1. SQL Developer 실행 및 DB 접속
2. 교재 준비
3. 공유 폴더 접근

------------------------------------------------------------------------------

SELECT \*

FROM TID ;

SELECT \*

FROM TACCT ;

문법에 따른 구분

- ORACLE JOIN : 테이블을 쉼표(,)로 구분, WHERE 절에 조인 조건 정의

- ANSI JOIN : 테이블을 키워드(JOIN)로 구분, ON 절에 조인 조건 정의

SELECT \* FROM EMP ;

SELECT \* FROM DEPT ;

SELECT \*

FROM EMP, DEPT

ORDER BY 1 ;

SELECT \*

FROM EMP , DEPT

WHERE EMP.DEPTNO = DEPT.DEPTNO ;

-- ORACLE JOIN

SELECT \*

FROM TID I

,TACCT A

WHERE I.LNID = A.LNID ;

-- ANSI JOIN

SELECT \*

FROM TID A

JOIN TACCT B

ON A.LNID = B.LNID ;

---------------------------------------------------

SELECT \* FROM TREPAY ;

SELECT \* FROM TREPAY ORDER BY 1, 2, 3 ;

-- 잘못된 조인

SELECT B.LNACT, B.LNACT\_SEQ, B.LN\_DT, B.LN\_AMT, C.\*

FROM TACCT B

JOIN TREPAY C

ON B.LNACT = C.LNACT

ORDER BY B.LNACT, B.LNACT\_SEQ, C.SEQ ;

SELECT \*

FROM TACCT B

JOIN TREPAY C

ON B.LNACT = C.LNACT

AND B.LNACT\_SEQ = C.LNACT\_SEQ ;

SELECT \*

FROM TACCT B

,TREPAY C

WHERE B.LNACT = C.LNACT

AND B.LNACT\_SEQ = C.LNACT\_SEQ ;

---------------------------------------------

SELECT \*

FROM TID A

,TACCT B

,TREPAY C

WHERE A.LNID = B.LNID

AND B.LNACT = C.LNACT

AND B.LNACT\_SEQ = C.LNACT\_SEQ ;

SELECT \*

FROM TID A

JOIN TACCT B

ON A.LNID = B.LNID

JOIN TREPAY C

ON B.LNACT = C.LNACT

AND B.LNACT\_SEQ = C.LNACT\_SEQ ;

SELECT \* -- ERROR

FROM TID A

JOIN TACCT B

JOIN TREPAY C

ON A.LNID = B.LNID

AND B.LNACT = C.LNACT

AND B.LNACT\_SEQ = C.LNACT\_SEQ ;

--------------------------------------------------

SELECT \*

FROM EMPLOYEES ;

SELECT \*

FROM DEPARTMENTS ;

SELECT \*

FROM EMPLOYEES E

,DEPARTMENTS D

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

SELECT \*

FROM EMPLOYEES E

JOIN DEPARTMENTS D

ON E.DEPARTMENT\_ID = D.DEPARTMENT\_ID ;

-----------------------------------------------------------

SELECT \*

FROM EMPLOYEES E INNER JOIN DEPARTMENTS D

ON E.DEPARTMENT\_ID = D.DEPARTMENT\_ID ;

SELECT \*

FROM EMPLOYEES E LEFT OUTER JOIN DEPARTMENTS D

ON E.DEPARTMENT\_ID = D.DEPARTMENT\_ID ;

SELECT \*

FROM EMPLOYEES E RIGHT OUTER JOIN DEPARTMENTS D

ON E.DEPARTMENT\_ID = D.DEPARTMENT\_ID ;

SELECT \*

FROM EMPLOYEES E FULL OUTER JOIN DEPARTMENTS D

ON E.DEPARTMENT\_ID = D.DEPARTMENT\_ID ;

---------------------------------------------------------------

SELECT \*

FROM EMPLOYEES E LEFT OUTER JOIN DEPARTMENTS D

ON E.DEPARTMENT\_ID = D.DEPARTMENT\_ID ;

SELECT \*

FROM EMPLOYEES E

,DEPARTMENTS D

WHERE E.DEPARTMENT\_ID = D.DEPARTMENT\_ID (+) ;

