

2

Limiter et trier des données

Objectifs

A la fin de ce chapitre, vous pourrez :

- **limiter le nombre de lignes extraites par une interrogation**
- **trier les lignes extraites par une interrogation**

Limiter le nombre de lignes à l'aide d'une sélection


EMPLOYEES

EMPLOYEE_ID	LAST_NAME	JOB_ID	DEPARTMENT_ID
100	King	AD_PRES	90
101	Kochhar	AD_VP	90
102	De Haan	AD_VP	90
103	Hunold	IT_PROG	60
104	Ernst	IT_PROG	60
107	Lorentz	IT_PROG	60
124	Mourgos	ST_MAN	50

...

20 rows selected.

"Extraire tous les employés du service 90"



EMPLOYEE_ID	LAST_NAME	JOB_ID	DEPARTMENT_ID
100	King	AD_PRES	90
101	Kochhar	AD_VP	90
102	De Haan	AD_VP	90

Limiter le nombre de lignes sélectionnées

- Limitez le nombre de lignes renvoyées à l'aide de la clause WHERE.

```
SELECT      * | {[DISTINCT] column | expression [alias], ...}  
FROM        table  
[WHERE      condition(s)];
```

- La clause WHERE se place après la clause FROM.

Utiliser la clause WHERE

```
SELECT employee_id, last_name, job_id, department_id
FROM   employees
WHERE  department_id = 90 ;
```

EMPLOYEE_ID	LAST_NAME	JOB_ID	DEPARTMENT_ID
100	King	AD_PRES	90
101	Kochhar	AD_VP	90
102	De Haan	AD_VP	90

Chaînes de caractères et dates

- Les chaînes de caractères et les dates doivent être placées entre apostrophes.
- La recherche tient compte des majuscules/minuscules pour les chaînes de caractères et du format pour les dates.
- Le format de date par défaut est DD-MON-RR.

```
SELECT last_name, job_id, department_id
FROM   employees
WHERE  last_name = 'Whalen';
```

Conditions de comparaison

Opérateur	Signification
=	Egal à
>	Supérieur à
>=	Supérieur ou égal à
<	Inférieur à
<=	Inférieur ou égal à
<>	Différent de

Utiliser des conditions de comparaison

```
SELECT last_name, salary
FROM employees
WHERE salary <= 3000;
```

LAST_NAME	SALARY
Matos	2600
Vargas	2500

Autres conditions de comparaison

Opérateur	Signification
BETWEEN ...AND...	Compris entre ... et ... (bornes comprises)
IN(set)	Correspond à une valeur de la liste
LIKE	Ressemblance partielle de chaînes de caractères
IS NULL	Correspond à une valeur NULL

Utiliser la condition BETWEEN

Utilisez la condition BETWEEN pour afficher des lignes en fonction d'une plage de valeurs.

```
SELECT last_name, salary
FROM employees
WHERE salary BETWEEN 2500 AND 3500;
```

↑
Limite inférieure

↑
Limite supérieure

LAST_NAME	SALARY
Rajs	3500
Davies	3100
Matos	2600
Vargas	2500

Utiliser la condition IN

Utilisez la condition d'appartenance IN pour vérifier la présence de valeurs dans une liste.

```
SELECT employee_id, last_name, salary, manager_id
FROM   employees
WHERE  manager_id IN (100, 101, 201);
```

EMPLOYEE_ID	LAST_NAME	SALARY	MANAGER_ID
202	Fay	6000	201
200	Whalen	4400	101
205	Higgins	12000	101
101	Kochhar	17000	100
102	De Haan	17000	100
124	Mourgos	5800	100
149	Zlotkey	10500	100
201	Hartstein	13000	100

8 rows selected.

Utiliser la condition LIKE

- Utilisez la condition LIKE pour rechercher des chaînes de caractères valides à l'aide de caractères génériques.
- Les conditions de recherche peuvent contenir des caractères ou des nombres littéraux :
 - % représente zéro ou plusieurs caractères.
 - _ représente un caractère.

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

Utiliser la condition LIKE

- Vous pouvez combiner plusieurs caractères génériques de recherche.

```
SELECT last_name  
FROM employees  
WHERE last_name LIKE '_o%';
```

LAST_NAME
Kochhar
Lorentz
Mourgos

- Vous pouvez utiliser l'identificateur ESCAPE pour rechercher les symboles % et _.

Utiliser les conditions NULL

Recherchez des valeurs NULL avec l'opérateur
IS NULL.

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

LAST_NAME	MANAGER_ID
King	

Conditions logiques

Opérateur	Signification
AND	Renvoie TRUE si les <i>deux</i> conditions sont vraies
OR	Renvoie TRUE si <i>l'une</i> des conditions est vraie
NOT	Renvoie la valeur TRUE si la condition qui suit l'opérateur est fausse

Utiliser l'opérateur AND

L'opérateur AND exige que les deux conditions soient vraies.

```
SELECT employee_id, last_name, job_id, salary
FROM   employees
WHERE  salary >=10000
AND    job_id LIKE '%MAN%';
```

EMPLOYEE_ID	LAST_NAME	JOB_ID	SALARY
149	Zlotkey	SA_MAN	10500
201	Hartstein	MK_MAN	13000

Utiliser l'opérateur OR

L'opérateur OR exige que l'une des conditions soit vraie.

```
SELECT employee_id, last_name, job_id, salary
FROM   employees
WHERE  salary >= 10000
OR     job_id LIKE '%MAN%';
```

EMPLOYEE_ID	LAST_NAME	JOB_ID	SALARY
100	King	AD_PRES	24000
101	Kochhar	AD_VP	17000
102	De Haan	AD_VP	17000
124	Mourgos	ST_MAN	5800
149	Zlotkey	SA_MAN	10500
174	Abel	SA_REP	11000
201	Hartstein	MK_MAN	13000
205	Higgins	AC_MGR	12000

8 rows selected.

