**로고학습일지**

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

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 학습일시 | 2018. 08. 06 (월) | 장소 | kt ds University B관 201호 | **시 간** | 09:00~18:00 |
| 학습범위 |  | | | | |
| 작 성 자 | 신예리 | | | **강 사** | 장민창 강사 |

|  |  |
| --- | --- |
| 학습안건 | SQL 문제2 |

|  |  |
| --- | --- |
| 학습내용 | 내용 |
| <DB>  재귀 참조  >자기 자신을 참조  >33. 다른 사원의 상사인 사원의 모든 정보를 조회하라.  계층 쿼리 (CONNECT BY PRIOR) ★★★★★  >39. 100번 사원을 상사로 두고 있는 모든 사원들의 모든 정보를 조회하라.  > SELECT LEVER -- 번호  , E.\*  FROM EMPLOYEES E  START WITH E.MANAGER\_ID = 100 -- 어디서부터 시작할텐가?  CONNECT BY PRIOR E.EMPLOYEE\_ID = E.MANAGER\_ID -- 둘을 조인해줌 (조인과 다른 점: 계층식으로 조인)  ;  🡺 100  101  103  109 ~ 113  200  203  204  205  206  102…  > 쓰이는 곳  -쇼핑몰 : 브랜드 패션 > 브랜드 의류 > 브랜드 여성의류, 브랜드 남성의류  > 브랜드 잡화  > 메뉴 구성할 때 반드시 사용되는 것!!!! |

|  |  |
| --- | --- |
| 특이사항 | JAVA 시험본 날 |

■ORACLE

-- 1. 모든 사원의 이름(FIRST\_NAME, LAST\_NAME)을 조회하라.

SELECT FIRST\_NAME || ' ' || LAST\_NAME NAME

FROM EMPLOYEES

;

-- 2. 모든 사원의 모든 정보를 조회하라.

SELECT \*

FROM EMPLOYEES

;

-- 3. 모든 도시 명을 조회하라.

SELECT CITY

FROM LOCATIONS

;

-- 4. 이름(FIRST\_NAME)이 M 으로 시작하는 사원의 모든 정보를 조회하라.

SELECT \*

FROM EMPLOYEES

WHERE FIRST\_NAME LIKE 'M%'

;

-- 5. 이름(FIRST\_NAME)의 두 번째 글자가 'a'인 사원의 이름(FIRST\_NAME)과 연봉을 조회하라.

SELECT FIRST\_NAME

, SALARY

FROM EMPLOYEES

WHERE FIRST\_NAME LIKE '\_a%'

;

-- 6. 도시 명을 오름차순 정렬하라.

SELECT CITY

FROM LOCATIONS

ORDER BY CITY

;

-- 7. 부서 명을 내림차순 정렬하라.

SELECT DEPARTMENT\_NAME

FROM DEPARTMENTS

ORDER BY DEPARTMENT\_NAME DESC

;

-- 8. 연봉이 7000 이상인 사원들의 모든 정보를 연봉순(오름차순)으로 정렬하라.

SELECT \*

FROM EMPLOYEES

WHERE SALARY > 7000

ORDER BY SALARY

;

-- 9. 인센티브(COMMISSION\_PCT)를 받지 않는 사원들의 모든 정보를 조회하라.

SELECT \*

FROM EMPLOYEES

WHERE COMMISSION\_PCT IS NULL

;

-- 10. 인센티브(COMMISSION\_PCT)를 받는 사원들의 모든 정보를 조회하라.

SELECT \*

FROM EMPLOYEES

WHERE COMMISSION\_PCT IS NOT NULL

;

-- 11. 2007년 06월 21일에 입사한 사원의 사원번호, 이름(FIRST\_NAME, LAST\_NAME) 그리고 부서번호를 조회하라.

SELECT EMPLOYEE\_ID

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

, DEPARTMENT\_ID

FROM EMPLOYEES

WHERE HIRE\_DATE = TO\_DATE('2007/06/21', 'YYYY-MM-DD')

;

-- 12. 2006년에 입사한 사원의 사원번호와 입사일을 조회하라.

