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

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

SQL

* QUERY: SELECT
* DML : INSERT, UPDATE, DELETE
* TCL : COMMIT, ROLLBACK
* DDL : CREATE, DROP, ALTER, TRUNCATE
* DCL : GRANT, REVOKE

5 SELECT : 검색 대상의 표현식 (컬럼 이름)

1 FROM : 검색 대상의 집합 (테이블 이름)

2 WHERE : 행 제한을 위한 조건식

3 GROUP BY

4 HAVING

6 ORDER BY : 정렬

SELECT \*

FROM TID

WHERE GENDER = '1' ;

SELECT \*

FROM TID

WHERE GENDER = '1'

ORDER BY SCORE, LNID ;

SELECT \*

FROM TID

WHERE GENDER = '1'

ORDER BY SCORE, BTHDAY ;

SELECT \*

FROM TID

WHERE GENDER = '1'

ORDER BY SCORE DESC, BTHDAY ;

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

SELECT lnid, bthday, grade

,CASE grade WHEN 'AA+' THEN 1

WHEN 'AA' THEN 2

WHEN 'AA-' THEN 3

END

FROM tid

WHERE id\_typ = '2'

AND grade IN ('AA+','AA','AA-')

ORDER BY CASE grade WHEN 'AA+' THEN 1

WHEN 'AA' THEN 2

WHEN 'AA-' THEN 3

END

,lnid ;

SELECT lnid, bthday, gender, score ,TO\_CHAR(bthday, 'YYYY')

FROM tid

WHERE id\_typ = '1'

ORDER BY TO\_CHAR(bthday, 'MMDD');

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

SELECT \*

FROM TID

WHERE GRADE IN ('A','a') ;

SELECT \*

FROM TID

WHERE UPPER(GRADE) = 'A' ;

SELECT CODE, GRADE, GRADE\_DESC

,LOWER(GRADE)

,SUBSTR(GRADE\_DESC,1,4)

,LENGTH(GRADE\_DESC)

FROM TCODE

WHERE GRADE\_DESC LIKE '신용상태%' ;

SELECT LNACT, LNACT\_SEQ, LNID, (LN\_AMT \* RATE)/365

FROM TACCT

WHERE LMT\_TYP IS NULL ;

SELECT LNACT, LNACT\_SEQ, LNID

,ROUND((LN\_AMT \* RATE)/365,2)

,ROUND((LN\_AMT \* RATE)/365,0)

,ROUND((LN\_AMT \* RATE)/365,-1)

FROM TACCT

WHERE LMT\_TYP IS NULL ;

SELECT LNACT, LNACT\_SEQ, LNID

,TRUNC((LN\_AMT \* RATE)/365,2)

,TRUNC((LN\_AMT \* RATE)/365,0)

,TRUNC((LN\_AMT \* RATE)/365,-1)

FROM TACCT

WHERE LMT\_TYP IS NULL ;

SELECT LNACT, LNACT\_SEQ, LNID, LN\_DT, LN\_AMT

FROM TACCT

WHERE LMT\_TYP IS NULL ;

SELECT LNACT, LNACT\_SEQ, LNID, LN\_DT, LN\_DT + 30, LN\_AMT

FROM TACCT

WHERE LMT\_TYP IS NULL ;

SELECT LNACT, LNACT\_SEQ, LNID, LN\_DT, LN\_DT + 30, ADD\_MONTHS(LN\_DT,1), LN\_AMT

FROM TACCT

WHERE LMT\_TYP IS NULL ;

SELECT LNACT, LNACT\_SEQ, LNID, LN\_DT, LN\_DT - 30, ADD\_MONTHS(LN\_DT,-1), LN\_AMT

FROM TACCT

WHERE LMT\_TYP IS NULL ;

SELECT LNACT, LNACT\_SEQ, LNID, LN\_DT, SYSDATE , LN\_AMT

FROM TACCT

WHERE LMT\_TYP IS NULL ;

SELECT LNACT, LNACT\_SEQ, LNID, LN\_DT, SYSDATE , SYSDATE - LN\_DT, LN\_AMT

FROM TACCT

WHERE LMT\_TYP IS NULL ;

SELECT LNACT, LNACT\_SEQ, LNID, LN\_DT, SYSDATE , (SYSDATE - LN\_DT)/30, LN\_AMT

FROM TACCT

WHERE LMT\_TYP IS NULL ;

