**로고학습일지**

**kt ds University 자바 기반의 데이터 사이언티스트 양성과정**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 학습일시 | 2018. 08. 03 (금) | 장소 | kt ds University B관 201호 | **시 간** | 09:00~18:00 |
| 학습범위 | 손에 잡히는 10분 SQL 11장 | | | | |
| 작 성 자 | 신예리 | | | **강 사** | 장민창 강사 |

|  |  |
| --- | --- |
| 학습안건 | 서브쿼리 |

|  |  |
| --- | --- |
| 학습내용 | 내용 |
| 서브쿼리 이용하기  >2007/1/14에 입사한 사람이 받는 연봉보다 더 많은 연봉을 받는 사람들을 조회해라  >10번 부서의 평균 연봉보다 더 많은 연봉을 받는 사원들을 조회해라  >IT\_PROG 직무의 평균 연봉보다 더 많은 연봉을 받는 사원들을 조회해라  >자신의 상사보다 더 많은 연봉을 받는 사원들을 조회해라  🡺 결과와 컬럼을 비교할 때 반드시 필요함.  >FIRST\_NAME이 Steven인 사원과 같은 부서//에서 일을 하는 // 모든 사원들의 정보를 조회  ┗>1. EMPLOYEES  >**소괄호**가 먼저 시행된다.  -서브 쿼리를 먼저 실행해보자. 서브쿼리의 결과가 2개 이상인 경우 IN 사용 (=을 사용하면 에러)  >INLINE VIEW / INLINE TABLE : 쿼리의 결과를 테이블로 쓰는 것  SELECT \*  FROM (  SELECT \*  FROM (  SELECT DEPARTMENT\_ID  , AVG(SALARY) AVG\_SALARY  FROM EMPLOYEES  GROUP BY DEPARTMENT\_ID  )  WHERE ROWNUM = 1  ORDER BY AVG\_SALARY DESC  )  WHERE ROWNUM = 1  ;  -제일 많이 판매된 상품을 조회해라.  -조회수가 가장 많은 광고를 조회해라.  -Pagenation : 웹 페이지 밑에 1 2 3 4…. (데이터가 많아 페이지를 나누어 보여주는 것) |

|  |  |
| --- | --- |
| 특이사항 | 다음주 월요일 JAVA 시험 |

■ORACLE

/\*

FIRST\_NAME이 Steven인 사원과 같은 부서에서 일을 하는 모든 사원들의 정보를 조회

\*/

SELECT \*

FROM EMPLOYEES

WHERE DEPARTMENT\_ID IN (

SELECT DEPARTMENT\_ID

FROM EMPLOYEES

WHERE FIRST\_NAME = 'Steven'

) --소괄호가 먼저 시행

;

/\*

FIRST\_NAME이 Steven인 사원과 같은 부서에서 일을 하는 모든 사원들의 정보를 조회

단, 본인은 결과에서 제회한다.

\*/

SELECT \*

FROM EMPLOYEES

WHERE DEPARTMENT\_ID IN (

SELECT DEPARTMENT\_ID

FROM EMPLOYEES

WHERE FIRST\_NAME = 'Steven'

)

AND FIRST\_NAME != 'Steven'

;

/\*

전체사원의 평균 연봉보다 더 많은 연봉을 받는 사원들의 사원번호, 이름(FIRST\_NAME, LAST\_NAME), 연봉 그리고 부서명을 조회

연봉 오름차순 정렬

\*/

SELECT E.EMPLOYEE\_ID

, E.FIRST\_NAME || ' ' || E.LAST\_NAME NAME

, E.SALARY

, D.DEPARTMENT\_NAME

FROM EMPLOYEES E

, DEPARTMENTS D

WHERE E.DEPARTMENT\_ID = D.DEPARTMENT\_ID

AND E.SALARY > (

SELECT ROUND (AVG(SALARY) ,0)

FROM EMPLOYEES

)

ORDER BY E.SALARY

;

/\*

Seattle에서 근무하는 모든 사원들을 조회하라.

