COURS BASE DE DONNÉES

CHAPITRE 4
3. LE LANGAGE SQL: RESTREINDRE ET TRIER LES DONNÉES

Dr. Sondès Fayech

PLAN

- Limiter les lignes avec:
  - La clause WHERE
  - Les conditions de comparaison utilisant =, <=, BETWEEN, IN, LIKE et NULL
  - Conditions logiques utilisant les opérateurs AND, OR et NOT
- Règles de priorité pour les opérateurs dans une expression
- Tri des lignes à l'aide de la clause ORDER BY
- Variables de substitution
- Commandes DEFINE et VERIFY

Limitation des lignes à l'aide d'une sélection

EMPLOYEES

	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
4	103	Hunold	IT_PROG	60
5	104	Ernst	IT_PROG	60
6	107	Lorentz	IT_PROG	60

...

“récupérer 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

Limitation des lignes sélectionnées

- Restreindre 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.

Utilisation de 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 valeurs de date sont entourées de guillemets simples.
- Les valeurs de caractères sont sensibles à la case et les valeurs de date sont sensibles au format.
- Le format d'affichage de la date par défaut est DD-MON-RR.

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

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

Opérateurs de comparaison

Opérateur	Sens
=	Égal à
>	Plus grand que
>=	Plus grand ou égal à
<	Inférieur que
<=	Inférieur ou égal à
<>	Pas égal à
BETWEEN ...AND...	Entre deux valeurs (incluse)
IN(set)	Correspondre à un élément d'une liste
LIKE	Correspondre à un modèle de caractères
IS NULL	Est une valeur nulle

Utilisation d'opérateurs de comparaison


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

	LAST_NAME	SALARY
1	Matos	2600
2	Vargas	2500

Conditions à l'aide de l'opérateur BETWEEN

- Utilisez l'opérateur BETWEEN pour afficher les lignes en fonction d'un intervalle de valeurs:

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

Limite inférieure **Limite supérieure**

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

Conditions à l'aide de l'opérateur IN

- Utilisez l'opérateur IN pour tester les valeurs dans une liste:

```
SELECT employee_id, last_name, salary, manager_id
FROM   employees
WHERE  manager_id                          ;
```

	EMPLOYEE_ID	LAST_NAME	SALARY	MANAGER_ID
1	101	Kochhar	17000	100
2	102	De Haan	17000	100
3	124	Mourgos	5800	100
4	149	Zlotkey	10500	100
5	201	Hartstein	13000	100
6	200	Whalen	4400	101
7	205	Higgins	12000	101
8	202	Fay	6000	201

