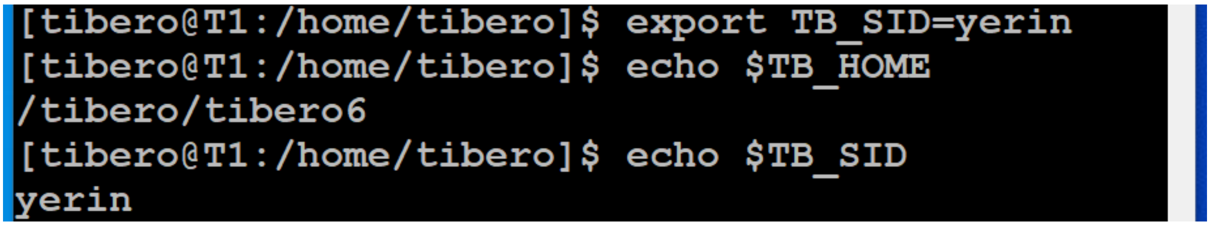
✨

Tibero 5/6장-SQL질의 및 실체화뷰

☰ 태그

```
[tibero@T1:/home/tibero]$ export TB_SID=yerin
[tibero@T1:/home/tibero]$ echo $TB_HOME
/tibero/tibero6
[tibero@T1:/home/tibero]$ echo $TB_SID
yerin
```

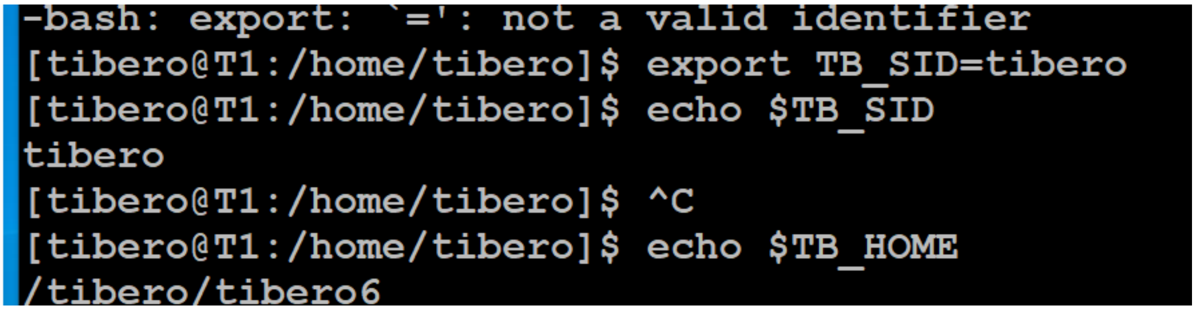
> 인스턴스를 yerin으로 바꾸기 , 예린 인스턴스에서 TB_HOME, TB_SID 조회하기



```
[tibero@T1:/home/tibero]$ export TB_SID=yerin
[tibero@T1:/home/tibero]$ echo $TB_HOME
/tibero/tibero6
[tibero@T1:/home/tibero]$ echo $TB_SID
yerin
```

> 인스턴스를 tibero로 바꾸기, 티베로 인스턴스에서 TB_HOME, TB_SID 조회하기

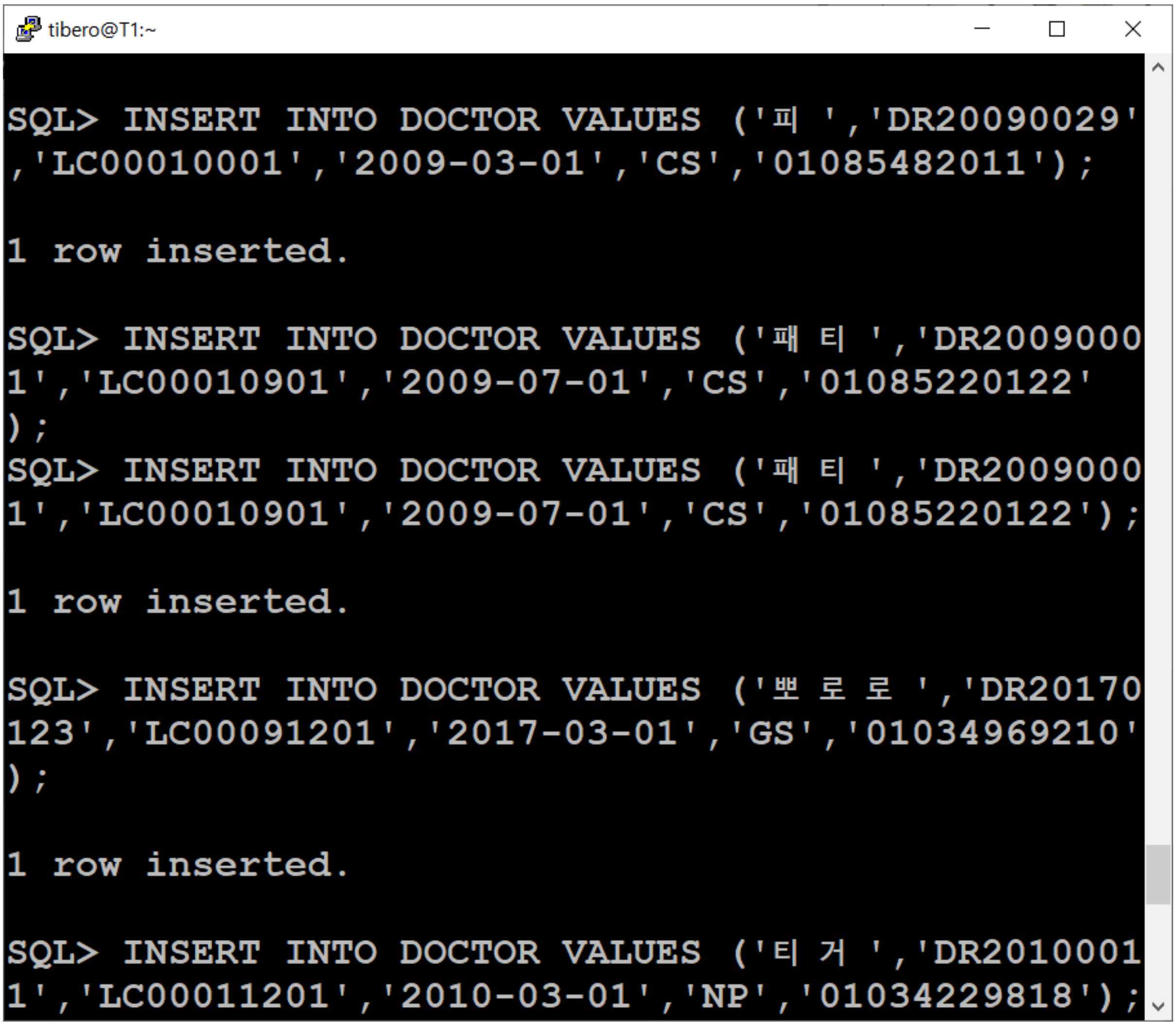
```
[tibero@T1:/home/tibero]$ echo $TB_SID
tibero
[tibero@T1:/home/tibero]$ ^C
[tibero@T1:/home/tibero]$ echo $TB_HOME
/tibero/tibero6
```



