

# *Subqueries*

---

문제해결에 서브쿼리 사용

서브쿼리 구문

서브쿼리 사용

서브쿼리 사용 지침

서브쿼리 유형

# 문제 해결에 서브쿼리 사용

- “누가 Abel의 급여보다 더 많이 받는가?”

메인쿼리



어느 사원의 급여가 Abel의 급여보다 많습니까?

서브쿼리



Abel의 급여는 얼마입니까?

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

- 서브 쿼리(inner query)는 메인 쿼리가 실행되기 전에 한 번만 실행된다.
- 서브 쿼리의 결과는 메인 쿼리(outer query)에 의해 사용된다.

# 서브쿼리 사용

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

LAST_NAME
King
Kochhar
De Haan
Hartstein
Higgins

# 서브쿼리 사용 지침

- 서브 쿼리는 괄호로 묶어야 한다.
- 서브 쿼리는 비교 연산자의 오른쪽에 위치한다.
- 서브 쿼리는 ORDER BY절을 포함할 수 없다.
- 단일행 서브 쿼리는 단일행 연산자를 사용하고, 다중행 서브 쿼리는 다중행 연산자를 사용한다.

# 서브쿼리 유형

## ■ 단일행 서브 쿼리



## ■ 다중행 서브 쿼리



# 단일행 서버쿼리

- 오직 하나의 행만을 반환한다.
- 단일 행 비교 연산자를 사용한다.

Operator	Meaning
=	Equal to
>	Greater than
>=	Greater than or equal to
<	Less than
<=	Less than or equal to
<>	Not equal to

# 단일행 서브쿼리 실행

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

LAST_NAME	JOB_ID	SALARY
Rajs	ST_CLERK	3500
Davies	ST_CLERK	3100



# 서브쿼리에서 그룹함수 사용

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

LAST_NAME	JOB_ID	SALARY
Vargas	ST_CLERK	2500

# HAVING절에 서브쿼리 사용

- 오라클 서버는 서브 쿼리들을 먼저 실행한다.
- 오라클 서버는 메인 쿼리 내의 HAVING 절에 결과를 반환해 준다.

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

# 이 명령문은 무엇이 잘못되었을까요?

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

(SELECT MIN(salary) \*

4행에 오류:

ORA-01427: 단일 행 부속 질의에 2개 이상의 행이 리턴되었습니다

다중행 서브쿼리에 단일 행 연산자를 사용했습니다.

# 이 명령문은 행을 반환할까요?

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


선택된 레코드가 없습니다.

서브쿼리는 어떠한 값도 반환하지 않는다. NULL 비교 연산의 결과는 NULL

# 다중행 서브쿼리

- 하나 이상의 행을 반환한다.
- 다중행 비교 연산자를 사용한다.

Operator	Meaning
IN	목록의 어떤 구성원과 같다.
ANY	값을 서브 쿼리에 의해 반환된 각각의 값과 비교한다.
ALL	값을 서브 쿼리에 의해 반환된 모든 값과 비교한다.

- =ANY 는 IN 과 동일 
- <>ALL 은 NOT IN 과 동일
- <ANY는 최대값보다 작음을 >ANY는 최소값보다 큼을 의미
- <ALL은 최소값보다 작음을 >ALL은 최대값보다 큼을 의미

## 다중행 서브쿼리에 ANY 연산자 사용

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

EMPLOYEE_ID	LAST_NAME	SALARY
124	Mourgos	5800
141	Rajs	3500
142	Davies	3100
143	Matos	2600
144	Vargas	2500
176	Taylor	8600
178	Grant	7000
200	Whalen	4400
202	Fay	6000
206	Gietz	8300

## 다중행 서브쿼리에 ALL 연산자 사용

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

EMPLOYEE_ID	LAST_NAME	SALARY
141	Rajs	3500
142	Davies	3100
143	Matos	2600
144	Vargas	2500

# 서브쿼리에서의 NULL 값

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


no rows selected

- 내부 질의에서 반환한 값 중 하나가 널 값이면 전체 질의가 행을 반환하지 않음
- 비교 연산자를 사용한 NULL 값의 비교 결과는 널





# 서브쿼리에서의 NULL 값

- 부하 직원이 있는 모든 사원을 표시 

```
SELECT emp.last_name  
FROM employees emp  
WHERE emp.employee_id IN  
      (SELECT mgr.manager_id  
       FROM employees mgr);
```

- 부하 직원이 없는 모든 사원을 표시

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

## 실습(아래 테이블은 단순 예시임)

1. Zlotkey와 동일한 부서에 속한 모든 사원의 이름과 입사일을 표시하는 질의를 작성하십시오. (Zlotkey는 제외)

LAST_NAME	HIRE_DATE
Abel	11-MAY-96
Taylor	24-MAR-98

2. 급여가 평균 급여보다 많은 모든 사원의 사원 번호와 이름을 표시하는 질의를 작성하고 결과를 급여에 대해 오름차순으로 정렬하십시오.

EMPLOYEE_ID	LAST_NAME	SALARY
103	Hunold	9000
149	Zlotkey	10500

3. 이름에 u가 포함된 사원과 같은 부서에서 일하는 모든 사원의 사원 번호와 이름을 표시하는 질의를 작성하십시오.

EMPLOYEE_ID	LAST_NAME
124	Mourgos
141	Rajs
142	Devi

## 실습(아래 테이블은 단순 예시임)

4. 부서 위치 ID가 1700인 모든 사원의 이름, 부서 번호 및 업무 ID를 표시하십시오.

LAST_NAME	DEPARTMENT_ID	JOB_ID
Whalen	10	AD_ASST
King	90	AD PRES
Kochhar	90	AD VP

5. King에게 보고하는 모든 사원 이름과 급여를 표시하십시오.

LAST_NAME	SALARY
Kochhar	17000
De Haan	17000
Mourgos	15000

6. Executive 부서의 모든 사원에 대한 부서 번호, 이름 및 업무 ID를 표시하십시오.

DEPARTMENT_ID	LAST_NAME	JOB_ID
90	King	AD PRES
90	Kochhar	AD VP

## 실습(아래 테이블은 단순 예시임)

7. 평균 급여보다 많은 급여를 받고 이름에 u가 포함된 사원과 같은 부서에서 근무하는 모든 사원의 번호, 이름 및 급여를 표시하십시오.

EMPLOYEE_ID	LAST_NAME	SALARY
103	Hunold	9000

8. 미국(locations.country\_id = 'US') 내에서 근무하는 직원들의 평균 급여보다 많은 급여를 받는 사원의 번호, 이름 및 급여를 표시하십시오.



9. 부서 별로 최고 급여를 받는 사원의 번호, 이름, 급여 및 부서 번호를 표시하고 부서 번호에 대해 오름 차순 정렬을 하시오.

## 실습(아래 테이블은 단순 예시임)

10. FROM 절을 사용하여 9 번 질의를 재 작성하시오.

11. 사원이 한 명 이상 존재하는 부서의 번호 및 부서 이름을 표시하시오.  
(exists 키워드 사용)



12. 다음을 참고하여 급여를 가장 적게 받는 사원 5명에 대한 정보를 표시하시오.

```
select rownum, employee_id, last_name, salary
from employees
where rownum <= 5
```