Correspondance de modèle à l'aide de l'opérateur LIKE

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

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

Combinaison de caractères génériques

- Vous pouvez combiner les deux caractères génériques (%, _) avec des caractères littéraux pour la correspondance de modèle:

```
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 rechercher les symboles% et _ réels.

Utilisation des conditions NULL

- Testez 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)

Définition de conditions à l'aide des opérateurs logiques

Opérateur	Sens
`AND`	Renvoie TRUE si les deux conditions sont vraies
`OR`	Renvoie TRUE si l'une des conditions est vraie
`NOT`	Renvoie TRUE si la condition est fausse

Utilisation de l'opérateur OR

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

Utilisation de 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

PLAN

- Limiter les lignes avec:
  - La clause WHERE
  - Les conditions de comparaison utilisant =, <=, BETWEEN, IN, LIKE et NULL
  - Conditions logiques utilisant les opérateurs AND, OR et NOT
- Règles de priorité pour les opérateurs dans une expression
- Tri des lignes à l'aide de la clause ORDER BY
- Variables de substitution
- Commandes DEFINE et VERIFY

Règles de priorité

Opérateur	Sens
1	Opérateurs arithmétique
2	Opérateurs de concatenation
3	Conditions de comparison
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 remplacer 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

PLAN

- Limiter les lignes avec:
  - La clause WHERE
  - Les conditions de comparaison utilisant =, <=, BETWEEN, IN, LIKE et NULL
  - Conditions logiques utilisant les opérateurs AND, OR et NOT
- Règles de priorité pour les opérateurs dans une expression
- Tri des lignes à l'aide de la clause ORDER BY
- Variables de substitution
- Commandes DEFINE et VERIFY

Utilisation de la clause ORDER BY

- Trier les lignes récupérées avec la clause ORDER BY:
  - ASC: Ordre croissant, par défaut
  - DESC: ordre décroissant
- La clause ORDER BY vient en dernier dans 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