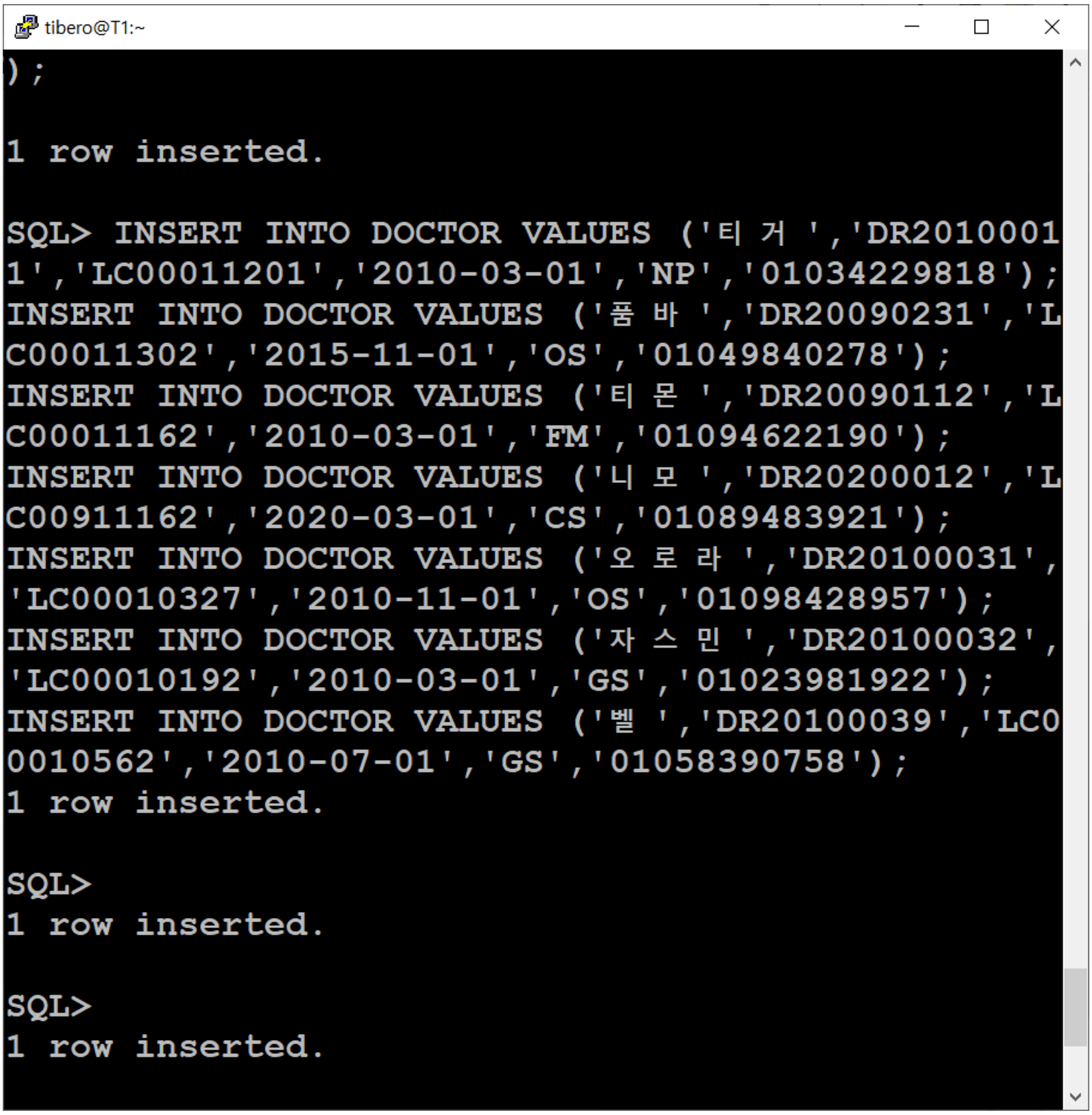
```
-bash: export: `=': not a valid identifier
[tibero@T1:/home/tibero]$ export TB_SID=tibero
[tibero@T1:/home/tibero]$ echo $TB_SID
tibero
[tibero@T1:/home/tibero]$ ^C
[tibero@T1:/home/tibero]$ echo $TB_HOME
/tibero/tibero6
```

테이블에 데이터를 삽입합니다.

```
INSERT INTO DOCTOR VALUES ('피','DR20090029','LC00010001','2009-03-01','CS','0108548201
1');
INSERT INTO DOCTOR VALUES ('패티','DR20090001','LC00010901','2009-07-01','CS','01085220
122');
INSERT INTO DOCTOR VALUES ('뽀로로','DR20170123','LC00091201','2017-03-01','GS','0103496
9210');
INSERT INTO DOCTOR VALUES ('티거','DR20100011','LC00011201','2010-03-01','NP','01034229
818');
INSERT INTO DOCTOR VALUES ('품바','DR20090231','LC00011302','2015-11-01','OS','01049840
278');
INSERT INTO DOCTOR VALUES ('티몬','DR20090112','LC00011162','2010-03-01','FM','01094622
190');
INSERT INTO DOCTOR VALUES ('니모','DR20200012','LC00911162','2020-03-01','CS','01089483
921');
INSERT INTO DOCTOR VALUES ('오로라','DR20100031','LC00010327','2010-11-01','OS','0109842
8957');
INSERT INTO DOCTOR VALUES ('자스민','DR20100032','LC00010192','2010-03-01','GS','0102398
1922');
INSERT INTO DOCTOR VALUES ('벨','DR20100039','LC00010562','2010-07-01','GS','0105839075
8');
```



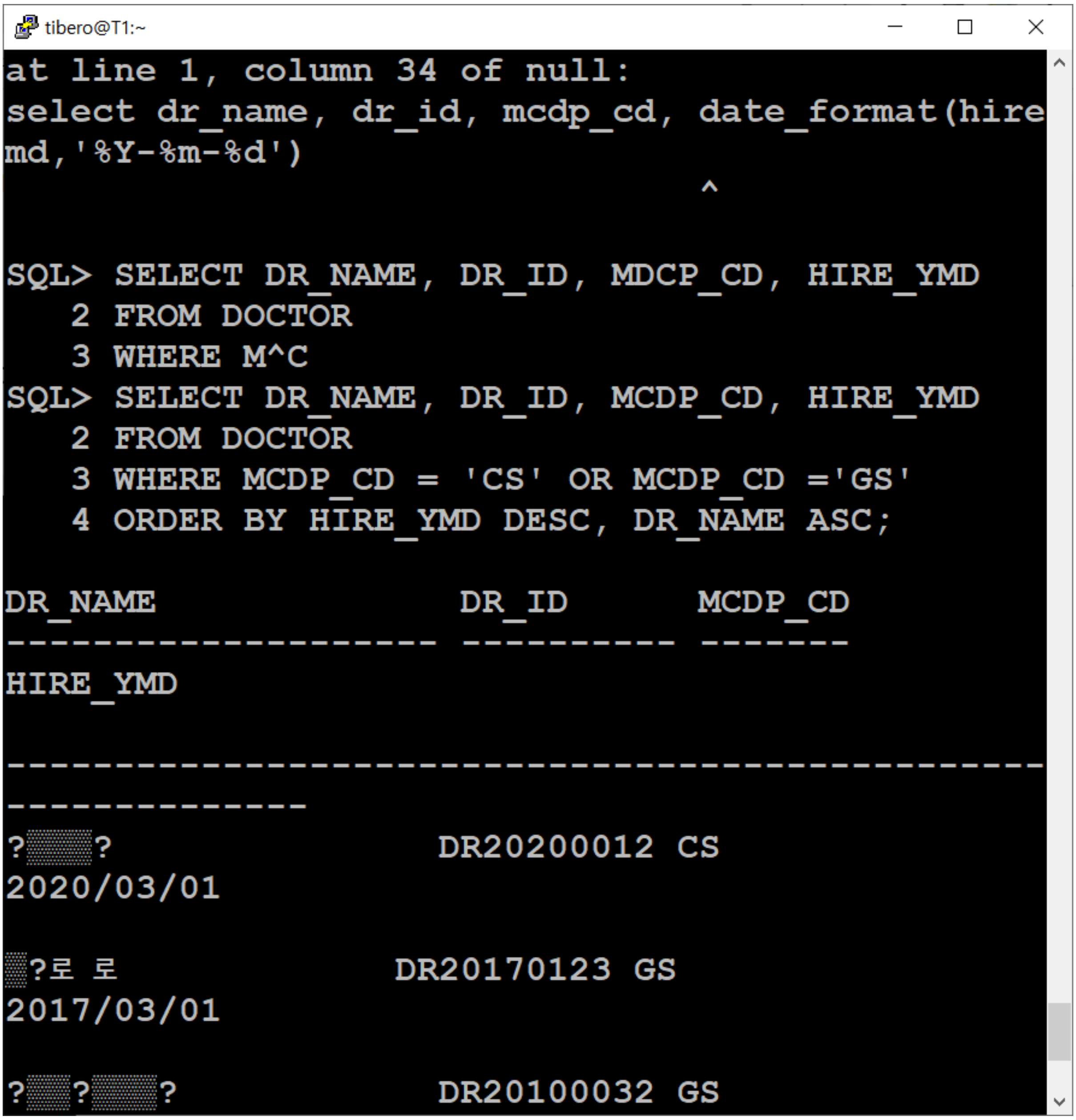
```
SQL> INSERT INTO DOCTOR VALUES ('피 ','DR20090029'
,'LC00010001','2009-03-01','CS','01085482011');

1 row inserted.

SQL> INSERT INTO DOCTOR VALUES ('패 티 ','DR2009000
1','LC00010901','2009-07-01','CS','01085220122'
);
SQL> INSERT INTO DOCTOR VALUES ('패 티 ','DR2009000
1','LC00010901','2009-07-01','CS','01085220122');

1 row inserted.

SQL> INSERT INTO DOCTOR VALUES ('뽀 로 로 ','DR20170
123','LC00091201','2017-03-01','GS','01034969210'
);

1 row inserted.

SQL> INSERT INTO DOCTOR VALUES ('티 거 ','DR2010001
1','LC00011201','2010-03-01','NP','01034229818');
```



```
);

1 row inserted.

SQL> INSERT INTO DOCTOR VALUES ('티거','DR2010001
1','LC00011201','2010-03-01','NP','01034229818');
INSERT INTO DOCTOR VALUES ('품바','DR20090231','L
C00011302','2015-11-01','OS','01049840278');
INSERT INTO DOCTOR VALUES ('티몬','DR20090112','L
C00011162','2010-03-01','FM','01094622190');
INSERT INTO DOCTOR VALUES ('니모','DR20200012','L
C00911162','2020-03-01','CS','01089483921');
INSERT INTO DOCTOR VALUES ('오로라','DR20100031',
'LC00010327','2010-11-01','OS','01098428957');
INSERT INTO DOCTOR VALUES ('자스민','DR20100032',
'LC00010192','2010-03-01','GS','01023981922');
INSERT INTO DOCTOR VALUES ('벨','DR20100039','LC0
0010562','2010-07-01','GS','01058390758');
1 row inserted.

SQL>
1 row inserted.

SQL>
1 row inserted.
```