SELECT EMPLOYEE\_ID

, HIRE\_DATE

FROM EMPLOYEES

WHERE HIRE\_DATE > TO\_DATE('2006/01/01', 'YYYY-MM-DD')

AND HIRE\_DATE < TO\_DATE('2006/12/31', 'YYYY-MM-DD')

;

-- 13. 이름(FIRST\_NAME)의 길이가 5글자 이상인 사원을 조회하라.

SELECT \*

FROM EMPLOYEES

WHERE FIRST\_NAME LIKE '%\_\_\_\_\_%'

;

-- 14. 부서번호별 사원수를 조회하라. (부서번호 오름차순 정렬)

SELECT DEPARTMENT\_ID

, COUNT(1)

FROM EMPLOYEES

GROUP BY DEPARTMENT\_ID

ORDER BY DEPARTMENT\_ID

;

-- 15. 직무 아이디별 평균 연봉을 조회하라. (직무 아이디 내림차순 정렬)

SELECT JOB\_ID

, AVG(SALARY)

FROM EMPLOYEES

GROUP BY JOB\_ID

ORDER BY JOB\_ID DESC

;

-- 16. 상사가 있는 사원들의 모든 정보를 조회하라.

SELECT \*

FROM EMPLOYEES

WHERE MANAGER\_ID IS NOT NULL

;

-- 17. 상사가 없는 사원들의 모든 정보를 조회하라.

SELECT \*

FROM EMPLOYEES

WHERE MANAGER\_ID IS NULL

;

-- 18. 모든 사원들의 사원번호, 이름(FIRST\_NAME, LAST\_NAME), 부서번호 그리고 부서명을 조회하라.

SELECT E.EMPLOYEE\_ID

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

, D.DEPARTMENT\_ID

, D.DEPARTMENT\_NAME

FROM EMPLOYEES E

,DEPARTMENTS D

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

;

-- 19. 모든 부서의 부서명과 도시명을 조회하라.

SELECT D.DEPARTMENT\_NAME

, L.CITY

FROM DEPARTMENTS D

, LOCATIONS L

WHERE D.LOCATION\_ID = L.LOCATION\_ID

;

-- 20. 모든 사원들의 사원번호, 부서명, 직무명을 조회하라.

SELECT E.EMPLOYEE\_ID

, D.DEPARTMENT\_NAME

, J.JOB\_TITLE

FROM EMPLOYEES E

, DEPARTMENTS D

, JOBS J

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

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

;

-- 21. 모든 사원들의 사원번호, 부서명, 직무명, 도시명을 조회하라.

SELECT E.EMPLOYEE\_ID

, D.DEPARTMENT\_NAME

, J.JOB\_TITLE

, L.CITY

FROM EMPLOYEES E

, DEPARTMENTS D

, JOBS J

, LOCATIONS L

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

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

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

ORDER BY EMPLOYEE\_ID

;

-- 22. 10번, 20번, 30번 부서에서 근무하는 사원들의 모든 정보를 조회하라.

SELECT \*

FROM EMPLOYEES

WHERE DEPARTMENT\_ID IN (10, 20, 30)

;

-- 23. 6인 미만의 사원이 근무하는 부서의 이름을 조회하라.

SELECT D.DEPARTMENT\_NAME

, COUNT(1)

FROM EMPLOYEES E

, DEPARTMENTS D

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

GROUP BY D.DEPARTMENT\_NAME

HAVING COUNT(1) < 6

;

-- 24. 4인 미만의 사원이 근무하는 부서의 평균 연봉과 부서명을 조회하라.

SELECT D.DEPARTMENT\_NAME

, AVG(SALARY)

, COUNT(1)

FROM EMPLOYEES E

, DEPARTMENTS D

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

GROUP BY D.DEPARTMENT\_NAME

HAVING COUNT(1) < 4

;

-- 25. IT 부서의 연봉 총합을 조회하라.

