

**Utiliser des sous-interrogations pour  
résoudre des interrogations**

# Utiliser une sous-interrogation pour résoudre un problème

Qui a un salaire supérieur à celui d'Abel ?

**Interrogation principale :**



**Quels employés ont un salaire supérieur à celui d'Abel ?**

**Sous-interrogation :**



**Quel est le salaire d'Abel ?**



# Syntaxe des sous-interrogations

```
SELECT select_list
FROM   table
WHERE  expr operator
        (SELECT select_list
           FROM   table);
```

- La sous-interrogation (interrogation interne) est exécutée *avant* l'interrogation principale (interrogation externe).
- Le résultat de la sous-interrogation est utilisé par l'interrogation principale.

# Utiliser une sous-interrogation

```
SELECT last_name, salary
FROM employees
WHERE salary > 11000
      (SELECT salary
        FROM employees
        WHERE last_name = 'Abel');
```

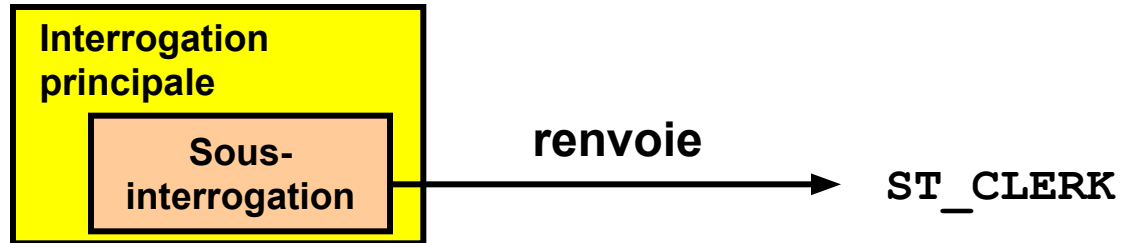
	LAST_NAME	SALARY
1	Hartstein	13000
2	Higgins	12000
3	King	24000
4	Kochhar	17000
5	De Haan	17000

# Règles d'utilisation des sous-interrogations

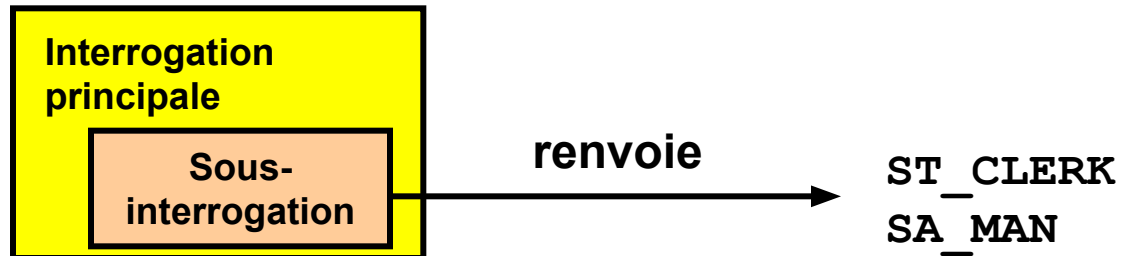
- Mettez les sous-interrogations entre parenthèses.
- Pour une meilleure lisibilité, la sous-interrogation doit figurer dans la partie droite de la condition de comparaison. Toutefois, elle peut apparaître de n'importe quel côté de l'opérateur.
- Utilisez des opérateurs monolignes avec les sous-interrogations monolignes et des opérateurs multilignes avec les sous-interrogations multilignes.

# Types de sous-interrogation

- Sous-interrogation monoligne



- Sous-interrogation multiligne







# Sous-interrogations monolingues

- Elles renvoient une seule ligne.
- Elles utilisent des opérateurs de comparaison monolingues.

Opérateur	Signification
=	Egal à
>	Supérieur à
>=	Supérieur ou égal à
<	Inférieur à
<=	Inférieur ou égal à
<>	Non égal à


# Exécuter des sous-interrogations monolingues




```
SELECT last_name, job_id, salary
FROM employees
WHERE job_id =  SA_REP
AND salary >  8600
      (SELECT job_id
       FROM employees
       WHERE last_name = 'Taylor')
      (SELECT salary
       FROM employees
       WHERE last_name = 'Taylor');
```

	 LAST_NAME	 JOB_ID	 SALARY
1	Abel	SA_REP	11000



# Utiliser des fonctions de groupe dans une sous-interrogation

```
SELECT last_name, job_id, salary
FROM employees
WHERE salary =  2500
              (SELECT MIN(salary)
               FROM employees);
```