> `DOCTOR` 테이블에서 진료과가 흉부외과(CS)이거나 일반외과(GS)인 의사의 이름, 의사ID, 진료과, 고용일자를 조회하는 SQL문을 작성해주세요. 이때 결과는 고용일자를 기준으로 내림차순 정렬하고, 고용일자가 같다면 이름을 기준으로 오름차순 정렬하기
>
> ```
> SQL> SELECT DR_NAME, DR_ID, MCDP_CD, HIRE_YMD
>   2 FROM DOCTOR
>   3 WHERE MCDP_CD = 'CS' OR MCDP_CD ='GS'
>   4 ORDER BY HIRE_YMD DESC, DR_NAME ASC;
> ```

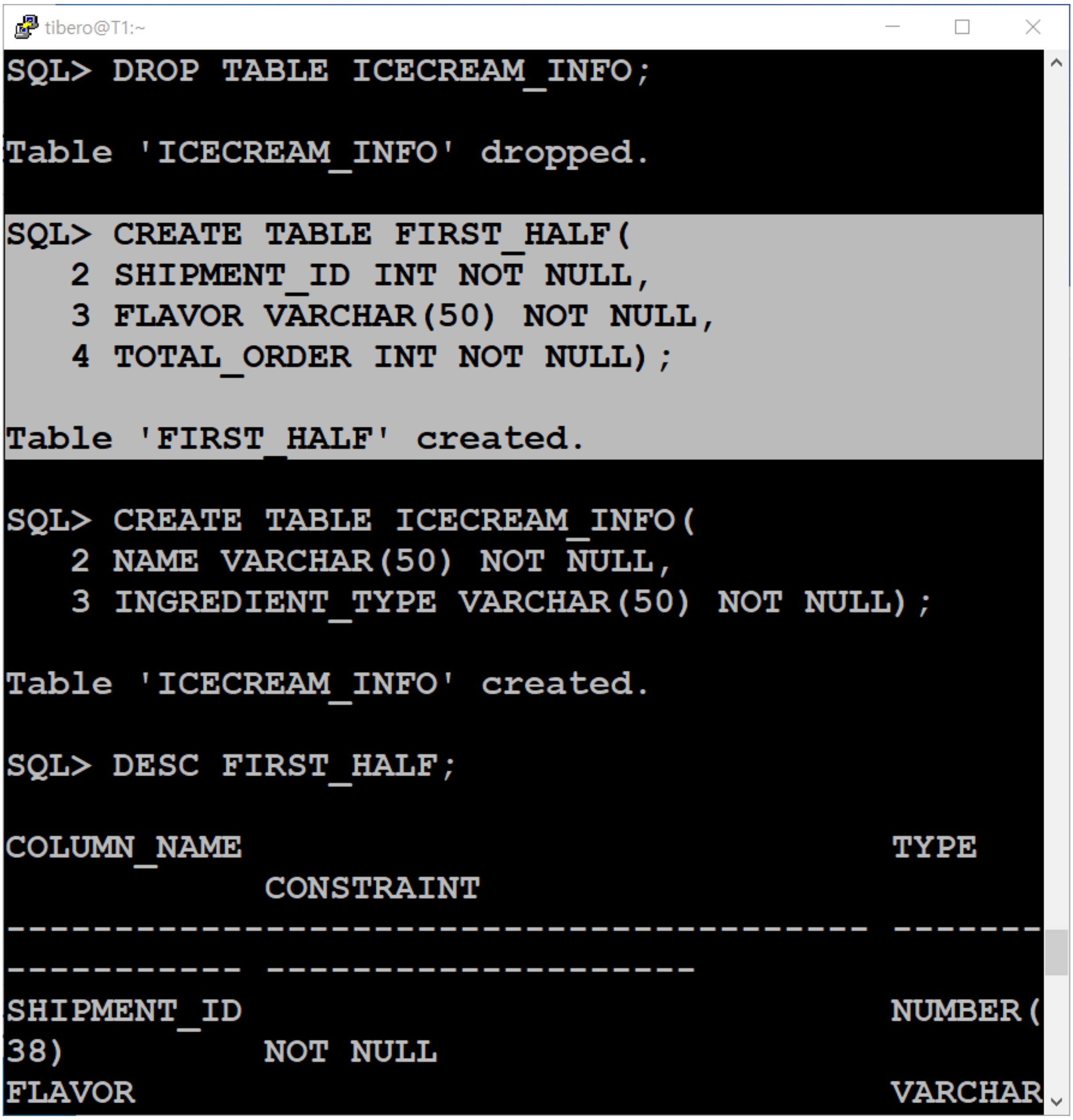
```
DR_NAME              DR_ID      MCDP_CD
-------------------- ---------- -------
HIRE_YMD
--------------------------------------------------------------
?▒▒▒?               DR20200012 CS
2020/03/01

▒?로로            DR20170123 GS
2017/03/01

?▒▒?▒▒▒?            DR20100032 GS
2010/03/01

?▒▒?▒▒               DR20090001 CS
2009/07/01

?▒▒                  DR20090029 CS
2009/03/01


5 rows selected.
```