SELECT \*

FROM EMPLOYEES E RIGHT OUTER JOIN DEPARTMENTS D

ON E.DEPARTMENT\_ID = D.DEPARTMENT\_ID ;

SELECT \*

FROM EMPLOYEES E

,DEPARTMENTS D

WHERE E.DEPARTMENT\_ID (+) = D.DEPARTMENT\_ID ;

SELECT \* -- ERROR

FROM EMPLOYEES E

,DEPARTMENTS D

WHERE E.DEPARTMENT\_ID (+) = D.DEPARTMENT\_ID (+) ;

SELECT \*

FROM EMP, DEPT ;

SELECT \*

FROM EMP CROSS JOIN DEPT ;

---------------------------------------------------------------------

SELECT employee\_id, last\_name, salary, job\_id, department\_id

FROM employees

WHERE salary > (SELECT salary

FROM employees

WHERE employee\_id = 107) ;

SELECT employee\_id, last\_name, salary, job\_id, department\_id -- ERROR

FROM employees

WHERE salary > (SELECT salary, DEPARTMENT\_ID

FROM employees

WHERE employee\_id = 107) ;

SELECT employee\_id, last\_name, salary, job\_id, department\_id

FROM employees

WHERE salary > (SELECT salary

FROM employees

WHERE employee\_id = 107) ;

SELECT employee\_id, last\_name, salary, job\_id, department\_id -- ERROR

FROM employees

WHERE salary = (SELECT MIN(salary)

FROM employees

GROUP BY DEPARTMENT\_ID) ;

SELECT employee\_id, last\_name, salary, job\_id, department\_id

FROM employees

WHERE salary > (SELECT AVG(salary)

FROM employees) ;

SELECT department\_id, SUM(salary)

FROM employees

GROUP BY department\_id

HAVING SUM(salary) > (SELECT AVG(SUM(salary))

FROM employees

GROUP BY department\_id);

SELECT employee\_id, last\_name, salary, job\_id, department\_id

FROM employees

WHERE salary IN (SELECT MIN(salary)

FROM employees

GROUP BY department\_id) ;

SELECT employee\_id, last\_name, salary, job\_id, department\_id

FROM employees

WHERE salary >ANY (SELECT AVG(salary)

FROM employees

GROUP BY department\_id) ;

SELECT \*

FROM EMP E

WHERE SAL > 2000 ;

SELECT \*

FROM EMP E

WHERE E.SAL > 2000 ;

SELECT \* -- ERROR

FROM EMP E

WHERE EMP.SAL > 2000 ;

SELECT e.employee\_id, e.last\_name, e.salary, e.job\_id, e.department\_id, a.avg\_sal

FROM employees e

JOIN (SELECT department\_id, AVG(salary) AS AVG\_SAL

FROM employees

GROUP BY department\_id) a

ON e.department\_id = a.department\_id ;

SELECT e.employee\_id, e.last\_name, e.salary,

e.job\_id, e.department\_id, a.avg\_sal

FROM employees e

JOIN (SELECT department\_id, AVG(salary) AS AVG\_SAL

FROM employees

GROUP BY department\_id) a

ON e.department\_id = a.department\_id

AND E.SALARY > A.AVG\_SAL ;

----------------------------------------------------

SELECT employee\_id, last\_name, salary, job\_id, department\_id

FROM employees

WHERE (SELECT AVG(salary)

FROM employees) < salary ;

--------------------------------------------------------

-- SELF JOIN

SELECT E.EMPLOYEE\_ID, E.LAST\_NAME, E.SALARY, E.JOB\_ID, E.DEPARTMENT\_ID,

S.EMPLOYEE\_ID, S.SALARY

FROM EMPLOYEES E

JOIN EMPLOYEES S

ON E.SALARY > S.SALARY

WHERE S.EMPLOYEE\_ID = 107 ;

SELECT E.EMPLOYEE\_ID, E.LAST\_NAME, E.SALARY, E.JOB\_ID, E.DEPARTMENT\_ID,

S.EMPLOYEE\_ID, S.SALARY

FROM EMPLOYEES E

JOIN (SELECT \*

FROM EMPLOYEES

WHERE EMPLOYEE\_ID = 107) S

ON E.SALARY > S.SALARY ;