Case 1. Join

Case 2. Sub Query

\*/

-- 1. Join

SELECT E.\*

, L.CITY

FROM EMPLOYEES E

, DEPARTMENTS D

, LOCATIONS L

WHERE E.DEPARTMENT\_ID = D.DEPARTMENT\_ID

AND L.LOCATION\_ID = D.LOCATION\_ID

AND L.CITY = 'Seattle'

ORDER BY EMPLOYEE\_ID

;

-- 2. Sub Query

SELECT \*

FROM EMPLOYEES

WHERE DEPARTMENT\_ID IN (

SELECT DEPARTMENT\_ID

FROM DEPARTMENTS

WHERE LOCATION\_ID = (

SELECT LOCATION\_ID

FROM LOCATIONS

WHERE CITY = 'Seattle'

)

)

ORDER BY EMPLOYEE\_ID

;

/\*

평균 급여가 가장 많은 부서를 조회하라.

(INLINE-VIEW / INLINE-TABLE) : 쿼리의 결과를 테이블로 쓰는 것

\*/

SELECT \*

FROM ( -- VIEW

SELECT DEPARTMENT\_ID

, AVG(SALARY) AVG\_SALARY

FROM EMPLOYEES

GROUP BY DEPARTMENT\_ID

)

ORDER BY AVG\_SALARY DESC

;

SELECT \*

FROM (

SELECT \*

FROM (

SELECT DEPARTMENT\_ID

, AVG(SALARY) AVG\_SALARY

FROM EMPLOYEES

GROUP BY DEPARTMENT\_ID

)

ORDER BY AVG\_SALARY DESC

)

WHERE ROWNUM = 1 -- 행번호 1번

;

/\*

부서별 평균 연봉이 회사 전체 평균보다 적은 부서명과 평균연봉을 조회하라

\*/

SELECT D.DEPARTMENT\_NAME

, A.AVG\_SALARY

FROM DEPARTMENTS D

, (SELECT DEPARTMENT\_ID

, AVG(SALARY) AVG\_SALARY

FROM EMPLOYEES E

GROUP BY DEPARTMENT\_ID

HAVING AVG(SALARY) < (

SELECT AVG(SALARY)

FROM EMPLOYEES

)

) A

WHERE D.DEPARTMENT\_ID = A.DEPARTMENT\_ID

ORDER BY D.DEPARTMENT\_NAME

;

-- 강사님1 (셋 중 가장 느림)

SELECT D.DEPARTMENT\_NAME

, AVG\_D.AVG\_SALARY

FROM (

SELECT DEPARTMENT\_ID

, AVG(SALARY) AVG\_SALARY

FROM EMPLOYEES

GROUP BY DEPARTMENT\_ID

) AVG\_D

, DEPARTMENTS D

WHERE AVG\_D.DEPARTMENT\_ID = D.DEPARTMENT\_ID

AND AVG\_SALARY < (

SELECT AVG(SALARY)

FROM EMPLOYEES

)

;

--강사님2 (2,3 속도 같음)

SELECT AVG\_D.DEPARTMENT\_NAME

, AVG\_D.AVG\_SALARY

FROM (

SELECT D.DEPARTMENT\_NAME

, AVG(SALARY) AVG\_SALARY

FROM EMPLOYEES E

, DEPARTMENTS D

WHERE D.DEPARTMENT\_ID = E.DEPARTMENT\_ID

GROUP BY D.DEPARTMENT\_NAME

) AVG\_D

WHERE AVG\_SALARY < (

SELECT AVG(SALARY)

FROM EMPLOYEES

)

;

--강사님3 (2,3 속도 같음)

SELECT D.DEPARTMENT\_NAME

, AVG(SALARY) AVG\_SALARY

FROM EMPLOYEES E

, DEPARTMENTS D

WHERE E.DEPARTMENT\_ID = D.DEPARTMENT\_ID

GROUP BY D.DEPARTMENT\_NAME

HAVING AVG(SALARY) < (

SELECT AVG(SALARY)