...

Trier

- 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

Trier

- Tri en utilisant la position numérique de la colonne:

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

3

- Tri par plusieurs colonnes:

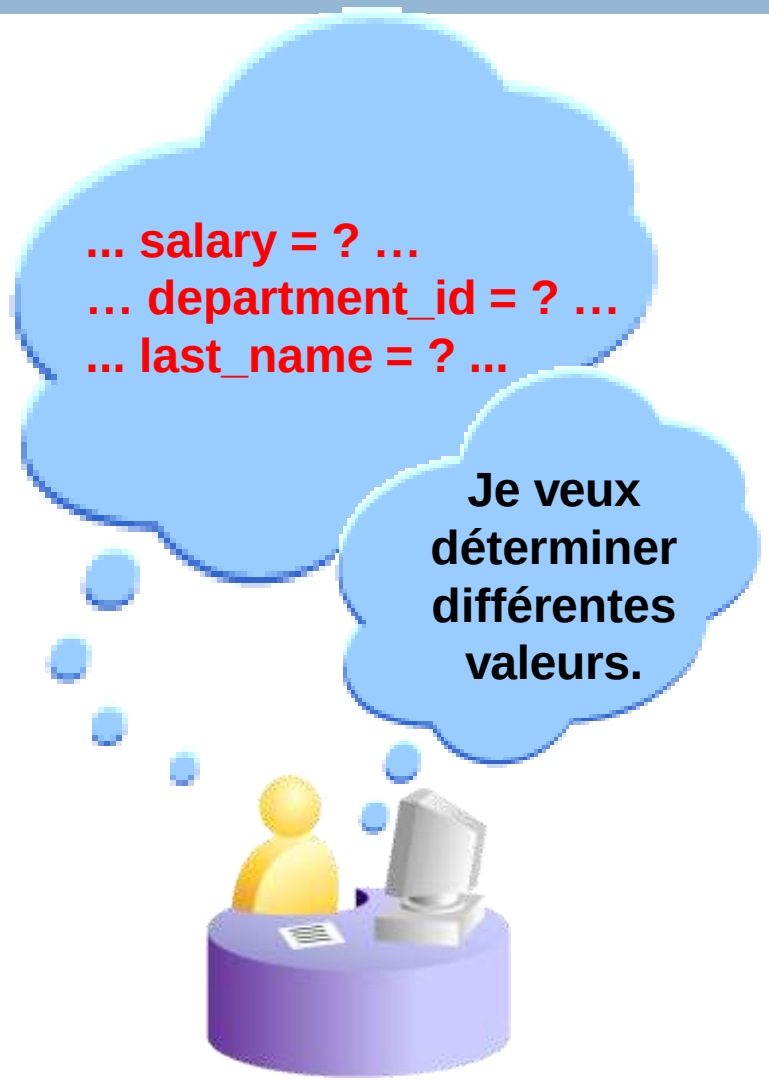
```
SELECT last_name, department_id, salary
FROM   employees
ORDER BY department_id, salary DESC;
```

4

PLAN

- Limiter les lignes avec:
  - La clause WHERE
  - Les conditions de comparaison utilisant =, <=, BETWEEN, IN, LIKE et NULL
  - Conditions logiques utilisant les opérateurs AND, OR et NOT
- Règles de priorité pour les opérateurs dans une expression
- Tri des lignes à l'aide de la clause ORDER BY
- Variables de substitution
- Commandes DEFINE et VERIFY

Variables de substitution



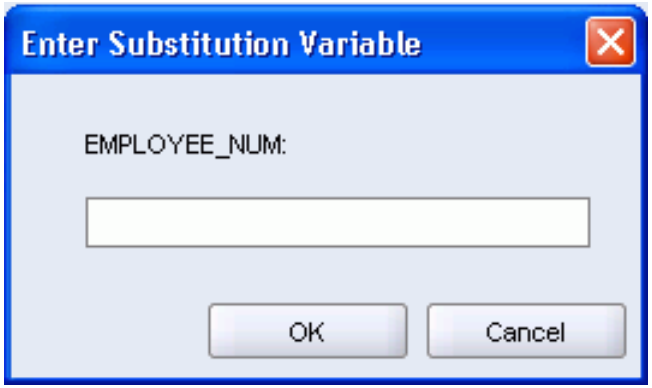
Variables de substitution

- Utilisez des variables de substitution pour:
  - Stocker temporairement des valeurs avec des substitutions single-ampersand (&) et double-ampersand (&&)
- Utilisez des variables de substitution pour compléter ce qui suit:
  - Les conditions WHERE
  - Clauses ORDER BY
  - Expressions de colonnes
  - Noms de tables
  - Instructions SELECT complètes

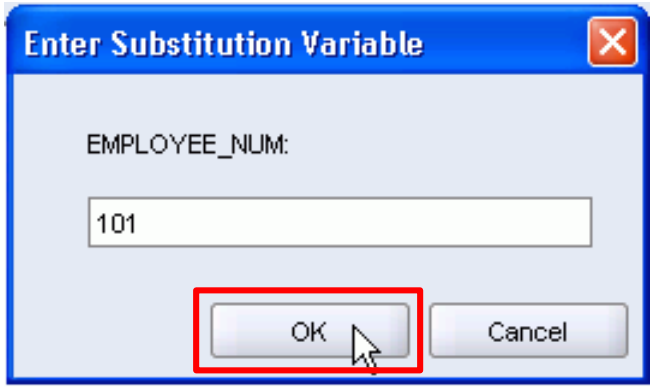
Utilisation de la variable de substitution Single-Ampersand (&)

- Utilisez une variable préfixée de (&) pour inviter l'utilisateur à saisir une valeur:

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



Utilisation de la variable de substitution Single-Ampersand (&)

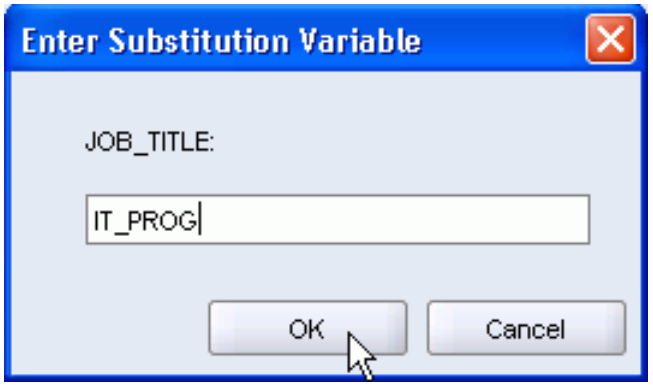


	EMPLOYEE_ID	LAST_NAME	SALARY	DEPARTMENT_ID
1	101	Kochhar	17000	90

Valeurs de caractères et de dates avec les variables de substitution

- Utilisez des guillemets simples pour les valeurs de dates et de caractères:

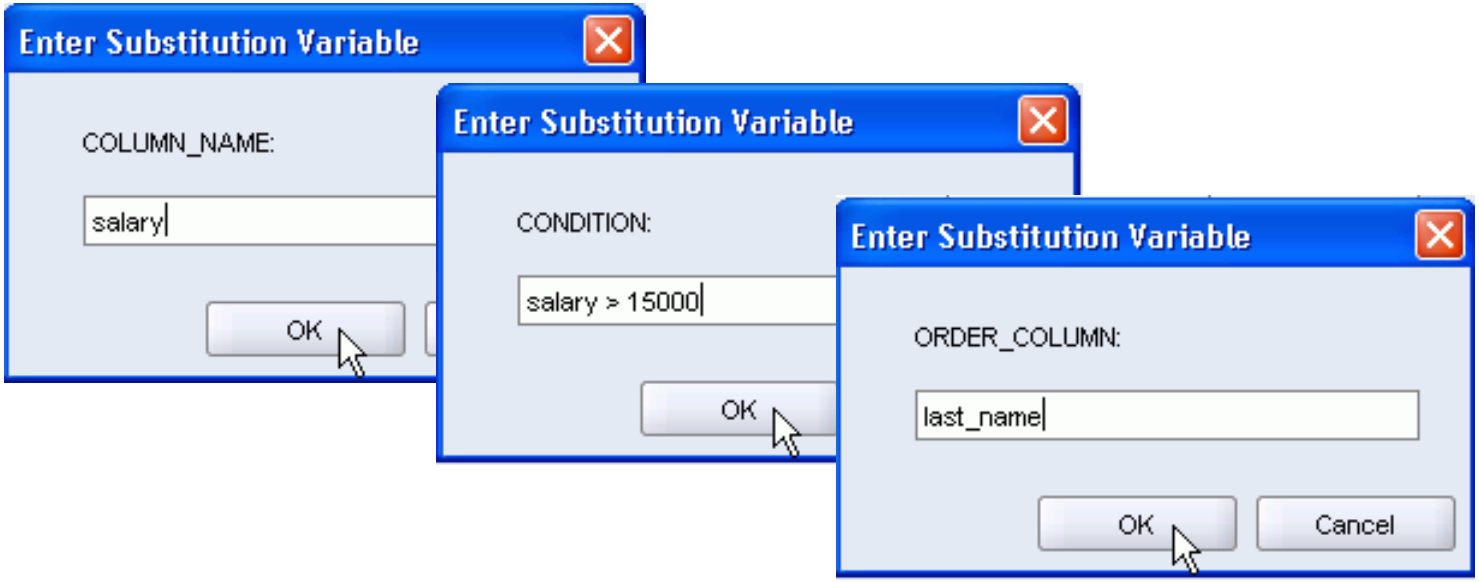
```
SELECT last_name, department_id, salary*12
FROM   employees
WHERE  job_id = '&job_title' ;
```



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

Spécification des noms de colonnes, des expressions et du texte

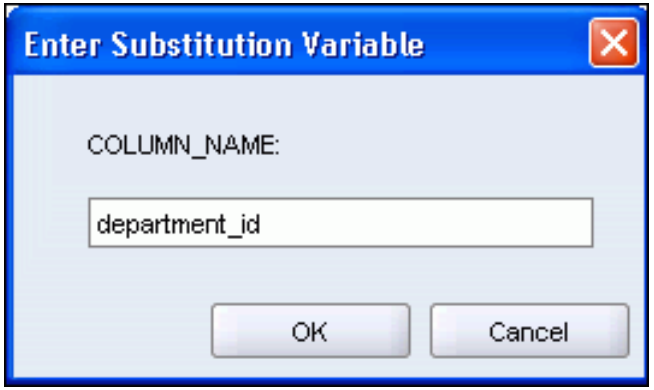