SELECT SUM(E.SALARY)

FROM EMPLOYEES E

, DEPARTMENTS D

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

AND D.DEPARTMENT\_NAME = 'IT'

;

-- 26. 대륙명(REGIONS)별 도시의 개수를 조회하라.

SELECT R.REGION\_NAME

, COUNT(1)

FROM REGIONS R

, (

SELECT C.COUNTRY\_ID

FROM LOCATIONS L

, COUNTRIES C

WHERE C.COUNTRY\_ID = L.COUNTRY\_ID

GROUP BY C.COUNTRY\_ID

) A

WHERE R.REGION\_ID = A.REGION\_ID

GROUP BY REGION\_NAME

;

--보경

SELECT R.REGION\_NAME

, COUNT(CITY)

FROM REGIONS R

, COUNTRIES C

, LOCATIONS L

WHERE R.REGION\_ID = C.REGION\_ID

AND C.COUNTRY\_ID = L.COUNTRY\_ID

GROUP BY R.REGION\_NAME

;

--강사님1

SELECT R.REGION\_NAME

, COUNT(1)

FROM COUNTRIES C

, LOCATIONS L

, REGIONS R

WHERE C.COUNTRY\_ID = L.COUNTRY\_ID

AND C.REGION\_ID = R.REGION\_ID

GROUP BY R.REGION\_NAME

;

--강사님2

SELECT R.REGION\_NAME

, C.CNT

FROM (

SELECT C.REGION\_ID

, COUNT(1) CNT

FROM COUNTRIES C

, LOCATIONS L

WHERE C.COUNTRY\_ID = L.COUNTRY\_ID

GROUP BY C.REGION\_ID

) C

, REGIONS R

WHERE C.REGION\_ID = R.REGION\_ID

;

-- 27. 도시명 별 부서의 개수를 조회하라.

SELECT L.CITY

, COUNT(1)

FROM DEPARTMENTS D

, LOCATIONS L

WHERE D.LOCATION\_ID = L.LOCATION\_ID

GROUP BY L.CITY

;

-- 28. 부서가 존재하지 않는 도시를 조회하라.

SELECT CITY

FROM LOCATIONS

WHERE LOCATION\_ID NOT IN (

SELECT LOCATION\_ID

FROM DEPARTMENTS

GROUP BY LOCATION\_ID

)

;

-- 29. 사원이 존재하지 않는 국가를 조회하라.

SELECT COUNTRY\_NAME

FROM COUNTRIES

WHERE COUNTRY\_ID NOT IN (

SELECT DISTINCT COUNTRY\_ID

FROM LOCATIONS L

, DEPARTMENTS D

, EMPLOYEES E

WHERE L.LOCATION\_ID = D.LOCATION\_ID

AND D.DEPARTMENT\_ID = E.DEPARTMENT\_ID

)

;

-- 30. 150번 사원보다 빨리 입사한 사원 중 가장 최신에 입사한 사원의 모든 정보를 조회하라.

--세현언니

SELECT \*

FROM (

SELECT \*

FROM EMPLOYEES

WHERE HIRE\_DATE < (

SELECT HIRE\_DATE

FROM EMPLOYEES

WHERE EMPLOYEE\_ID =150

)

ORDER BY HIRE\_DATE DESC

)

WHERE ROWNUM = 1

;

-- 31. 직무를 전환한 이력이 있는 사원의 모든 정보를 조회하라.

SELECT \*

FROM EMPLOYEES

WHERE EMPLOYEE\_ID IN (

SELECT EMPLOYEE\_ID

FROM JOB\_HISTORY

)

;

-- 32. 부서장으로 있는 사원의 모든 정보와 부서명을 조회하라.

SELECT \*

FROM EMPLOYEES

WHERE EMPLOYEE\_ID IN (

SELECT MANAGER\_ID

FROM DEPARTMENTS

)

;

-- 33. 다른 사원의 상사인 사원의 모든 정보를 조회하라.