SELECT LNACT, LNACT\_SEQ, LNID, LN\_DT, SYSDATE

,MONTHS\_BETWEEN(SYSDATE,LN\_DT), LN\_AMT

FROM TACCT

WHERE LMT\_TYP IS NULL ;

SELECT LNACT, LNACT\_SEQ, LNID, LN\_DT, SYSDATE

,TRUNC(MONTHS\_BETWEEN(SYSDATE,LN\_DT)), LN\_AMT

FROM TACCT

WHERE LMT\_TYP IS NULL ;

SELECT LNACT, LNACT\_SEQ, LNID, LN\_AMT, LN\_DT

,TO\_CHAR(LN\_DT,'YYYY')

,TO\_CHAR(LN\_DT,'MM/DD')

,TO\_CHAR(LN\_DT,'DAY')

,TO\_CHAR(LN\_DT,'Q')

FROM TACCT

WHERE LMT\_TYP IS NULL ;

SELECT LNACT, LNACT\_SEQ, LNID, LN\_AMT, LN\_DT

,TO\_CHAR(LN\_DT,'YYYY')

,TO\_CHAR(LN\_DT,'MM/DD')

,TO\_CHAR(LN\_DT,'DAY')

,TO\_CHAR(LN\_DT,'Q')

,TO\_CHAR(LN\_DT,'D')

FROM TACCT

WHERE LMT\_TYP IS NULL

AND TO\_CHAR(LN\_DT,'D') IN ('1','7');

SELECT LNACT, LNACT\_SEQ, LNID, LN\_AMT, LN\_DT

,TO\_CHAR(LN\_DT,'YYYY')

,TO\_CHAR(LN\_DT,'MM/DD')

,TO\_CHAR(LN\_DT,'DAY')

,TO\_CHAR(LN\_DT,'Q')

,TO\_CHAR(LN\_DT,'D')

FROM TACCT

WHERE LMT\_TYP IS NULL

AND TO\_CHAR(LN\_DT,'D') NOT IN ('1','7');

SELECT LNACT, LNACT\_SEQ, LNID, LN\_AMT

,TO\_CHAR(LN\_AMT,'L999,999,999,999')

,TO\_CHAR(LN\_AMT,'$999,999,999,999')

FROM TACCT

WHERE LMT\_TYP IS NULL ;

SELECT LNACT, LNACT\_SEQ, LNID, LN\_AMT

,TO\_CHAR(LN\_AMT,'L000,000,999,999')

,TO\_CHAR(LN\_AMT,'L999,999,999,999')

FROM TACCT

WHERE LMT\_TYP IS NULL ;

SELECT LNACT, LNACT\_SEQ, LNID, LN\_AMT, RATE

,(LN\_AMT \* RATE)/365

FROM TACCT ;

SELECT LNACT, LNACT\_SEQ, LNID, LN\_AMT, RATE, NVL(RATE,0)

,(LN\_AMT \* RATE)/365

FROM TACCT ;

SELECT lnact, lnact\_seq, acct\_typ, lnid, ln\_amt, rate

,TRUNC((ln\_amt \* rate) / 12) AS 이자1

,TRUNC((ln\_amt \* NVL(rate,0)) / 12) AS 이자2

FROM tacct

WHERE branch = '10';

SELECT lnid, bthday, score, gender,

CASE gender WHEN '1' THEN '남'

WHEN '2' THEN '여'

ELSE '법인'

END AS 성별

FROM tid

WHERE id\_typ = '1';

SELECT lnact, lnact\_seq, ln\_amt, rate, rate\_typ,

CASE WHEN rate\_typ = '1'

THEN TRUNC((ln\_amt \* rate)/12)

WHEN rate\_typ = '2'

THEN TRUNC((ln\_amt \* (rate+0.01))/12)

ELSE 0

END AS 이자

FROM tacct

WHERE lmt\_typ IS NULL ;

select \* from v$version ;

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

**6장**

SELECT SUM(ln\_amt), AVG(ln\_amt),

MAX(ln\_amt), MIN(ln\_amt)

FROM tacct

WHERE lmt\_typ IS NULL ;

SELECT MAX(ln\_dt), MIN(ln\_dt)

FROM tacct

WHERE lmt\_typ IS NULL ;

SELECT COUNT(\*)

FROM tid ;

SELECT COUNT(gender)

FROM tid ;

SELECT COUNT(DISTINCT gender)

FROM tid ;

SELECT SUM(LN\_AMT), SUM(DISTINCT LN\_AMT)

FROM TACCT ;