```
SELECT employee_id, last_name, job_id,&column_name
FROM   employees
WHERE  &condition
ORDER BY &order_column ;
```



Utilisation de la variable de substitution Double-Ampersand &&

- Utilisez la double perluète (&&) si vous souhaitez réutiliser la valeur de la variable sans que l'utilisateur ne soit invité à chaque fois:

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

...

PLAN

- Limiter les lignes avec:
  - La clause WHERE
  - Les conditions de comparaison utilisant =, <=, BETWEEN, IN, LIKE et NULL
  - Conditions logiques utilisant les opérateurs AND, OR et NOT
- Règles de priorité pour les opérateurs dans une expression
- Tri des lignes à l'aide de la clause ORDER BY
- Variables de substitution
- Commandes DEFINE et VERIFY

Utilisation de la commande DEFINE

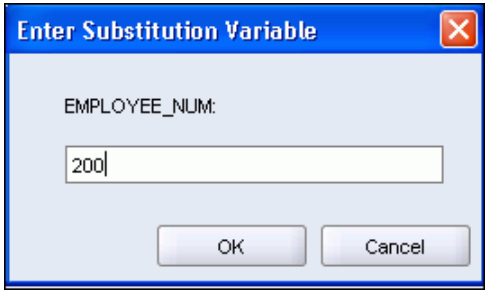
- Utilisez la commande DEFINE pour créer et affecter une valeur à une variable.
- Utilisez la commande UNDEFINE pour supprimer une variable.

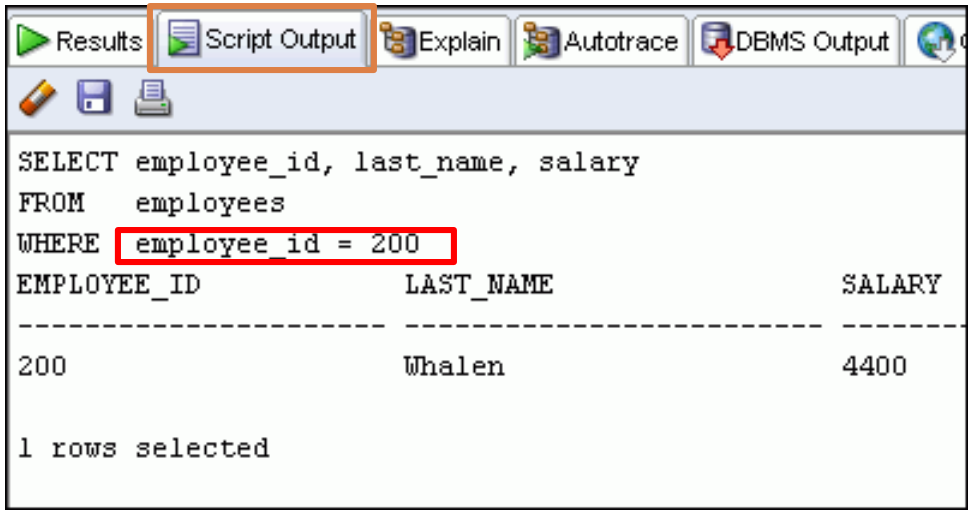
```
DEFINE employee_num = 200

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

UNDEFINE employee_num
```

Utilisation de la commande VERIFY

- Utilisez la commande VERIFY pour afficher le texte d'une commande après avoir remplacé les variables de substitution par des valeurs, suivie du résultat de l'exécution de la commande:

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





TP3

- Ce TP couvre les sujets suivants:
  - Sélection de données et modification de l'ordre des lignes affichées
  - Restriction des lignes à l'aide de la clause WHERE
  - Tri des lignes à l'aide de la clause ORDER BY
  - Utilisation de variables de substitution pour ajouter de la flexibilité à vos instructions SQL SELECT

CHAPITRE 4

4. LE LANGAGE SQL:
UTILISATION DES FONCTIONS À UNE SEULE LIGNE POUR PERSONNALISER LA SORTIE