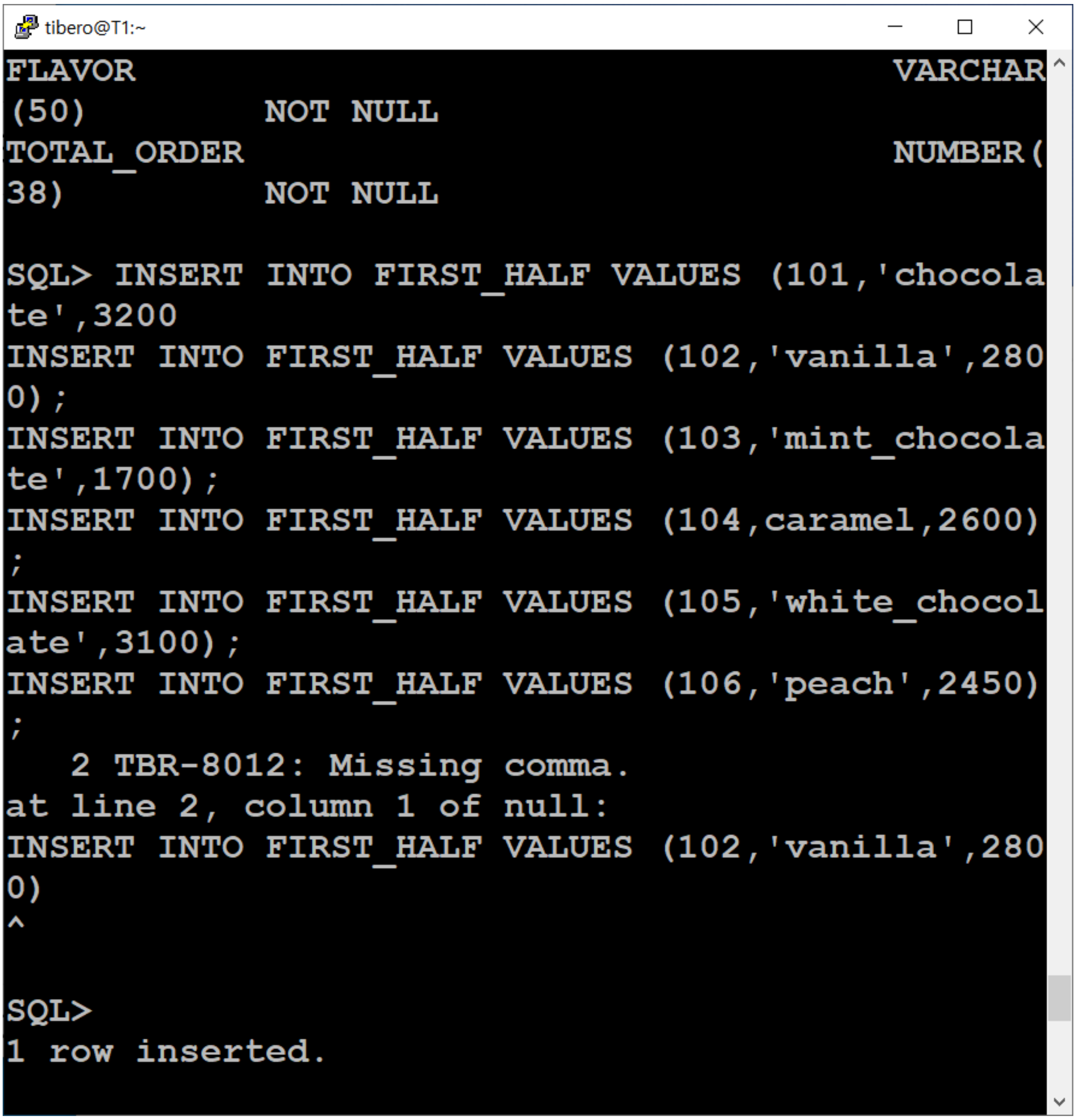
```
at line 1, column 34 of null:
select dr_name, dr_id, mcdp_cd, date_format(hire
md,'%Y-%m-%d')
                                ^

SQL> SELECT DR_NAME, DR_ID, MDCP_CD, HIRE_YMD
   2 FROM DOCTOR
   3 WHERE M^C
SQL> SELECT DR_NAME, DR_ID, MCDP_CD, HIRE_YMD
   2 FROM DOCTOR
   3 WHERE MCDP_CD = 'CS' OR MCDP_CD ='GS'
   4 ORDER BY HIRE_YMD DESC, DR_NAME ASC;

DR_NAME                     DR_ID      MCDP_CD
-------------------- ---------- -------
HIRE_YMD

------------------------------------------------
--------------
?▒▒▒?                DR20200012 CS
2020/03/01

▒?로 로                DR20170123 GS
2017/03/01

?▒▒?▒▒▒?             DR20100032 GS
```

FIRST_HALF 테이블을 만들고 데이터를 삽입합니다.

```
SQL> CREATE TABLE MY_SPACE.FIRST_HALF(
SHIPMENT_ID INT NOT NULL,
FLAVOR VARCHAR(50) NOT NULL,
TOTAL_ORDER INT NOT NULL);

Table 'FIRST_HALF' created.
INSERT INTO FIRST_HALF VALUES (101,'chocolate',3200
INSERT INTO FIRST_HALF VALUES (102,'vanilla',2800);
INSERT INTO FIRST_HALF VALUES (103,'mint_chocolate',1700);
INSERT INTO FIRST_HALF VALUES (104,caramel,2600);
INSERT INTO FIRST_HALF VALUES (105,'white_chocolate',3100);
INSERT INTO FIRST_HALF VALUES (106,'peach',2450);
INSERT INTO FIRST_HALF VALUES (107,'watermelon',2150);
INSERT INTO FIRST_HALF VALUES (108,'mango',2900);
INSERT INTO FIRST_HALF VALUES (109,'strawberry',3100);
INSERT INTO FIRST_HALF VALUES (110,'melon',3150);
INSERT INTO FIRST_HALF VALUES (111,'orange',2900);
INSERT INTO FIRST_HALF VALUES (112,'pineapple',2900);
```



