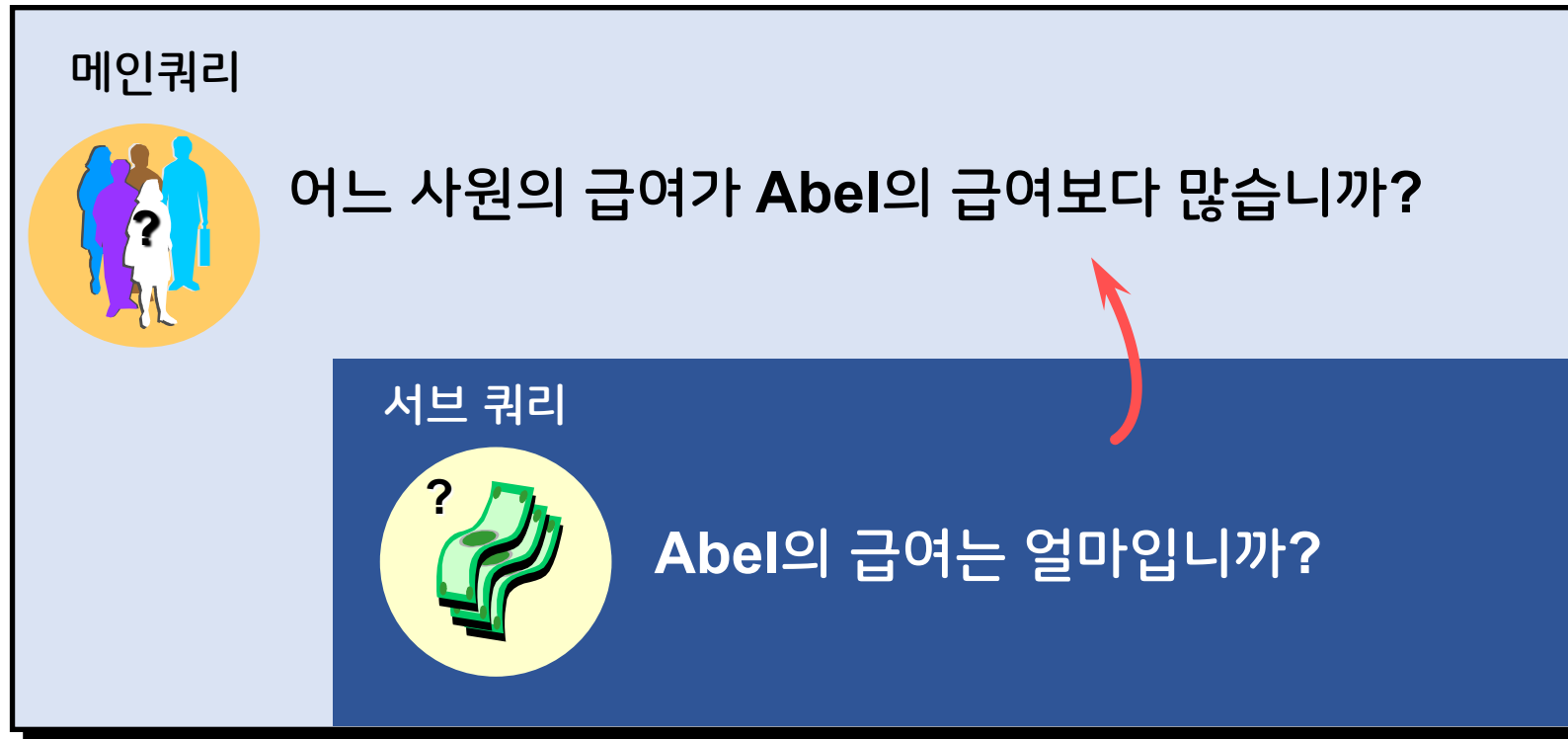


6. 기본 서브쿼리(SUBQUERY)



문제 해결에 서브 쿼리 사용

- Abel보다 급여가 많은 사원은 누구입니까?