5 SELECT : 검색 대상의 표현식 (컬럼 이름)

1 FROM : 검색 대상의 집합 (테이블 이름)

2 WHERE : 행 제한을 위한 조건식

3 GROUP BY : 그룹 생성을 위한 표현식

4 HAVING : 그룹 제한을 위한 조건식

6 ORDER BY : 정렬을 위한 표현식

SELECT branch, ln\_amt

FROM tacct

WHERE lmt\_typ IS NULL

ORDER BY BRANCH ;

SELECT branch, SUM(ln\_amt)

FROM tacct

WHERE lmt\_typ IS NULL

GROUP BY branch;

SELECT SUM(ln\_amt)

FROM tacct

WHERE lmt\_typ IS NULL

GROUP BY branch;

SELECT branch, prod\_cd, SUM(ln\_amt) -- ERROR

FROM tacct ;

SELECT branch, prod\_cd, SUM(ln\_amt) -- ERROR

FROM tacct

GROUP BY branch ;

SELECT PROD\_CD, SUM(LN\_AMT)

FROM TACCT

WHERE LMT\_TYP IS NULL

GROUP BY PROD\_CD ;

SELECT branch, SUM(ln\_amt)

FROM tacct

WHERE lmt\_typ IS NULL

GROUP BY branch;

SELECT branch, prod\_cd, ln\_amt

FROM tacct

WHERE lmt\_typ IS NULL

ORDER BY branch, prod\_cd ;

SELECT branch, prod\_cd, SUM(ln\_amt)

FROM tacct

WHERE lmt\_typ IS NULL

GROUP BY branch, prod\_cd ;

SELECT TO\_CHAR(LN\_DT,'YYYY'), BRANCH, SUM(LN\_AMT)

FROM TACCT

WHERE LMT\_TYP IS NULL

GROUP BY TO\_CHAR(LN\_DT,'YYYY'), BRANCH ;

SELECT TO\_CHAR(LN\_DT,'YYYY') AS 연도

,BRANCH AS 지점

,SUM(LN\_AMT) AS 대출금액합계

FROM TACCT

WHERE LMT\_TYP IS NULL

GROUP BY TO\_CHAR(LN\_DT,'YYYY'), BRANCH

ORDER BY TO\_CHAR(LN\_DT,'YYYY'), BRANCH ;

SELECT TO\_CHAR(LN\_DT,'YYYYMM') AS 연월

,BRANCH AS 지점

,SUM(LN\_AMT) AS 대출금액합계

FROM TACCT

WHERE LMT\_TYP IS NULL

GROUP BY TO\_CHAR(LN\_DT,'YYYYMM'), BRANCH

ORDER BY TO\_CHAR(LN\_DT,'YYYYMM'), BRANCH ;

SELECT branch, prod\_cd, SUM(ln\_amt) -- ERROR

FROM tacct

WHERE lmt\_typ IS NULL

AND SUM(ln\_amt) > 2000000000

GROUP BY branch, prod\_cd ;

SELECT branch, prod\_cd, SUM(ln\_amt)

FROM tacct

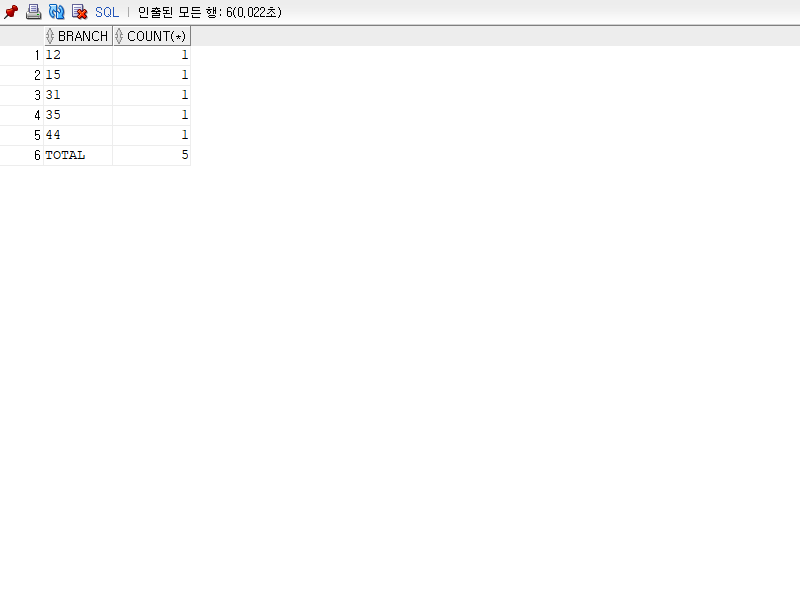
WHERE lmt\_typ IS NULL

GROUP BY branch, prod\_cd

HAVING SUM(ln\_amt) > 2000000000 ;