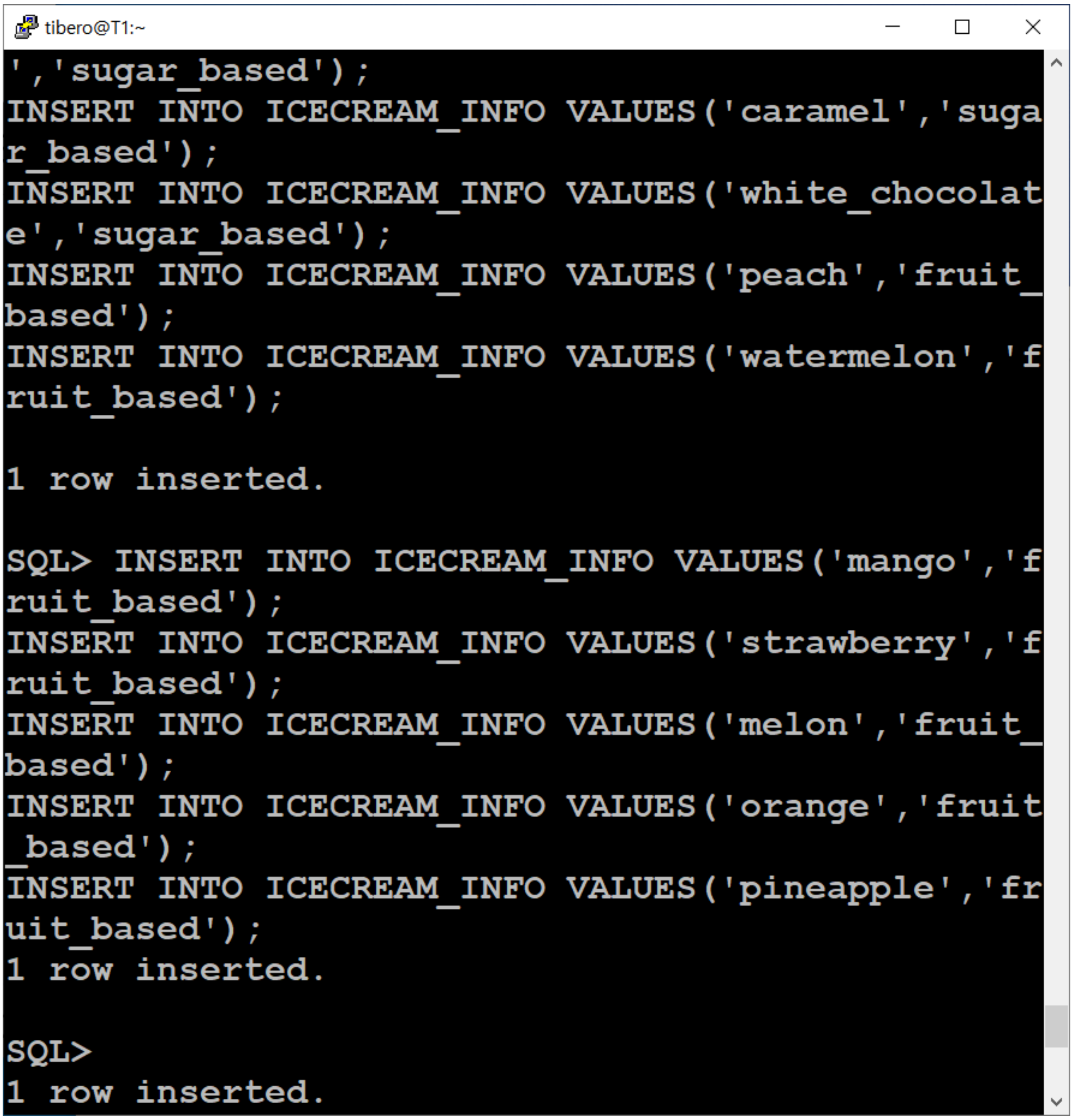
```
SQL> DROP TABLE ICECREAM_INFO;

Table 'ICECREAM_INFO' dropped.

SQL> CREATE TABLE FIRST_HALF(
   2 SHIPMENT_ID INT NOT NULL,
   3 FLAVOR VARCHAR(50) NOT NULL,
   4 TOTAL_ORDER INT NOT NULL);

Table 'FIRST_HALF' created.

SQL> CREATE TABLE ICECREAM_INFO(
   2 NAME VARCHAR(50) NOT NULL,
   3 INGREDIENT_TYPE VARCHAR(50) NOT NULL);

Table 'ICECREAM_INFO' created.

SQL> DESC FIRST_HALF;

COLUMN_NAME                             TYPE
           CONSTRAINT
---------------------------------------- -------
---------- --------------------
SHIPMENT_ID                             NUMBER(
38)        NOT NULL
FLAVOR                                  VARCHAR
```



```
FLAVOR                                          VARCHAR
(50)         NOT NULL
TOTAL_ORDER                                     NUMBER(
38)          NOT NULL

SQL> INSERT INTO FIRST_HALF VALUES (101,'chocola
te',3200
INSERT INTO FIRST_HALF VALUES (102,'vanilla',280
0);
INSERT INTO FIRST_HALF VALUES (103,'mint_chocola
te',1700);
INSERT INTO FIRST_HALF VALUES (104,caramel,2600)
;
INSERT INTO FIRST_HALF VALUES (105,'white_chocol
ate',3100);
INSERT INTO FIRST_HALF VALUES (106,'peach',2450)
;
   2 TBR-8012: Missing comma.
at line 2, column 1 of null:
INSERT INTO FIRST_HALF VALUES (102,'vanilla',280
0)
^


SQL>
1 row inserted.
```

ICECREAM_INFO 테이블을 만들고 데이터를 삽입한다

```
SQL> CREATE TABLE ICECREAM_INFO(
NAME VARCHAR(50) NOT NULL,
INGREDIENT_TYPE VARCHAR(50) NOT NULL);

Table 'ICECREAM_INFO' created.

INSERT INTO ICECREAM_INFO VALUES('chocolate','sugar_based');
INSERT INTO ICECREAM_INFO VALUES('vanilla','sugar_based');
INSERT INTO ICECREAM_INFO VALUES('mint_chocolate','sugar_based');
INSERT INTO ICECREAM_INFO VALUES('caramel','sugar_based');
INSERT INTO ICECREAM_INFO VALUES('white_chocolate','sugar_based');
INSERT INTO ICECREAM_INFO VALUES('peach','fruit_based');
INSERT INTO ICECREAM_INFO VALUES('watermelon','fruit_based');
INSERT INTO ICECREAM_INFO VALUES('mango','fruit_based');
INSERT INTO ICECREAM_INFO VALUES('strawberry','fruit_based');
INSERT INTO ICECREAM_INFO VALUES('melon','fruit_based');
INSERT INTO ICECREAM_INFO VALUES('orange','fruit_based');
INSERT INTO ICECREAM_INFO VALUES('pineapple','fruit_based');
```

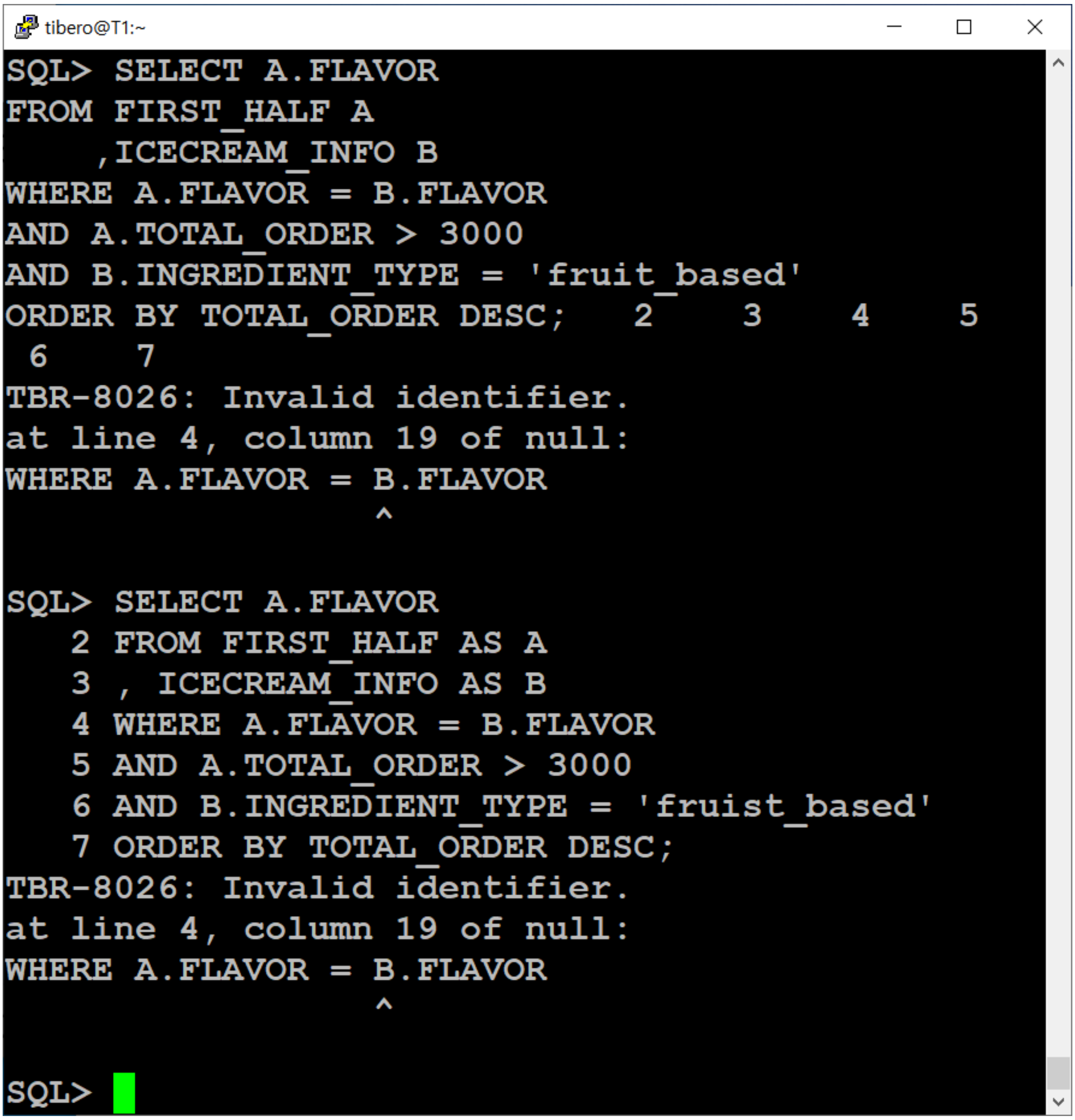


```
','sugar_based');
INSERT INTO ICECREAM_INFO VALUES('caramel','suga
r_based');
INSERT INTO ICECREAM_INFO VALUES('white_chocolat
e','sugar_based');
INSERT INTO ICECREAM_INFO VALUES('peach','fruit_
based');
INSERT INTO ICECREAM_INFO VALUES('watermelon','f
ruit_based');

1 row inserted.

SQL> INSERT INTO ICECREAM_INFO VALUES('mango','f
ruit_based');
INSERT INTO ICECREAM_INFO VALUES('strawberry','f
ruit_based');
INSERT INTO ICECREAM_INFO VALUES('melon','fruit_
based');
INSERT INTO ICECREAM_INFO VALUES('orange','fruit
_based');
INSERT INTO ICECREAM_INFO VALUES('pineapple','fr
uit_based');
1 row inserted.

SQL>
1 row inserted.
```