서브 쿼리

- 서브 쿼리의 개념
 - 서브쿼리는 SELECT 문의 절에 삽입된 SELECT 문입니다.
 - 간단한 명령문으로 강력한 기능을 제공하는 명령문을 작성할 수 있습니다.
- 다음과 같은 여러 SQL 절에 서브 쿼리를 포함시킬 수 있습니다.
 - WHERE 절
 - HAVING 절
 - FROM 절

서브 쿼리 구문

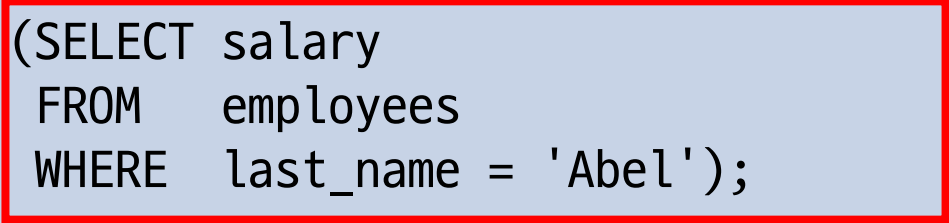
- 서브 쿼리(내부 질의)는 기본 질의 실행 전에 한 번 실행됩니다.
- 서브 쿼리의 결과는 메인쿼리(외부 질의)에 사용됩니다.
- 구문

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

서브 쿼리 사용

- 다음 서브쿼리는 Abel 사원의 급여를 확인하고 메인쿼리는 서브쿼리의 결과를 사용하여 해당 급여보다 급여가 많은 사원을 모두 표시합니다.

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



- 서브 쿼리(내부 질의)는 기본 질의 실행 전에 한 번 실행 됩니다.
- 서브 쿼리의 결과는 기본 질의(메인 쿼리)에 사용됩니다.

서브 쿼리 사용 지침

- 서브 쿼리를 괄호로 묶습니다.
- 비교 조건의 오른쪽에 서브 쿼리를 넣습니다.
- TOP-N 분석이 아니면 서브 쿼리 내의 ORDER BY 절은 필요없습니다.
- 서브 쿼리에 사용되는 비교 조건은 단일 행 연산자 및 다중 행 연산자로 분류됩니다.
 - 단일 행 서브 쿼리에는 단일 행 연산자를 사용하고 다중 행 서브 쿼리에는 다중 행 연산자를 사용해야 합니다.

서브 쿼리 유형

▪ 단일 행 서브쿼리

- 내부 SELECT 문에서 한 행만 반환하는 쿼리



▪ 다중 행 서브쿼리

- 다중 행 서브 쿼리: 내부 SELECT 문에서 여러 행을 반환하는 질의



단일 행 서브 쿼리

- 서버쿼리의 실행결과로 한 행만 반환합니다.
- 단일 행 비교 연산자를 사용합니다.
- 단일 행 비교 연산자



연산자	의미
=	같음
>	보다 큼
>=	크거나 같음
<	보다 작음
<=	작거나 같음
<>	같지 않음

단일 행 서브쿼리의 실행

- 다음 예제는 사원 141과 업무 ID가 동일하면서 사원 143보다 급여가 많은 사원을 표시합니다.

```
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);
```

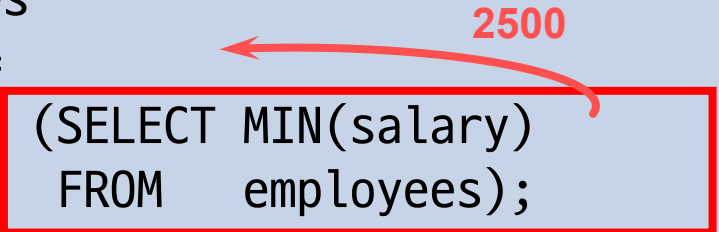
ST_CLERK

2600

서브 쿼리에서 그룹 함수 사용

- 단일 행을 반환하는 그룹 함수를 서브 쿼리에서 사용하여 기본 질의에서 데이터를 표시할 수 있습니다.
- 다음 예제는 최소 급여를 받는 모든 사원의 이름, 업무 ID 및 급여를 표시하며 MIN 그룹 함수는 단일 값 (2500)을 외부 질의에 반환합니다.

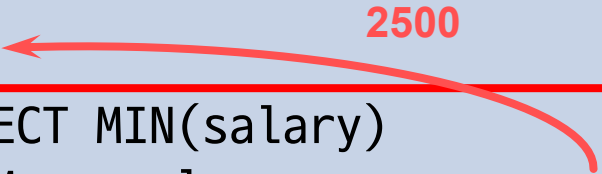
```
SELECT last_name, job_id, salary
FROM   employees
WHERE  salary =
```



```
      (SELECT MIN(salary)
        FROM   employees);
```

HAVING 절에 서브 쿼리 사용

- Oracle server는 서브 쿼리를 먼저 실행합니다.
- Oracle server는 메인쿼리의 HAVING 절에 결과를 반환합니다.
- 다음은 최소 급여가 부서 50의 최소 급여보다 많은 부서를 모두 표시합니다.

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

서브 쿼리 오류

- 서브 쿼리의 일반적인 오류는 단일 행 서브 쿼리에서 여러 행이 반환되는 것입니다.