1. **대출 계좌정보(TACCT) 테이블을 이용하여 다음의 데이터를 검색하세요.**

* 검색: 지점(BRANCH)별 연체 계좌 수와 전체 연체 계좌 수 검색
* 조건1: 한도 계좌 제외 (LMT\_TYP:NULL)
* 조건2: 연체 중인 계좌 (DLQ\_CNT > 0, DLQ\_DT IS NOT NULL )
* 그룹1: 지점(BRANCH)별 연체 계좌 수
* 그룹2: 전체 연체 계좌 수
* 정렬: 지점(BRANCH) 오름차순



**답안1**

SELECT BRANCH, COUNT(\*)

FROM TACCT

WHERE LMT\_TYP IS NULL

AND DLQ\_DT IS NOT NULL

GROUP BY BRANCH

**UNION ALL**

SELECT NULL, COUNT(\*)

FROM TACCT

WHERE LMT\_TYP IS NULL

AND DLQ\_DT IS NOT NULL

ORDER BY 1 ;

**답안2**

SELECT NVL(BRANCH,'TOTAL') AS BRANCH, COUNT(\*)

FROM TACCT

WHERE LMT\_TYP IS NULL

AND DLQ\_DT IS NOT NULL

GROUP BY **ROLLUP**(BRANCH)

ORDER BY BRANCH ;

select decode(grouping(branch),1,'total',branch) as branch, count(\*)

from tacct

where dlq\_cnt > 0

group by rollup(branch);

SELECT branch, prod\_cd, SUM(ln\_amt)

FROM tacct

WHERE lmt\_typ IS NULL

GROUP BY ROLLUP(branch, prod\_cd) ;

1. **대출 계좌정보(TACCT) 테이블을 이용하여 다음의 데이터를 검색하세요.**

* 검색: 지점별, 월별(2022년 1월~3월) 개설 계좌 수를 피벗 테이블로 검색
* 조건1: 한도 계좌 제외 (LMT\_TYP:NULL)
* 조건2: 대출 일자(LN\_DT)가 2022년 1월부터 3월까지
* 그룹: 지점(BRANCH)별, 월별
* 정렬: 지점(BRANCH) 오름차순
* 힌트: COUNT(CASE 문)

SELECT BRANCH

,TO\_CHAR(LN\_DT,'MM')

,COUNT(\*)

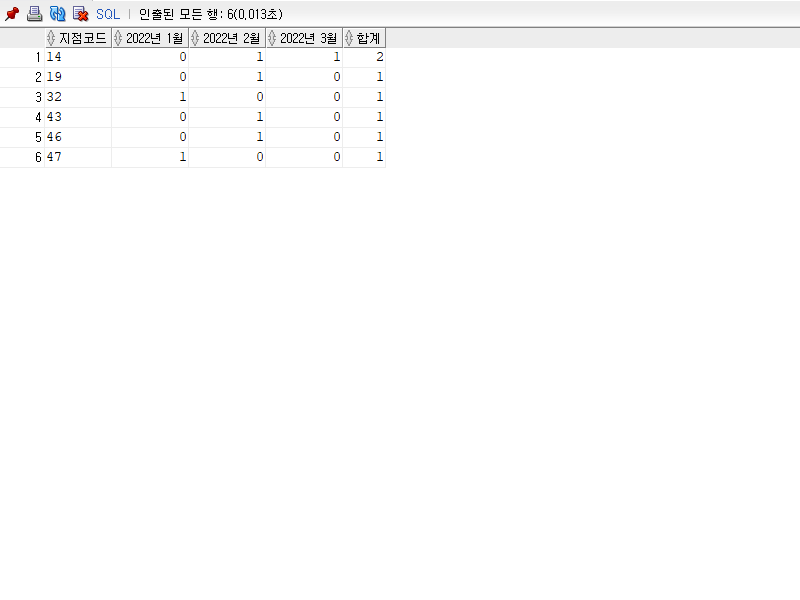
FROM TACCT

WHERE LMT\_TYP IS NULL

AND LN\_DT BETWEEN '2022/01/01' AND '2022/03/31'

GROUP BY BRANCH, TO\_CHAR(LN\_DT,'MM')

ORDER BY 1, 2 ;



SELECT BRANCH

,TO\_CHAR(LN\_DT,'MM')