> 상반기 아이스크림 총주문량이 3,000보다 높으면서 아이스크림의 주성분이 과일인 아이스크림의 맛을 총주문량이 큰 순서대로 조회하는 SQL 문을 작성해주세요.

아래는 MYSQL에서 실행하는 코드이다. 티베로에서는 오류가 난다. 어떻게 해야 실행할 수 있을까?

```
SELECT A.FLAVOR
FROM FIRST_HALF A
    ,ICECREAM_INFO B
WHERE A.FLAVOR = B.FLAVOR
AND A.TOTAL_ORDER > 3000
AND B.INGREDIENT_TYPE = 'fruit_based'
ORDER BY TOTAL_ORDER DESC;
```



```
SQL> SELECT A.FLAVOR
FROM FIRST_HALF A
    ,ICECREAM_INFO B
WHERE A.FLAVOR = B.FLAVOR
AND A.TOTAL_ORDER > 3000
AND B.INGREDIENT_TYPE = 'fruit_based'
ORDER BY TOTAL_ORDER DESC;   2    3    4    5
 6    7
TBR-8026: Invalid identifier.
at line 4, column 19 of null:
WHERE A.FLAVOR = B.FLAVOR
                 ^


SQL> SELECT A.FLAVOR
   2 FROM FIRST_HALF AS A
   3 , ICECREAM_INFO AS B
   4 WHERE A.FLAVOR = B.FLAVOR
   5 AND A.TOTAL_ORDER > 3000
   6 AND B.INGREDIENT_TYPE = 'fruist_based'
   7 ORDER BY TOTAL_ORDER DESC;
TBR-8026: Invalid identifier.
at line 4, column 19 of null:
WHERE A.FLAVOR = B.FLAVOR
                 ^


SQL> 
```

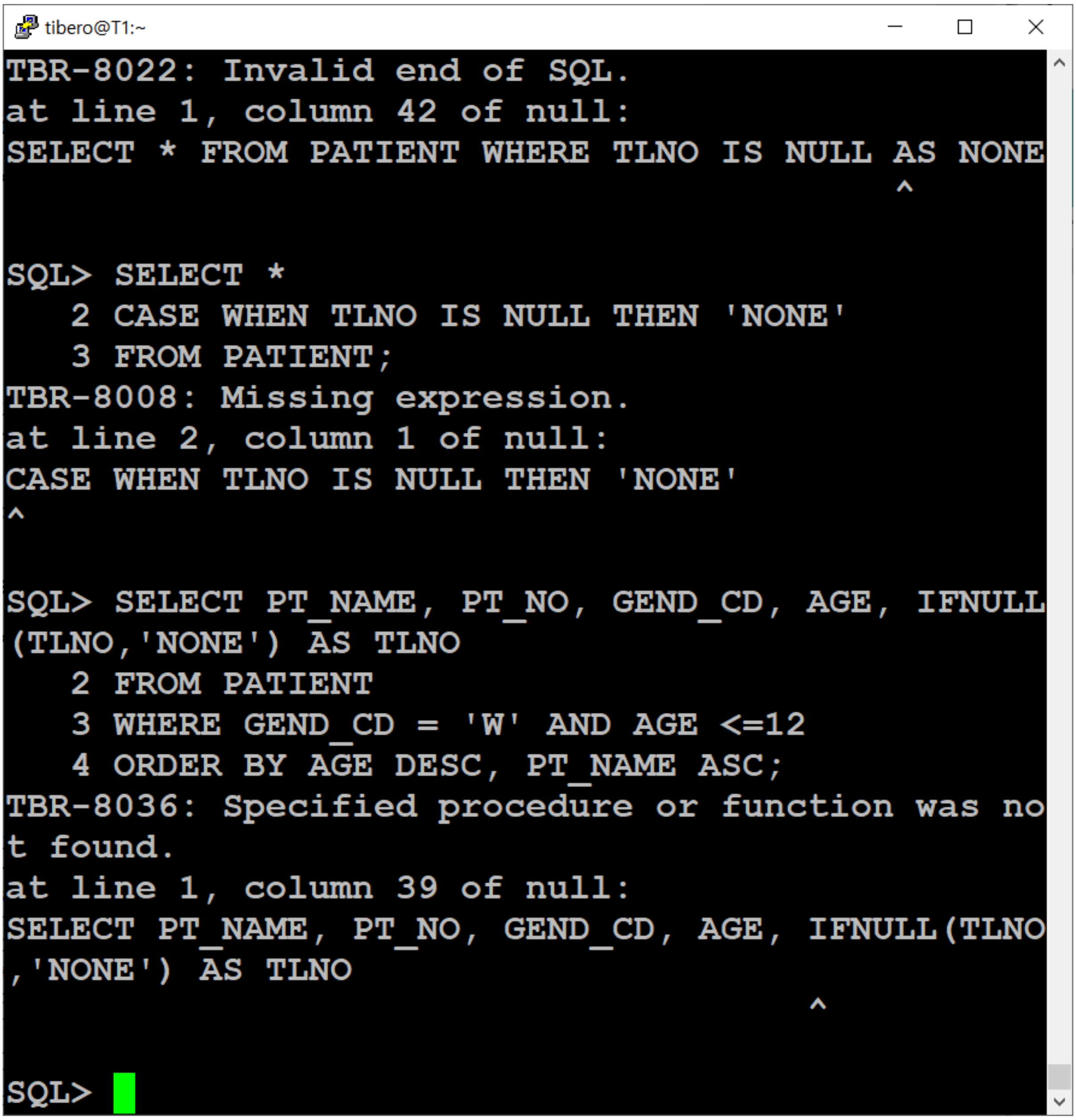
> PATIENT 테이블에 관하여 데이터를 삽입한다.

```
INSERT INTO PATIENT VALUES ('PT22000003','브라운','M','18','01031246641');
INSERT INTO PATIENT VALUES ('PT22000004','크롱','M','7',NULL);
INSERT INTO PATIENT VALUES ('PT22000006','뽀뽀','W','8',NULL);
INSERT INTO PATIENT VALUES ('PT22000009','한나','W','12','01032323117');
INSERT INTO PATIENT VALUES ('PT22000012','뿡뿡이','M','5',NULL);
INSERT INTO PATIENT VALUES ('PT22000013','크리스','M','30','01059341192');
INSERT INTO PATIENT VALUES ('PT22000014','토프','W','22','01039458213');
INSERT INTO PATIENT VALUES ('PT22000018','안나','W','11',NULL);
INSERT INTO PATIENT VALUES ('PT22000019','바라','W','10','01079068799');
INSERT INTO PATIENT VALUES ('PT22000021','릴로','W','33','01023290767');
```