	 LAST_NAME	 JOB_ID	 SALARY
1	Vargas	ST_CLERK	2500

# Clause HAVING avec des sous-interrogations

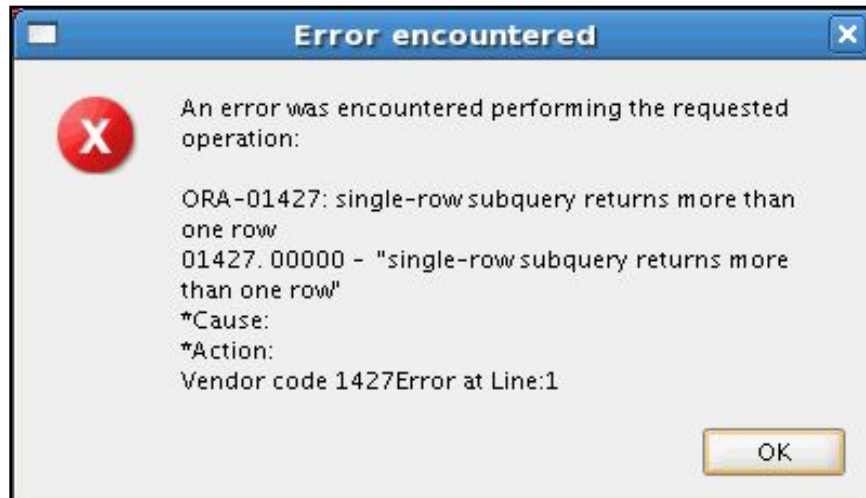
- Le serveur Oracle exécute d'abord les sous-interrogations.
- Le serveur Oracle renvoie les résultats dans la clause HAVING de l'interrogation principale.

```
SELECT    department_id, MIN(salary)
FROM      employees
GROUP BY  department_id
HAVING    MIN(salary) > 2500
           (SELECT MIN(salary)
            FROM      employees
            WHERE      department_id = 50);
```

	DEPARTMENT_ID	MIN(SALARY)
1	(null)	7000
2	20	6000
3	90	17000
4	110	8300
5	80	8600
6	10	4400
7	60	4200

# Qu'est-ce qui est incorrect dans cette instruction ?

```
SELECT employee_id, last_name
FROM employees
WHERE salary =
      (SELECT MIN(salary)
       FROM employees
       GROUP BY department id);
```



**Opérateur monoligne  
avec une  
sous-interrogation  
multiligne**

# L'interrogation interne ne renvoie aucune ligne

```
SELECT last_name, job_id
FROM employees
WHERE job_id =
      (SELECT job_id
       FROM employees
       WHERE last_name = 'Haas');
```

0 rows selected

**La sous-interrogation ne renvoie aucune ligne car il n'existe aucun employé nommé "Haas".**

# Sous-interrogations multilingues

- Elles renvoient plusieurs lignes.
- Elles utilisent des opérateurs de comparaison multilingues.

Opérateur	Signification
IN	Egalité avec n'importe quel membre de la liste
ANY	Doit être précédé par =, !=, >, <, <=, >=. Compare une valeur à chaque valeur d'une liste ou à une valeur renvoyée par une interrogation. Le résultat de l'évaluation est <code>FALSE</code> si l'interrogation ne renvoie aucune ligne.
ALL	Doit être précédé par =, !=, >, <, <=, >=. Compare une valeur à toutes les valeurs d'une liste ou aux valeurs renvoyées par une interrogation. Le résultat de l'évaluation est <code>TRUE</code> si l'interrogation ne renvoie aucune ligne.

# Utiliser l'opérateur ANY

## dans des sous-interrogations multilingues

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

9000, 6000, 4200

	EMPLOYEE_ID	LAST_NAME	JOB_ID	SALARY
1	144	Vargas	ST_CLERK	2500
2	143	Matos	ST_CLERK	2600
3	142	Davies	ST_CLERK	3100
4	141	Rajs	ST_CLERK	3500
5	200	Whalen	AD_ASST	4400

...

9	206	Gietz	AC_ACCOUNT	8300
10	176	Taylor	SA_REP	8600

# Utiliser l'opérateur ALL dans des sous-interrogations multilingues

```
SELECT employee_id, last_name, job_id, salary
FROM employees 9000, 6000, 4200
WHERE salary < ALL
              (SELECT salary
               FROM employees
               WHERE job_id = 'IT_PROG')
AND job_id <> 'IT_PROG';
```

	EMPLOYEE_ID	LAST_NAME	JOB_ID	SALARY
1	141	Rajs	ST_CLERK	3500
2	142	Davies	ST_CLERK	3100
3	143	Matos	ST_CLERK	2600
4	144	Vargas	ST_CLERK	2500

# Utiliser l'opérateur EXISTS

```
SELECT * FROM departments
WHERE NOT EXISTS
(SELECT * FROM employees
 WHERE
employees.department id=departments.department id) ;
```

	DEPARTMENT_ID	DEPARTMENT_NAME	MANAGER_ID	LOCATION_ID
1	190	Contracting	(null)	1700



# Valeurs NULL dans une sous-interrogation

```
SELECT emp.last_name  
FROM   employees emp  
WHERE  emp.employee_id NOT IN  
                                (SELECT mgr.manager_id  
                                FROM   employees mgr);
```

0 rows selected