FROM EMPLOYEES

)

;

/\*

직무 아이디별 평균 연봉이 회사 전체 평균 연봉보다 크거나 같은 직무를 가진 사원들의 모든 정보를 출력

\*/

SELECT E.\*

FROM EMPLOYEES E

, (

SELECT JOB\_ID

, AVG(SALARY) AVG\_SALARY

FROM EMPLOYEES

GROUP BY JOB\_ID

HAVING AVG(SALARY) >= (

SELECT AVG(SALARY)

FROM EMPLOYEES

)

) A

WHERE E.JOB\_ID = A.JOB\_ID

;

--김경준

SELECT E.\*

FROM (

SELECT JOB\_ID

, AVG(SALARY) JOB\_AVG

FROM EMPLOYEES

GROUP BY JOB\_ID

) J

, EMPLOYEES E

WHERE J.JOB\_AVG >= (

SELECT AVG(SALARY)

FROM EMPLOYEES

)

AND J.JOB\_ID = E.JOB\_ID

;

--강사님

SELECT \*

FROM EMPLOYEES E

, (

SELECT JOB\_ID

, AVG(SALARY) AVG\_SALARY

FROM EMPLOYEES

GROUP BY JOB\_ID

HAVING AVG(SALARY) >= (

SELECT AVG(SALARY)

FROM EMPLOYEES

)

) JOB\_AVG

WHERE E.JOB\_ID = JOB\_AVG.JOB\_ID

;

/\*

사원이 단 한명도 존재하지 않는 도시명을 조회

\*/

SELECT CITY

FROM LOCATIONS

WHERE LOCATION\_ID NOT IN (

SELECT DISTINCT LOCATION\_ID

FROM EMPLOYEES E

, DEPARTMENTS D

WHERE E.DEPARTMENT\_ID = D.DEPARTMENT\_ID

)

ORDER BY CITY

;

/\*

강사님

EMPLOYEES - DEPARTMENTS - LOCATIONS

1. 사원이 없는 부서의 LOCATION\_ID 조회 (중복제거)

2. LOCATIONS에서 사원이 있는 부서의 LOCATION\_ID 제외

\*/

SELECT CITY

FROM LOCATIONS

WHERE LOCATION\_ID NOT IN (

SELECT DISTINCT LOCATION\_ID

FROM EMPLOYEES E

, DEPARTMENTS D

WHERE E.DEPARTMENT\_ID = D.DEPARTMENT\_ID

)

;

SELECT CITY

FROM LOCATIONS

WHERE LOCATION\_ID IN (

SELECT LOCATION\_ID

FROM LOCATIONS

MINUS

SELECT DISTINCT LOCATION\_ID

FROM EMPLOYEES E

, DEPARTMENTS D

WHERE E.DEPARTMENT\_ID = D.DEPARTMENT\_ID

)

;

/\*

부서의 사원수가 5명 미만인 부서의 부서별 평균 연봉을 구해 평균 연봉이 가장 높은

부서의 부서명을 조회

\*/

SELECT DEPARTMENT\_NAME

FROM DEPARTMENTS D

, (

SELECT DEPARTMENT\_ID

, AVG(SALARY)

FROM EMPLOYEES

GROUP BY DEPARTMENT\_ID

HAVING COUNT(1) < 5

ORDER BY AVG(SALARY) DESC

) A

WHERE ROWNUM = 1

AND D.DEPARTMENT\_ID = A.DEPARTMENT\_ID

;

--강사님

SELECT DEPARTMENT\_NAME

FROM (

SELECT D.DEPARTMENT\_NAME

, AVG(SALARY) AVG

FROM DEPARTMENTS D

, EMPLOYEES E

WHERE D.DEPARTMENT\_ID = E.DEPARTMENT\_ID

GROUP BY D.DEPARTMENT\_NAME

HAVING COUNT(1) < 5

ORDER BY AVG DESC

)

WHERE ROWNUM = 1

;