> `PATIENT` 테이블에서 12세 이하인 여자환자의 환자이름, 환자번호, 성별코드, 나이, 전화번호를 조회하는 SQL문을 작성해주세요. 이때 전화번호가 없는 경우, 'NONE'으로 출력시켜 주시고 결과는 나이를 기준으로 내림차순 정렬하고, 나이 같다면 환자이름을 기준으로 오름차순 정렬해주세요.

```
my sql모드
SELECT PT_NO, PT_NAME, GEND_CD, AGE, TLNO
FROM PATIENT
WHERE
AGE > 12 AND PLNO AS NONE = NULL AND PLNO IS NOT NULL
ORDER BY AGE DESC, PT_NAME ASC
```

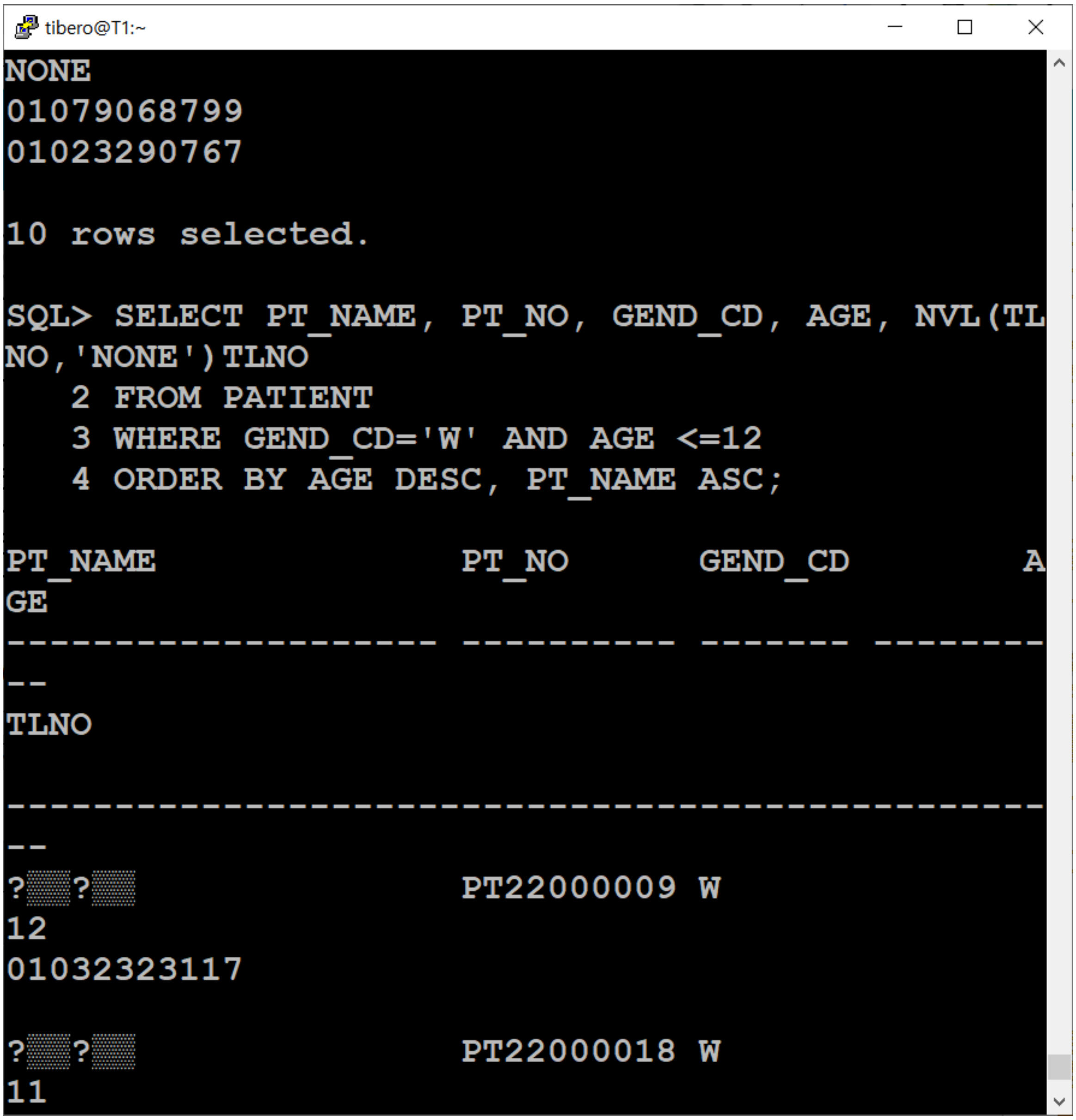
```
tibero 모드
SELECT PT_NAME, PT_NO, GEND_CD, AGE,IFNULL(TLNO,'NONE') as TLNO
from PATIENT
where GEND_CD = 'w'and AGE <= 12
order by AGE desc, PT_NAME asc;
```



```
TBR-8022: Invalid end of SQL.
at line 1, column 42 of null:
SELECT * FROM PATIENT WHERE TLNO IS NULL AS NONE
                                         ^


SQL> SELECT *
   2 CASE WHEN TLNO IS NULL THEN 'NONE'
   3 FROM PATIENT;
TBR-8008: Missing expression.
at line 2, column 1 of null:
CASE WHEN TLNO IS NULL THEN 'NONE'
^


SQL> SELECT PT_NAME, PT_NO, GEND_CD, AGE, IFNULL
(TLNO,'NONE') AS TLNO
   2 FROM PATIENT
   3 WHERE GEND_CD = 'W' AND AGE <=12
   4 ORDER BY AGE DESC, PT_NAME ASC;
TBR-8036: Specified procedure or function was no
t found.
at line 1, column 39 of null:
SELECT PT_NAME, PT_NO, GEND_CD, AGE, IFNULL(TLNO
,'NONE') AS TLNO
                                      ^


SQL>
```

```
 SQL> SELECT NVL(TLNO,'NONE')TLNO
   2 FROM PATIENT;

TLNO
--------------------------------------------------
01031246641
NONE
NONE
01032323117
NONE
01059341192
01039458213
NONE
01079068799
01023290767
```

```
SQL> SELECT PT_NAME, PT_NO, GEND_CD, AGE, NVL(TLNO,'NONE')TLNO
   2 FROM PATIENT
   3 WHERE GEND_CD='W' AND AGE <=12
   4 ORDER BY AGE DESC, PT_NAME ASC;

PT_NAME              PT_NO      GEND_CD        AGE
-------------------- ---------- ------- ----------
TLNO
--------------------------------------------------
?▒?▒                 PT22000009 W               12
01032323117

?▒?▒                 PT22000018 W               11
NONE

바라                PT22000019 W               10
01079068799

▒?▒?                 PT22000006 W                8
NONE


4 rows selected.
```



```
NONE
01079068799
01023290767

10 rows selected.

SQL> SELECT PT_NAME, PT_NO, GEND_CD, AGE, NVL(TL
NO,'NONE')TLNO
  2 FROM PATIENT
  3 WHERE GEND_CD='W' AND AGE <=12
  4 ORDER BY AGE DESC, PT_NAME ASC;

PT_NAME                    PT_NO        GEND_CD         A
GE
-------------------- ---------- ------- --------
--
TLNO

------------------------------------------------
--
?▒?▒                       PT22000009 W
12
01032323117

?▒?▒                       PT22000018 W
11
```

