





# Restreindre et trier les données



# Limiter les lignes par sélection





## EMPLOYEES

	 EMPLOYEE_ID	 LAST_NAME	 JOB_ID	 DEPARTMENT_ID
1	200	Whalen	AD_ASST	10
2	201	Hartstein	MK_MAN	20
3	202	Fay	MK_REP	20
4	205	Higgins	AC_MGR	110
5	206	Gietz	AC_ACCOUNT	110

...

**"extraire tous  
les employés  
du département 90"**



	 EMPLOYEE_ID	 LAST_NAME	 JOB_ID	 DEPARTMENT_ID
1	100	King	AD_PRES	90
2	101	Kochhar	AD_VP	90
3	102	De Haan	AD_VP	90

# Limiter les lignes sélectionnées

- Limitez les lignes renvoyées à l'aide de la clause WHERE :

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

- La clause WHERE suit 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
1	100	King	AD_PRES	90
2	101	Kochhar	AD_VP	90
3	102	De Haan	AD_VP	90

# Chaînes de caractères et dates

- Les chaînes de caractères et les dates sont indiquées entre apostrophes.
- Dans les chaînes de caractères, les majuscules sont distinguées des minuscules. Les dates doivent respecter un format spécifique.
- Le format par défaut d'affichage de la date est DD-MON-RR.

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

```
SELECT last_name  
FROM employees  
WHERE hire_date = '17-FEB-96' ;
```

# Opérateurs de comparaison

Opérateur	Signification
=	Egal à
>	Supérieur à
>=	Supérieur ou égal à
<	Inférieur à
<=	Inférieur ou égal à
<>	Non égal à
BETWEEN ...AND...	Compris entre deux valeurs (incluses)
IN (set)	Correspond à une valeur
LIKE	Correspond à un modèle
IS NULL	Est une valeur NULL

# Utiliser des opérateurs de comparaison

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

	LAST_NAME	SALARY
1	Matos	2600
2	Vargas	2500

# Indication d'une plage à l'aide de l'opérateur BETWEEN

- Utilisez l'opérateur BETWEEN pour afficher des lignes en fonction d'une plage de valeurs :

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

	LAST_NAME	SALARY
1	Rajs	3500
2	Davies	3100
3	Matos	2600
4	Vargas	2500

↑  
**Limite  
inférieure**

↑  
**Limite  
supérieure**



# Condition d'appartenance à l'aide de l'opérateur IN

- Utilisez l'opérateur IN pour rechercher les valeurs d'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
1	201	Hartstein	13000	100
2	101	Kochhar	17000	100
3	102	De Haan	17000	100
4	124	Mourgos	5800	100
5	149	Zlotkey	10500	100
6	200	Whalen	4400	101
7	205	Higgins	12000	101
8	202	Fay	6000	201

# Correspondance avec un modèle à l'aide de l'opérateur LIKE

- Utilisez l'opérateur LIKE pour effectuer des recherches avec des caractères génériques.
- Les conditions de recherche peuvent contenir des caractères ou des nombres littéraux :
  - % indique zéro ou plusieurs caractères.
  - \_ indique un caractère.

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

# Combiner des caractères génériques

- Vous pouvez combiner les deux caractères génériques (% , \_ ) avec des caractères littéraux pour définir le modèle à rechercher :

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

	LAST_NAME
1	Kochhar
2	Lorentz
3	Mourgos

- Vous pouvez utiliser l'identificateur ESCAPE pour effectuer une recherche sur les caractères % et \_.

# Utiliser les conditions NULL

- Recherchez les valeurs NULL avec l'opérateur IS NULL.

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

	LAST_NAME	MANAGER_ID
1	King	(null)

# Définir des conditions à l'aide des opérateurs logiques

Opérateur	Signification
AND	Renvoie <code>TRUE</code> si <i>les deux</i> conditions sont vraies
OR	Renvoie <code>TRUE</code> si <i>l'une ou l'autre</i> des conditions est vraie
NOT	Renvoie <code>TRUE</code> si la condition est fausse

# Utiliser l'opérateur AND

- AND requiert 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
1	201	Hartstein	MK_MAN	13000
2	149	Zlotkey	SA_MAN	10500

# Utiliser l'opérateur OR

- OR requiert que l'une ou l'autre 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
1	201	Hartstein	MK_MAN	13000
2	205	Higgins	AC_MGR	12000
3	100	King	AD_PRES	24000
4	101	Kochhar	AD_VP	17000
5	102	De Haan	AD_VP	17000
6	124	Mourgos	ST_MAN	5800
7	149	Zlotkey	SA_MAN	10500
8	174	Abel	SA_REP	11000