,CASE TO\_CHAR(LN\_DT,'MM') WHEN '01' THEN LNACT END

,CASE TO\_CHAR(LN\_DT,'MM') WHEN '02' THEN LNACT END

,CASE TO\_CHAR(LN\_DT,'MM') WHEN '03' THEN LNACT END

FROM TACCT

WHERE LMT\_TYP IS NULL

AND LN\_DT BETWEEN '2022/01/01' AND '2022/03/31' ;

SELECT BRANCH

,COUNT(CASE TO\_CHAR(LN\_DT,'MM') WHEN '01' THEN LNACT END) AS "1월"

,COUNT(CASE TO\_CHAR(LN\_DT,'MM') WHEN '02' THEN LNACT END) AS "2월"

,COUNT(CASE TO\_CHAR(LN\_DT,'MM') WHEN '03' THEN LNACT END) AS "3월"

,COUNT(\*)

FROM TACCT

WHERE LMT\_TYP IS NULL

AND LN\_DT BETWEEN '2022/01/01' AND '2022/03/31'

GROUP BY BRANCH

ORDER BY 1 ;

SELECT BRANCH

,SUM(CASE TO\_CHAR(LN\_DT,'MM') WHEN '01' THEN LN\_AMT END) AS "1월"

,SUM(CASE TO\_CHAR(LN\_DT,'MM') WHEN '02' THEN LN\_AMT END) AS "2월"

,SUM(CASE TO\_CHAR(LN\_DT,'MM') WHEN '03' THEN LN\_AMT END) AS "3월"

,SUM(LN\_AMT)

FROM TACCT

WHERE LMT\_TYP IS NULL

AND LN\_DT BETWEEN '2022/01/01' AND '2022/03/31'

GROUP BY BRANCH

ORDER BY 1 ;

SELECT BRANCH

,NVL(SUM(CASE TO\_CHAR(LN\_DT,'MM') WHEN '01' THEN LN\_AMT END),0) AS "1월"

,NVL(SUM(CASE TO\_CHAR(LN\_DT,'MM') WHEN '02' THEN LN\_AMT END),0) AS "2월"

,NVL(SUM(CASE TO\_CHAR(LN\_DT,'MM') WHEN '03' THEN LN\_AMT END),0) AS "3월"

,SUM(LN\_AMT)

FROM TACCT

WHERE LMT\_TYP IS NULL

AND LN\_DT BETWEEN '2022/01/01' AND '2022/03/31'

GROUP BY ROLLUP(BRANCH)

ORDER BY 1 ;

SELECT branch AS "지점코드"

,NVL("1월",0) AS "2022년 1월"

,NVL("2월",0) AS "2022년 2월"

,NVL("3월",0) AS "2022년 3월"

,NVL("1월",0)+NVL("2월",0)+NVL("3월",0) AS "합계"

FROM (SELECT branch, TO\_CHAR(ln\_dt,'YYYYMM') AS YYYYMM, COUNT(\*) AS CNT

FROM tacct

WHERE lmt\_typ IS NULL

AND ln\_dt BETWEEN '2022/01/01' AND '2022/03/31'

GROUP BY branch, TO\_CHAR(ln\_dt,'YYYYMM'))

PIVOT (SUM(cnt) FOR yyyymm IN ('202201' AS "1월"

,'202202' AS "2월"

,'202203' AS "3월"))

ORDER BY 1;

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

7. JOIN

둘 이상의 집합의 데이터를 동시에 검색 !!

SELECT \*

FROM TID

JOIN TACCT

ON TID.LNID = TACCT.LNID ;

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

SELECT \*

FROM TID

WHERE LNID = '10000' ;

SELECT \*

FROM TACCT

WHERE LNID = '10000'

AND LMT\_TYP IS NULL;

SELECT \*

FROM TID A

JOIN TACCT B

ON A.LNID = B.LNID

WHERE A.LNID = '10000'

AND B.LMT\_TYP IS NULL;

SELECT \*

FROM TID A

JOIN TACCT B

ON A.LNID = B.LNID ;

SELECT \*

FROM TID A

,TACCT B

WHERE A.LNID = B.LNID ;

SELECT \*

FROM TID A

JOIN TCREDIT B

ON A.LNID = B.LNID ;

SELECT \*

FROM TCREDIT A

JOIN TCODE B

ON A.CODE = B.CODE ;

SELECT \*

FROM TACCT A

JOIN TREPAY B

ON A.LNACT = B.LNACT

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