Utiliser l'opérateur NOT

```
SELECT last_name, job_id
FROM employees
WHERE job_id
      NOT IN ('IT_PROG', 'ST_CLERK', 'SA_REP');
```

LAST_NAME	JOB_ID
King	AD_PRES
Kochhar	AD_VP
De Haan	AD_VP
Mourgos	ST_MAN
Zlotkey	SA_MAN
Whalen	AD_ASST
Hartstein	MK_MAN
Fay	MK_REP
Higgins	AC_MGR
Gietz	AC_ACCOUNT

10 rows selected.

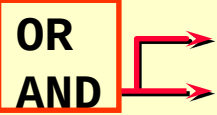
Règles de priorité

Ordre d'évaluation	Opérateur
1	Opérateurs arithmétiques
2	Opérateur de concaténation
3	Conditions de comparaison
4	IS [NOT] NULL, LIKE, [NOT] IN
5	[NOT] BETWEEN
6	Condition logique NOT
7	Condition logique AND
8	Condition logique OR

Les parenthèses permettent de modifier les règles de priorité.

Règles de priorité

```
SELECT last_name, job_id, salary
FROM   employees
WHERE  job_id = 'SA_REP'
OR     job_id = 'AD_PRES'
AND    salary > 15000;
```



LAST_NAME	JOB_ID	SALARY
King	AD_PRES	24000
Abel	SA_REP	11000
Taylor	SA_REP	8600
Grant	SA_REP	7000

Règles de priorité

Utilisez des parenthèses pour forcer la priorité.

```
SELECT last_name, job_id, salary
FROM employees
WHERE (job_id = 'SA_REP'
OR job_id = 'AD_PRES')
AND salary > 15000;
```

LAST_NAME	JOB_ID	SALARY
King	AD_PRES	24000

Clause ORDER BY

- Triez des lignes à l'aide de la clause **ORDER BY**.
 - **ASC** : ordre croissant (par défaut)
 - **DESC** : ordre décroissant
- La clause **ORDER BY** se place à la fin de l'instruction **SELECT**.

```
SELECT last_name, job_id, department_id, hire_date
FROM employees
ORDER BY hire_date ;
```

LAST_NAME	JOB_ID	DEPARTMENT_ID	HIRE_DATE
King	AD_PRES	90	17-JUN-87
Whalen	AD_ASST	10	17-SEP-87
Kochhar	AD_VP	90	21-SEP-89
Hunold	IT_PROG	60	03-JAN-90
Ernst	IT_PROG	60	21-MAY-91

...

20 rows selected.

Trier par ordre décroissant

```
SELECT last_name, job_id, department_id, hire_date
FROM employees
ORDER BY hire_date DESC ;
```

LAST_NAME	JOB_ID	DEPARTMENT_ID	HIRE_DATE
Zlotkey	SA_MAN	80	29-JAN-00
Mourgos	ST_MAN	50	16-NOV-99
Grant	SA_REP		24-MAY-99
Lorentz	IT_PROG	60	07-FEB-99
Vargas	ST_CLERK	50	09-JUL-98
Taylor	SA_REP	80	24-MAR-98
Matos	ST_CLERK	50	15-MAR-98
Fay	MK_REP	20	17-AUG-97
Davies	ST_CLERK	50	29-JAN-97

...

20 rows selected.

Trier par alias de colonne

```
SELECT employee_id, last_name, salary*12 annsal  
FROM employees  
ORDER BY annsal;
```

EMPLOYEE_ID	LAST_NAME	ANNSAL
144	Vargas	30000
143	Matos	31200
142	Davies	37200
141	Rajs	42000
107	Lorentz	50400
200	Whalen	52800
124	Mourgos	69600
104	Ernst	72000
202	Fay	72000
178	Grant	84000

...

20 rows selected.

Trier sur plusieurs colonnes

- L'ordre des éléments de la liste ORDER BY donne l'ordre du tri.

```
SELECT last_name, department_id, salary
FROM employees
ORDER BY department_id, salary DESC;
```

LAST_NAME	DEPARTMENT_ID	SALARY
Whalen	10	4400
Hartstein	20	13000
Fay	20	6000
Mourgos	50	5800
Rajs	50	3500
Davies	50	3100
Matos	50	2600
Vargas	50	2500

...

20 rows selected.

- Vous pouvez effectuer un tri sur une colonne ne figurant pas dans la liste SELECT.

Synthèse

Ce chapitre vous a permis d'apprendre à :

- utiliser la clause **WHERE** pour limiter le nombre de lignes de résultat
 - utiliser les conditions de comparaison
 - utiliser les conditions **BETWEEN**, **IN**, **LIKE** et **NULL**
 - appliquer les opérateurs logiques **AND**, **OR** et **NOT**
- utiliser la clause **ORDER BY** pour trier les lignes de résultat

```
SELECT      * | {[DISTINCT] column | expression [alias], ...}  
FROM        table  
[WHERE      condition(s)]  
[ORDER BY   {column, expr, alias} [ASC | DESC]];
```

Présentation de l'exercice 2

Dans cet exercice, vous allez :

- **sélectionner des données et modifier l'ordre d'affichage des lignes,**
- **limiter le nombre de lignes à l'aide de la clause WHERE,**
- **trier des lignes à l'aide de la clause ORDER BY.**