SELECT \*

FROM EMPLOYEES

WHERE EMPLOYEE\_ID IN (

SELECT DISTINCT MANAGER\_ID

FROM EMPLOYEES

)

;

--재귀참조

SELECT DISTINCT MANAGER.\*

FROM EMPLOYEES MANAGER

, EMPLOYEES E

WHERE MANAGER.EMPLOYEE\_ID = E.MANAGER\_ID

;

-- 34. 이름(FIRST\_NAME)이 C로 시작하고 s로 끝나는 사원의 부서명, 직무명, 도시명, 국가명, 대륙명을 조회하라.

SELECT E.FIRST\_NAME

, D.DEPARTMENT\_NAME

, J.JOB\_TITLE

, L.CITY

, C.COUNTRY\_NAME

, R.REGION\_NAME

FROM EMPLOYEES E

, DEPARTMENTS D

, JOBS J

, LOCATIONS L

, COUNTRIES C

, REGIONS R

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

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

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

AND C.COUNTRY\_ID = L.COUNTRY\_ID

AND R.REGION\_ID = C.REGION\_ID

AND E.FIRST\_NAME LIKE 'C%s'

;

-- 35. 직무를 전환한 이력이 없는 사원의 모든 정보를 조회하라.

SELECT \*

FROM EMPLOYEES

WHERE EMPLOYEE\_ID NOT IN (

SELECT EMPLOYEE\_ID

FROM JOB\_HISTORY

)

;

-- 36. 150번 사원보다 늦게 입사한 사원 중 150번 사원보다 더 많은 연봉을 받는 사원을 조회하라.

SELECT \*

FROM (

SELECT \*

FROM EMPLOYEES

WHERE HIRE\_DATE > (

SELECT HIRE\_DATE

FROM EMPLOYEES

WHERE EMPLOYEE\_ID = 150

)

)

WHERE SALARY > (

SELECT SALARY

FROM EMPLOYEES

WHERE EMPLOYEE\_ID = 150

)

;

SELECT E.\*

FROM (

SELECT \*

FROM EMPLOYEES

WHERE EMPLOYEE\_ID = 150

) A,

EMPLOYEES E

WHERE E.HIRE\_DATE > A.HIRE\_DATE

AND E.SALARY > A.SALARY

;

-- 37. 자신의 상사가 자신 보다 늦게 입사한 사원의 모든 정보를 조회하라.

--재귀참조

SELECT E.\*

FROM EMPLOYEES E

, EMPLOYEES MANAGER

WHERE E.MANAGER\_ID = MANAGER.EMPLOYEE\_ID

AND E.HIRE\_DATE < MANAGER.HIRE\_DATE

;

-- 38. 100번 사원을 직속 상사로 두고 있는 사원들의 모든 정보를 조회하라.

SELECT \*

FROM EMPLOYEES

WHERE MANAGER\_ID = 100

;

-- 39. 100번 사원을 상사로 두고 있는 모든 사원들의 모든 정보를 조회하라.

--계층쿼리

SELECT LEVEL -- 번호

, E.\*

FROM EMPLOYEES E

START WITH E.MANAGER\_ID = 100 -- 어디부터 시작할텐가?

CONNECT BY PRIOR E.EMPLOYEE\_ID = E.MANAGER\_ID -- 우선순위

;

--가장 깊은 LEVELSELECT MAX(LEVEL)

FROM EMPLOYEES E

START WITH E.MANAGER\_ID = 100

CONNECT BY PRIOR E.EMPLOYEE\_ID = E.MANAGER\_ID

;

--112번 사원의 모든 상사를 조회하라.

SELECT LEVEL

, E.\*

FROM EMPLOYEES E

START WITH E.EMPLOYEE\_ID = 112

CONNECT BY PRIOR E.MANAGER\_ID = E.EMPLOYEE\_ID

;

-- 150번 사원의 모든 상사들의 이름(FIRST\_NAME)과 부서명을 조회하라.