# 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
1	De Haan	AD_VP
2	Fay	MK_REP
3	Gietz	AC_ACCOUNT
4	Hartstein	MK_MAN
5	Higgins	AC_MGR
6	King	AD_PRES
7	Kochhar	AD_VP
8	Mourgos	ST_MAN
9	Whalen	AD_ASST
10	Zlotkey	SA_MAN



# Règles de priorité




Opérateur	Signification
1	Opérateurs arithmétiques
2	Opérateur de concaténation
3	Conditions de comparaison
4	IS [NOT] NULL, LIKE, [NOT] IN
5	[NOT] BETWEEN
6	Non égal à
7	Condition logique NOT
8	Condition logique AND
9	Condition logique OR

Vous pouvez utiliser des parenthèses pour 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;
```

1

	 LAST_NAME	 JOB_ID	 SALARY
1	King	AD_PRES	24000
2	Abel	SA_REP	11000
3	Taylor	SA_REP	8600
4	Grant	SA_REP	7000

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

2

	 LAST_NAME	 JOB_ID	 SALARY
1	King	AD_PRES	24000

# Utiliser la clause ORDER BY

- Trier les lignes extraites avec la clause ORDER BY :
  - ASC: ordre croissant (par défaut)
  - DESC: ordre décroissant
- La clause ORDER BY est le dernier élément 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
1	King	AD_PRES	90	17-JUN-87
2	Whalen	AD_ASST	10	17-SEP-87
3	Kochhar	AD_VP	90	21-SEP-89
4	Hunold	IT_PROG	60	03-JAN-90
5	Ernst	IT_PROG	60	21-MAY-91
6	De Haan	AD_VP	90	13-JAN-93

...

# Tri

- Tri par ordre décroissant :

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

1

- Tri par alias de colonne :

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

2

# Tri

- Tri selon une colonne repérée par sa position numérique :

```
SELECT last_name, job_id, department_id,  
hire_date  
FROM employees  
ORDER BY 3;
```

3

- Tri en fonction de plusieurs colonnes :

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

4

# Variables de substitution



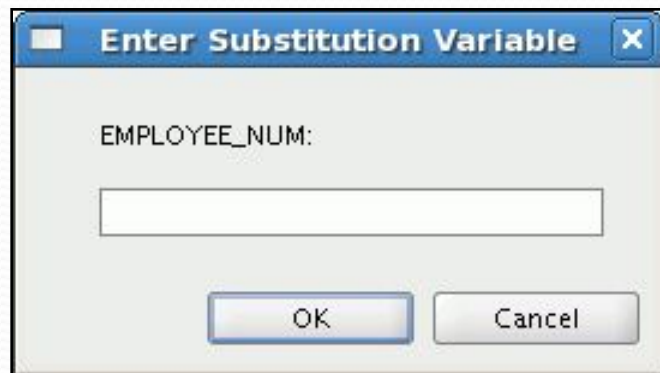
# Variables de substitution

- Utilisez des variables de substitution pour :
  - stocker temporairement des valeurs (variables de substitution avec esperluette d'interprétation simple (&) ou double (&&))
- Utilisez des variables de substitution pour remplacer :
  - des conditions WHERE
  - des clauses ORDER BY
  - des expressions de colonne
  - des noms de table
  - des instructions SELECT entières

# substitution avec esperluette simple

- Utilisez une variable comprenant une esperluette d'interprétation (&) comme préfixe pour inviter l'utilisateur à entrer une valeur :

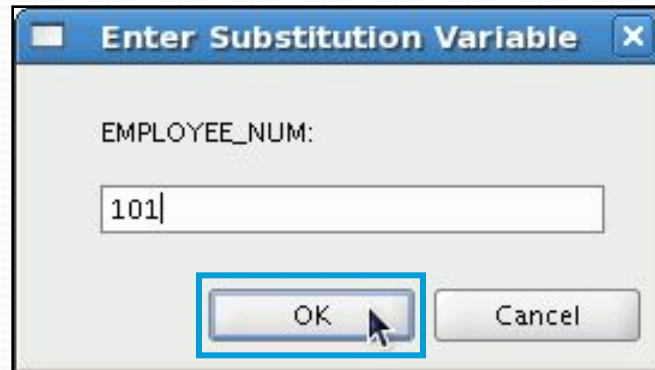
```
SELECT employee_id, last_name, salary,  
department_id  
FROM employees  
WHERE employee_id = &employee_num;
```



The screenshot shows a dialog box titled "Enter Substitution Variable" with a close button (X) in the top right corner. Inside the dialog, the text "EMPLOYEE\_NUM:" is displayed above a text input field. At the bottom of the dialog, there are two buttons: "OK" and "Cancel".



# Utiliser une variable de substitution avec esperluette simple



Enter Substitution Variable

EMPLOYEE\_NUM:

101

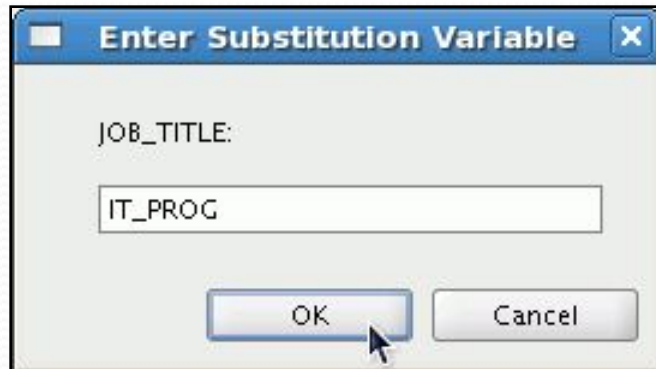
OK Cancel

	EMPLOYEE_ID	LAST_NAME	SALARY	DEPARTMENT_ID
1	101	Kochhar	17000	90

# Chaînes de caractères et dates avec variables de substitution

- Utilisez des apostrophes pour les dates et les chaînes de caractères :

