensemblistes

Une jointure permet d'extraire des donn´ees `a partir de plusieurs tables (et / ou vues) en utilisant des conditions de jointure

La condition de jointure peut ˆetre exprim´ee:

dans la clause WHERE: WHERE table1.C1=table2.C1
dans la clause ON: ON table1.C1=table2.C1

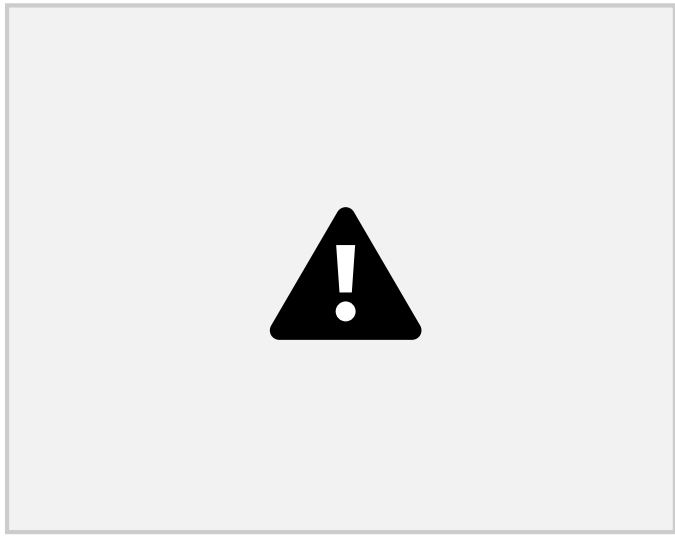
Pr´ec´eder le nom de la colonne par le nom de la table lorsque nom de la colonne figure dans plusieurs tables

Il existe deu types de jointure:

jointure interne

jointure externe

Dans ce qui suit on utilisera les tables tab1 et tab2 suivantes pour les exemples:



7690

Jointure externe
Equijointure / Non-´equijointure
Auto-jointure
Jointure naturelle
Produit cart´esien

Extraction de donn´ees Les fonctions
Les sous int´errogations
Les op´erateurs ensemblistes

Une jointure interne (appel´e aussi jointure simple) est une jointure de deux tables ou plus qui retourne uniquement les lignes qui satisfont la condition de jointure

```
SELECT T1.colonne1, . . . ,T1.colonneN,T2.colonne1, . . .
,T2.colonneM
FROM T1 [ INNER ] JOIN T2
ON <condition_jointure>
WHERE <condition>
```

```
SELECT tab1.a, tab1.b, tab2.a,tab2.b
FROM tab1 INNER JOIN tab2
ON tab1.a=tab2.a and tab1.b=tab2.b;
```

7790

Jointure interne

Equijointure / Non-´equijointure
Auto-jointure
Jointure naturelle
Produit cart´esien

Extraction de donn´ees Les fonctions
Les sous int´errogations
Les op´erateurs ensemblistes

Une jointure externe ´etend le r´esultat d'une jointure interne

Une jointure externe renvoie toutes les lignes qui satisfont la condition de jointure et renvoie ´egalement une partie ou l'ensemble des lignes d'une table pour lesquelles aucune ligne de l'autre table satisfait la condition de jointure. Il existe 3 types de jointures externe:

jointure externe gauche: jointure entre A et B => afficher les lignes de A qui ne satisfont pas la condition de jointure
jointure externe droite: jointure entre A et B => afficher les lignes de B qui ne satisfont pas la condition de jointure
jointure externe compl`ete: jointure entre A et B => afficher les lignes de A et B qui ne satisfont pas la condition de jointure

SELECT T1.colonne1, . . . ,T1.colonneN,T2.colonne1, . . .
,T2.colonneM
FROM T1 {LEFT | RIGHT | FULL} [OUTER] JOIN T2
ON <condition_jointure>
WHERE <condition>

Exemple 3:

```
tab2.a,tab2.b
 FROM tab1 LEFT OUTER JOIN
tab2
ON tab1.a=tab2.a and
```

Exemple 2:

```
SELECT tab1.a, tab1.b,
tab2.a,tab2.b
FROM tab1 RIGHT OUTER JOIN
tab2
ON tab1.a=tab2.a and
  SELECT tab1.a, tab1.b,
 tab2.a,tab2.b
  FROM tab1 FULL OUTER JOIN
 tab2
  ON tab1.a=tab2.a and
 tab1.b=tab2.b;
```

Exemple 1:

```
SELECT tab1.a, tab1.b,
```



tab1.b=tab2.b;
tab1.b=tab2.b;

Extraction de donn´ees
Les fonctions
Les sous int´errogations
Les op´erateurs ensemblistes
Jointure interne
Jointure externe
Auto-jointure
Jointure naturelle
Produit cart´esien

ON tab1.a=tab2.a;

Une ´equijointure est une jointure avec une condition de jointure contenant un op´erateur d'´egalit´e (= , LIKE , etc)

Exemple 1:

```
SELECT tab1.a, tab1.b, tab2.a,tab2.b
FROM tab1 LEFT OUTER JOIN tab2
```

Une non-´equijointure est une jointure avec une condition de jointure contenant un op´erateur d'in´egalit´e (< ,<= , > , >= , BETWEEN , etc)

Exemple 2:

```
SELECT tab1.a, tab2.a, tab2.c
FROM tab1 INNER JOIN tab2 $
ON tab1.a>=tab2.a and tab1.a<=tab2.c;
```

8090