



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

# Objectifs

**A la fin de ce TP, vous pourrez :**

- **définir des sous-interrogations**
- **décrire les types de problème que les sous-interrogations permettent de résoudre**
- **répertorier les types de sous-interrogation**
- **écrire des sous-interrogations monoligne et multiligne**

# Utiliser une sous-interrogation pour résoudre un problème

Qui a un salaire plus élevé que celui d'Abel ?

Interrogation principale :



Quels employés ont un salaire plus élevé que celui d'Abel ?

Sous-interrogation :



Quel est le salaire d'Abel ?

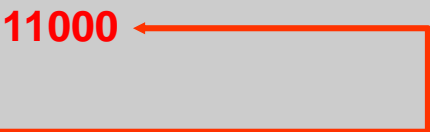
# Syntaxe d'une sous-interrogation

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

- La sous-interrogation (interrogation interne) s'exécute une fois 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
FROM   employees
WHERE  salary >
      (SELECT salary
       FROM   employees
       WHERE  last_name = 'Abel');
```



A red arrow points from the value '11000' to the 'salary >' comparison in the main query. A red box highlights the subquery.

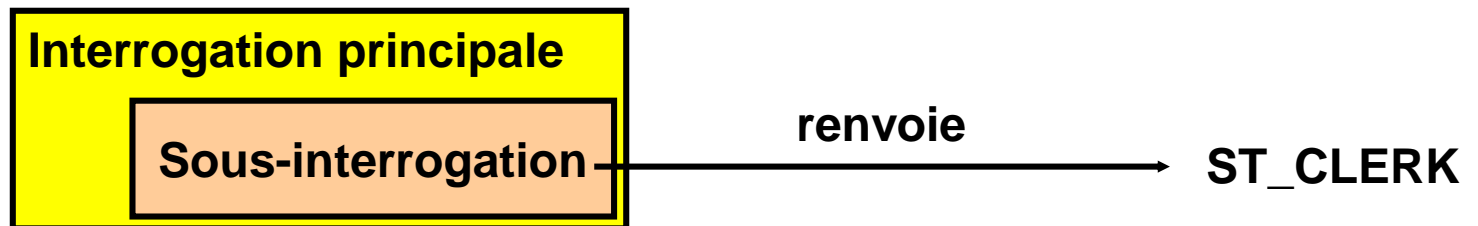
LAST_NAME
King
Kochhar
De Haan
Hartstein
Higgins

# Règles d'utilisation des sous-interrogations

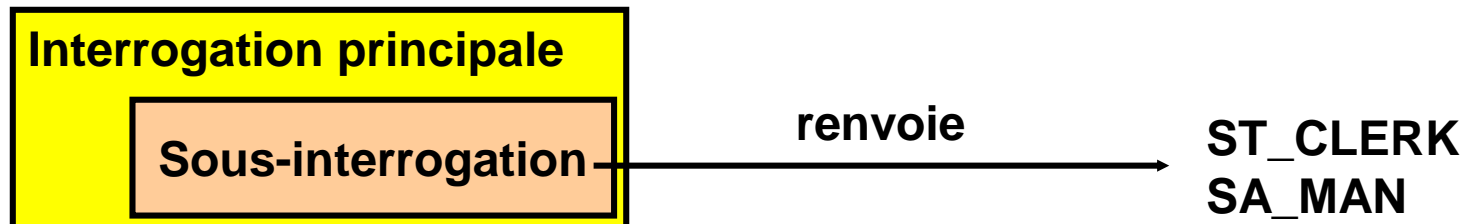
- Incluez les sous-interrogations entre parenthèses.
- Placez les sous-interrogations à droite de la condition de comparaison.
- La clause `ORDER BY` de la sous-interrogation n'est pas nécessaire.
- Utilisez des opérateurs monoligne avec les sous-interrogations monoligne, et des opérateurs multiligne avec les sous-interrogations multiligne.

# Types de sous-interrogation

- **Sous-interrogation monoligne**



- **Sous-interrogation multiligne**





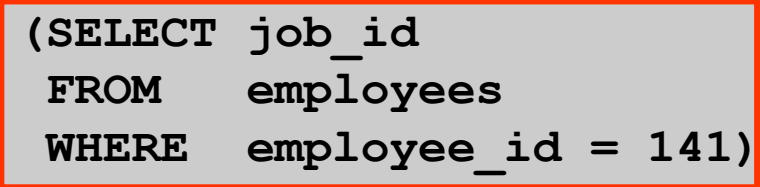

# Sous-interrogations monoligne

- Renvoient une seule ligne
- Utilisent des opérateurs de comparaison monoligne

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




# Exécuter des sous-interrogations monoligne

```
SELECT last_name, job_id, salary
FROM   employees
WHERE  job_id =  ST_CLERK
AND    salary > 
      (
        (SELECT job_id
         FROM   employees
         WHERE  employee_id = 141)
      )
      (
        (SELECT salary
         FROM   employees
         WHERE  employee id = 143));
```

LAST_NAME	JOB_ID	SALARY
Rajs	ST_CLERK	3500
Davies	ST_CLERK	3100

# Utiliser des fonctions de groupe dans une sous-interrogation

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



LAST_NAME	JOB_ID	SALARY
Vargas	ST_CLERK	2500

# La 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) > 2100
           ← (SELECT MIN(salary)
              FROM    employees
              WHERE    department_id = 50);
```

# Quel est le problème de cette instruction ?

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

```
ERROR at line 4:
ORA-01427: single-row subquery returns more than
one row
```

**Opérateur monoligne avec sous-interrogation multiligne**

# Cette instruction renvoie-t-elle des lignes ?

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

no rows selected

**La sous-interrogation ne renvoie aucune valeur.**

# Sous-interrogations multiligne

- **Renvoient plusieurs lignes**
- **Utilisent des opérateurs de comparaison multiligne**

Opérateur	Signification
IN	Egal à un membre quelconque de la liste
ANY	Compare la valeur à n'importe quelle valeur renvoyée par la sous-interrogation
ALL	Compare la valeur à toutes les valeurs renvoyées par la sous-interrogation

# Utiliser l'opérateur ANY dans les sous-interrogations multiligne

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

EMPLOYEE_ID	LAST_NAME	JOB_ID	SALARY
124	Mourgos	ST_MAN	5800
141	Rajs	ST_CLERK	3500
142	Davies	ST_CLERK	3100
143	Matos	ST_CLERK	2600
144	Vargas	ST_CLERK	2500

...  
10 rows selected.

**<ANY** signifie moins que le maximum. **>ANY** signifie plus que le minimum.  
**=ANY** est équivalent à **IN**.

# Utiliser l'opérateur ALL dans les sous-interrogations multiligne

```
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
141	Rajs	ST_CLERK	3500
142	Davies	ST_CLERK	3100
143	Matos	ST_CLERK	2600
144	Vargas	ST_CLERK	2500

>ALL signifie plus que le maximum et <ALL signifie moins que le minimum.  
L'opérateur NOT peut être utilisé avec les opérateurs IN, ANY et ALL.



# Valeurs NULL dans une sous-interrogation

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

no rows selected

# Valeurs NULL dans une sous-interrogation

```
SELECT last_name
FROM   employees
WHERE  employee_id NOT IN
      (SELECT manager_id
       FROM   employees
       WHERE  manager_id IS NOT NULL);
```

# Synthèse

Ce TP vous a permis d'apprendre à :

- identifier dans quels cas une sous-interrogation peut permettre de répondre à une question
- écrire des sous-interrogations lorsqu'une interrogation est basée sur des valeurs inconnues

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