```
SELECT last_name, department_id, salary*12  
FROM employees  
WHERE job_id = '&job_title' ;
```



A screenshot of a database client dialog box titled "Enter Substitution Variable". It contains a label "JOB\_TITLE:" and a text input field with the value "IT\_PROG". At the bottom are "OK" and "Cancel" buttons. A mouse cursor is pointing at the "OK" button.

	LAST_NAME	DEPARTMENT_ID	SALARY*12
1	Hunold	60	108000
2	Ernst	60	72000
3	Lorentz	60	50400

# Indiquer des noms de colonne, des expressions et du texte

```
SELECT employee_id, last_name, job_id, &column name  
FROM employees  
WHERE &condition  
ORDER BY &order column ;
```

**Enter Substitution Variable** [X]

COLUMN\_NAME:

salary

OK

**Enter Substitution Variable** [X]

CONDITION:

salary >15000

OK

**Enter Substitution Variable** [X]

ORDER\_COLUMN:

last\_name

OK Cancel

# Utiliser une variable de substitution avec esperluette double

- Utilisez l'esperluette d'interprétation double (&&) si vous souhaitez réutiliser la valeur de la variable sans solliciter l'utilisateur à chaque fois :

```
SELECT employee_id, last_name, job_id,   
&&column_name  
FROM   
ORDER BY &&column_name;
```

☐ Enter Substitution Variable

COLUMN\_NAME:

	EMPLOYEE_ID	LAST_NAME	JOB_ID	DEPARTMENT_ID
1	200	Whalen	AD_ASST	10
2	201	Hartstein	MK_MAN	20
3	202	Fay	MK_REP	20

...

# Utiliser la commande DEFINE

- Utilisez la commande DEFINE pour créer et affecter une valeur à une variable.
- Utilisez la commande UNDEFINE pour supprimer une variable.

```
DEFINE employee_num = 200  
  
SELECT employee_id, last_name, salary,  
department_id  
FROM employees  
WHERE employee_id = &employee_num ;  
  
UNDEFINE employee_num
```

# Utiliser la commande VERIFY

- Utilisez la commande VERIFY pour basculer de l'affichage d'une variable de substitution avant son remplacement par une valeur à l'affichage de cette variable après son remplacement :

```
SET VERIFY ON
```

```
SELECT employee_id, last_name, salary  
FROM   employees  
WHERE  employee_id = &employee_num;
```

Enter Substitution Variable

EMPLOYEE\_NUM:

200

OK Cancel

Results Script Output Explain Autotrace DBMS Output OWA Output

```
SELECT employee_id, last_name, salary  
FROM   employees  
WHERE  employee_id = 200
```

EMPLOYEE_ID	LAST_NAME	SALARY
200	Whalen	4400

1 rows selected