SELECT LEVEL

, E.FIRST\_NAME

, D.DEPARTMENT\_NAME

, E.EMPLOYEE\_ID

, E.MANAGER\_ID

FROM EMPLOYEES E

, DEPARTMENTS D

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

START WITH E.EMPLOYEE\_ID = 150 -- 150번 사원 기준

CONNECT BY PRIOR E.MANAGER\_ID = E.EMPLOYEE\_ID -- 150번 사원의 MANAGER\_ID는 다른 사원의 EMPLOYEE\_ID이다

;

-- 40. 사원수가 가장 많은 도시의 이름을 조회하라.

SELECT \*

FROM (

SELECT L.CITY

, COUNT(1)

FROM EMPLOYEES E

, DEPARTMENTS D

, LOCATIONS L

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

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

GROUP BY L.CITY

ORDER BY COUNT(1) DESC

)

WHERE ROWNUM = 1

;

-- 41. 직무의 종류가 가장 많은 부서의 이름을 조회하라.

SELECT \*

FROM (

SELECT DEPARTMENT\_NAME

, COUNT(DISTINCT JOB\_ID)

FROM EMPLOYEES E

, DEPARTMENTS D

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

GROUP BY DEPARTMENT\_NAME

ORDER BY COUNT(DISTINCT JOB\_ID) DESC

)

WHERE ROWNUM = 1

;

--강사님

SELECT DEPARTMENT\_NAME

FROM (

SELECT DEPARTMENT\_NAME

, COUNT(1) CNT

FROM (

SELECT D.DEPARTMENT\_NAME

, E.JOB\_ID

FROM EMPLOYEES E

, DEPARTMENTS D

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

GROUP BY D.DEPARTMENT\_NAME

, E.JOB\_ID

)

GROUP BY DEPARTMENT\_NAME

ORDER BY CNT DESC

)

WHERE ROWNUM = 1

;

-- 42. 담당 직무의 최대 연봉을 받고 있는 사원들의 모든 정보를 조회하라.

SELECT \*

FROM EMPLOYEES E

, (

SELECT JOB\_ID

, MAX(SALARY) MAX\_SALARY

FROM EMPLOYEES

GROUP BY JOB\_ID

) A

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

AND E.SALARY = A.MAX\_SALARY

;

SELECT E.\*

FROM EMPLOYEES E

, JOBS J

WHERE E.JOB\_ID = J.JOB\_ID

AND E.SALARY = J.MAX\_SALARY

;

-- 43. 담당 직무의 최소 연봉을 받고 있는 사원들의 모든 정보를 조회하라.

SELECT \*

FROM EMPLOYEES E

, (

SELECT JOB\_ID

, MIN(SALARY) MIN\_SALARY

FROM EMPLOYEES

GROUP BY JOB\_ID

) A

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

AND E.SALARY = A.MIN\_SALARY

;

SELECT E.\*

FROM EMPLOYEES E

, JOBS J

WHERE E.JOB\_ID = J.JOB\_ID

AND E.SALARY = J.MIN\_SALARY

;

-- 44. 부서와 직무별 최대연봉, 최소연봉, 사원수를 조회하라.

SELECT D.DEPARTMENT\_NAME

, J.JOB\_TITLE

, MAX(SALARY) , MIN(SALARY)

, COUNT(1)

FROM EMPLOYEES E

, DEPARTMENTS D

, JOBS J

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

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

GROUP BY D.DEPARTMENT\_NAME

, J.JOB\_TITLE

;

-- 45. 사원수가 가장 많은 도시에서 근무하는 모든 사원들의 연봉 총합을 조회하라.

SELECT SUM\_SAL

FROM (

SELECT L.CITY

, SUM(SALARY) SUM\_SAL

, COUNT(1)

FROM EMPLOYEES E

, DEPARTMENTS D

, LOCATIONS L

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

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

GROUP BY L.CITY

ORDER BY COUNT(1) DESC

)

WHERE ROWNUM = 1

;