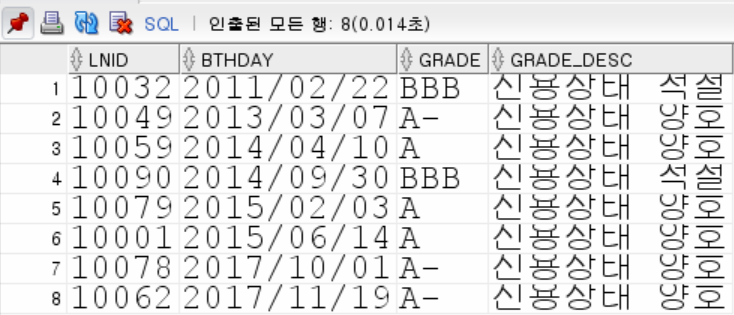
1. **2010년 이후 설립된 '신용상태 양호','신용상태 적절' 등급의 법인 차주 정보를 검색하세요.**

- TID(차주정보), TCODE(신용등급 정보) 테이블 사용

- 등급 설명(TCODE.GRADE\_DESC)을 함께 검색

- 설립일(BTHDAY)을 기준으로 오름차순 정렬

**검색 결과**



SELECT A.LNID, A.BTHDAY, A.GRADE, B.GRADE\_DESC

FROM TID A

JOIN TCODE B

ON A.GRADE = B.GRADE

WHERE A.BTHDAY >= '2010/01/01'

AND B.GRADE\_DESC in ('신용상태 양호','신용상태 적절')

ORDER BY A.BTHDAY;

SELECT A.LNID, A.BTHDAY, A.GRADE, B.GRADE\_DESC

FROM TID A

,TCODE B

WHERE A.GRADE = B.GRADE

AND A.BTHDAY >= '2010/01/01'

AND B.GRADE\_DESC in ('신용상태 양호','신용상태 적절')

ORDER BY A.BTHDAY;

**SELECT A.LNID, A.BTHDAY, A.GRADE, B.GRADE\_DESC**

**FROM TID A**

**JOIN (SELECT GRADE, GRADE\_DESC**

**FROM TCODE) B**

**on A.GRADE = B.GRADE**

**WHERE A.BTHDAY > '20100101'**

**and B.GRADE\_DESC in ('신용상태 양호','신용상태 적절')**

**ORDER BY A.BTHDAY;**

1. **다음 요구 조건에 만족하는 행을 검색하세요.**

- TACCT(대출계좌), TCREDIT(신용평가정보), TCODE(신용등급 코드) 테이블을 사용

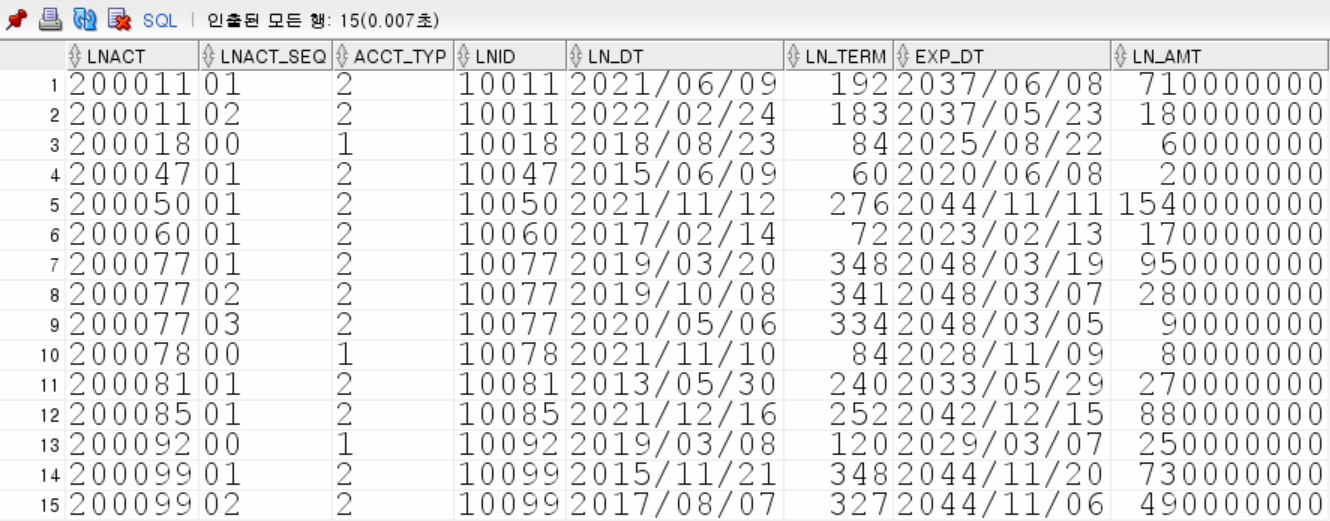
- 한도 계좌 제외 검색 (TACCT.LMT\_TYP:NULL)