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

ORA-01427: single-row subquery returns more than one row
01427. 00000 - "single-row subquery returns more than one row"

다중 행 서브 쿼리에 단일 행 연산자를 사용했습니다.

서브쿼리에서 발생 가능한 문제

- 서브 쿼리에서 발생할 수 있는 일반적인 문제는 내부 질의에서 행을 반환하지 않는 경우입니다.
- 이 명령문은 유효하지만 이름이 Haas라는 사원이 없으므로 실행했을 때 선택되는 행이 없습니다.

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

- 서브쿼리의 결과가 널이지만 manager_id가 널인 데이터를 출력할 수 없습니다.

다중 행 서브 쿼리

- 여러 행을 반환하는 서브 쿼리를 여러 행 서브 쿼리라고 하는데 여러 행 서브 쿼리에서는 단일 행 연산자 대신 여러 값을 처리하는 여러 행 연산자를 사용합니다.
 - 여러 행을 반환합니다.
 - 여러 행을 비교할 수 있는 다중행 비교 연산자를 사용합니다.
- 다중행 비교 연산자

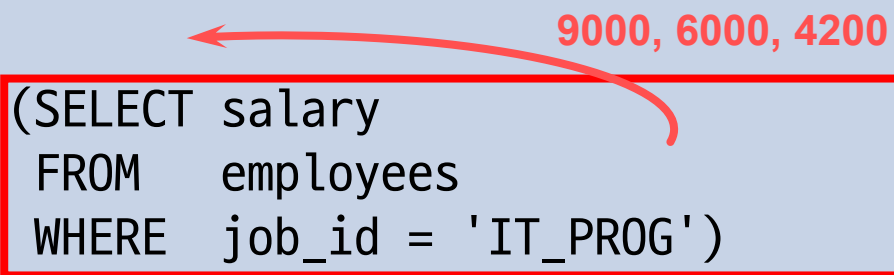


연산자	의미
IN	목록에 있는 임의의 멤버와 동일합니다.
ANY	값을 서브 쿼리에 의해 반환된 각 값과 비교합니다.
ALL	값을 서브 쿼리에 의해 반환된 모든 값과 비교합니다.

다중 행 서브 쿼리에 ANY 연산자 사용

- 다음 예제는 IT 프로그래머가 아니면서 급여가 임의의 IT 프로그래머보다 낮은 사원을 표시합니다.
 - 프로그래머의 최고 급여는 \$9,000입니다.

```
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';
```

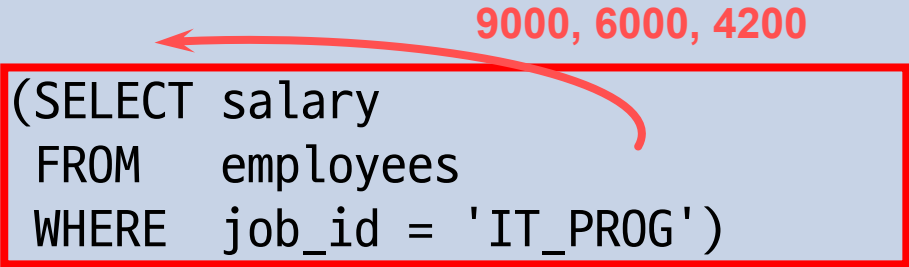


9000, 6000, 4200

다중 행 서브 쿼리에 ALL 연산자 사용

- 다음 예제는 업무 ID가 IT_PROG인 모든 사원보다 급여가 적으면서 업무가 IT_PROG가 아닌 사원을 표시합니다.

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



9000, 6000, 4200

서브 쿼리에서의 널 값

- IN 연산자를 사용할 때는 서브 쿼리 결과 집합에 널 값이 있어도 됩니다.
- 다음은 부하 직원이 있는 모든 사원을 표시합니다.

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

서브 쿼리에서의 널 값

- 부하 직원이 없는 모든 사원을 표시하기위한 다음 SQL 문은 논리적으로는 12개의 행을 반환해야 하지만 아무 행도 반환되지 않습니다.
- 내부 질의에서 반환한 값 중 하나가 널 값이므로 전체 질의가 행을 반환하지 않는 것입니다.

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

no rows selected

서브 쿼리에서의 널 값

- 서브 쿼리에 다음과 같은 WHERE 절을 포함시켜 부하 직원이 없는 모든 사원을 표시할 수 있습니다.

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

- 이 경우 이후 학습할 NOT EXISTS 연산자로 대체될 수 있습니다.

Thank you 😊