- '04'(TCREDIT.ACODE) 평가사에서 '신용상태 우수'(TCODE.GRADE\_DESC) 등급의 신용 평가를

받은 차주 검색

- 신용평가 만료일(TCREDIT.END\_DT)이 오늘 이후인 평가만 사용

- 검색 결과는 계좌번호, 계좌일련번호를 기준으로 오름차순 정렬



SELECT a.lnact, a.lnact\_seq, a.acct\_typ, a.lnid, a.ln\_dt, a.ln\_term,

a.exp\_dt, a.ln\_amt

FROM tacct a

JOIN tcredit b

ON a.lnid = b.lnid

JOIN tcode c

ON b.code = c.code

WHERE a.lmt\_typ IS NULL

AND b.acode = '04'

AND b.end\_dt > SYSDATE

AND c.grade\_desc = '신용상태 우수'

ORDER BY a.lnact, a.lnact\_seq ;

SELECT A.LNACT, A.LNACT\_SEQ, A.ACCT\_TYP, A.LNID, A.LN\_dT,

A.LN\_TERM, A.EXP\_DT, A.LN\_AMT

FROM TACCT A

JOIN (SELECT LNID, CODE, ACODE, END\_DT

FROM TCREDIT) B ON A.LNID = B.LNID

JOIN (SELECT CODE, GRADE, GRADE\_DESC

FROM TCODE) C ON B.CODE = C.CODE

WHERE A.LMT\_TYP IS NULL

AND B.ACODE = '04'

AND C.GRADE\_DESC = '신용상태 우수'

AND B.END\_DT >= SYSDATE

ORDER BY A.LNACT, A.LNACT\_SEQ;

SELECT A.LNACT, A.LNACT\_SEQ, A.ACCT\_TYP,

A.LNID, A.LN\_DT, A.LN\_TERM, A.EXP\_DT, A.LN\_AMT

FROM TACCT A JOIN TCREDIT B

ON A.LNID = B.LNID JOIN TCODE C

ON B.CODE = C.CODE

WHERE A.LMT\_TYP IS NULL

AND B.ACODE = '04'

AND B.END\_DT >= '20220514'

AND C.GRADE\_DESC = '신용상태 우수'

ORDER BY A.LNID, A.LNACT\_SEQ;

select t.lnact, t.lnact\_seq, t.acct\_typ, tc.lnid, ln\_dt, ln\_term, exp\_dt, ln\_amt

from tacct t

join tcredit tc

on t.lnid = tc.lnid

join tcode tco

on tc.code = tco.code

where t.lmt\_typ is null

and tc.acode = '04'

and tco.grade\_desc = '신용상태 우수'

and tc.end\_dt > sysdate

order by t.lnact, t.lnact\_seq;

**답안**

SELECT lnact, lnact\_seq, acct\_typ, lnid, ln\_dt, ln\_term, exp\_dt, ln\_amt

FROM tacct

WHERE lmt\_typ IS NULL

AND lnid IN (SELECT lnid

FROM tcredit

WHERE acode = '04'

AND end\_dt > SYSDATE

AND code IN (SELECT code

FROM tcode

WHERE grade\_desc = '신용상태 우수'))

ORDER BY lnact, lnact\_seq ;