`MEMBER_PROFILE` 테이블에서 생일이 3월인 여성 회원의 ID, 이름, 성별, 생년월일을 조회하는 SQL문을 작성해주세요. 이때 전화번호가 NULL인 경우는 출력대상에서 제외시켜 주시고, 결과는 회원ID를 기준으로 오름차순 정렬해주세요.

```
INSERT INTO MEMBER_PROFILE VALUES('jiho92@naver.com','이지호','01076432111','W','1992-02
-12');
INSERT INTO MEMBER_PROFILE VALUES('jiyoon22@hotmail.com','김지윤','01032324117','W','199
2-02-22');
INSERT INTO MEMBER_PROFILE VALUES('jihoon93@hanmail.net','김지훈','01023258688','M','199
3-02-23');
```

```
INSERT INTO MEMBER_PROFILE VALUES('seoyeons@naver.com','박서연','01076482209','W','1993-
03-16');
INSERT INTO MEMBER_PROFILE VALUES('yoonsy94@gmail.com','윤서연','NULL','W','1994-03-1
9');
```

멤버의 생일을 달력의 기준으로 바꿔준다.

```
SQL> SELECT MEMBER_ID,MEMBER_NAME,GENDER,TLNO,TO_CHAR(DATE_OF_BIRTH,'MM')
   2 FROM MEMBER_PROFILE
   3 WHERE GENDER='W' AND TO_CHAR(DATE_OF_BIRTH,'MM') = '03' AND TLNO IS NOT NULL;

MEMBER_ID
--------------------------------------------------------------------------------
MEMBER_NAME                                        GENDER
-------------------------------------------------- ------
TLNO                                               TO_CHAR(DATE_OF_BIRTH,'MM')
-------------------------------------------------- ---------------------------
seoyeons@naver.com
박서?▒                                            W
01076482209                                        03

yoonsy94@gmail.com
?▒?▒?▒                                             W
NULL                                               03


2 rows selected.
```

```
MEMBER_ID                                 VNOT NULL
MEMBER_NAME                               VNOT NULL
TLNO                                      V
GENDER                                    V
DATE_OF_BIRTH                             D
SELECT MEMBER_ID, MEMBER_NAME, GENDER, TLNO, TO_CHAR(DATE_OF_BIRTH,'MM')
   2 FROM MEMBER_PROFILE
   3 WHERE GENDER='W' AND TLNO IS NOT NULL

SQL> SELECT TO_CHAR(DATE_OF_BIRTH,'MM')
   2 FROM MEMBER_PROFILE
   3 WHERE TO_CHAR(DATE_OF_BIRTH,'MM') = '03';

TO_CHAR(DATE_OF_BIRTH,'MM')
---------------------------
```

```
03
03

2 rows selected.
```

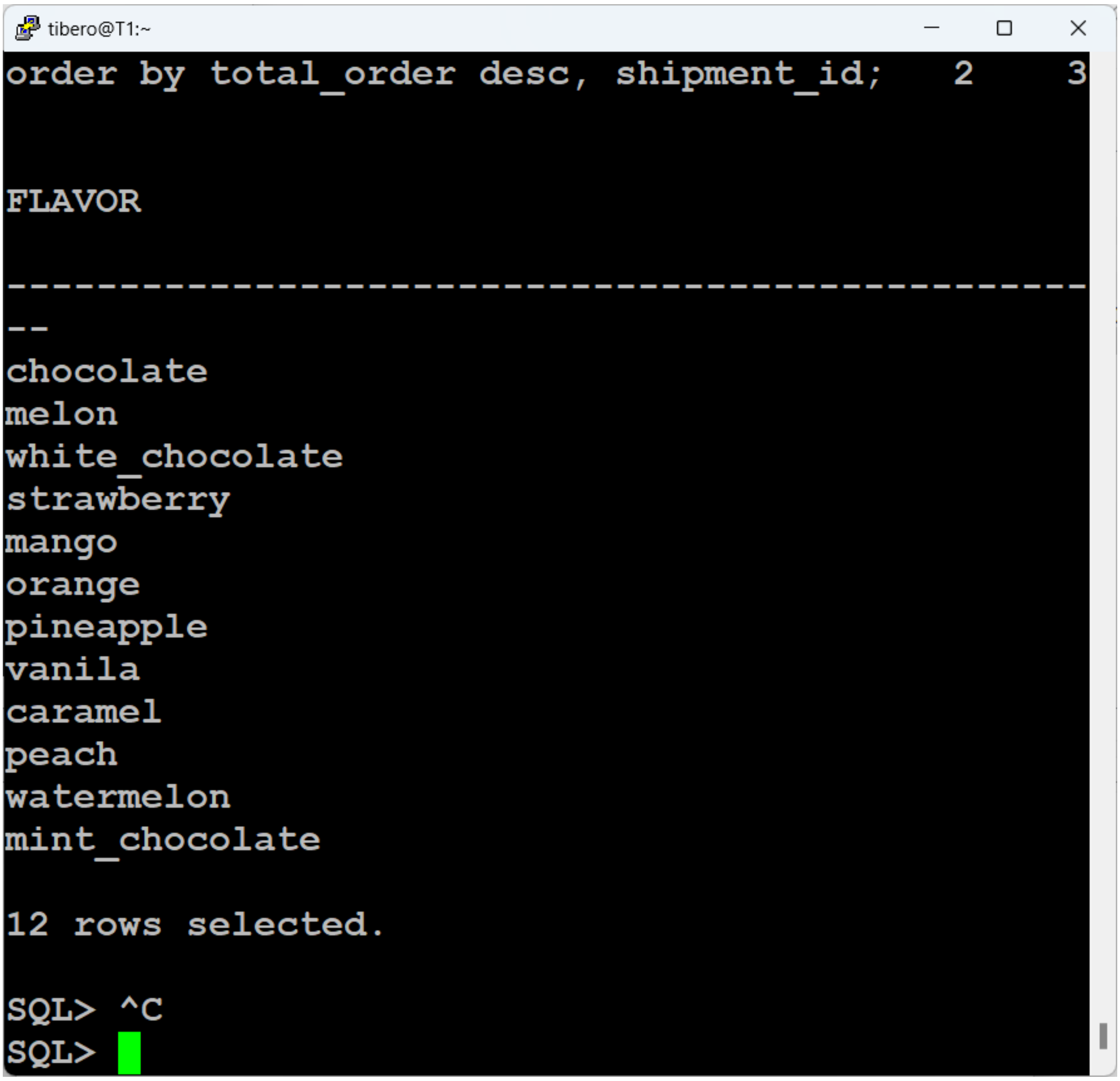
> `FOOD_FACTORY` 테이블에서 강원도에 위치한 식품공장의 공장 ID, 공장 이름, 주소를 조회하는 SQL문을 작성해주세요. 이때 결과는 공장 ID를 기준으로 오름차순 정렬해주세요.

```
(FACTORY_ID','(FACTORY_NAME','ADDRESS','TLNO
INSERT INTO FOOD_FACTORY VALUES('FT19980003','(주)맛있는라면','강원도 정선군 남면 칠현로 67
9','033-431-3122');
INSERT INTO FOOD_FACTORY VALUES('FT19980004','(주)맛있는기름','경기도 평택시 포승읍 포승공단순환
로 245','031-651-2410');
INSERT INTO FOOD_FACTORY VALUES('FT20010001','(주)맛있는소스','경상북도 구미시 1공단로7길 58-1
1','054-231-2121');
INSERT INTO FOOD_FACTORY VALUES('FT20010002','(주)맛있는통조림','전라남도 영암군 미암면 곤미현로
1336','061-341-5210');
INSERT INTO FOOD_FACTORY VALUES('FT20100001','(주)맛있는차','전라남도 장성군 서삼면 장산리 233-
1번지','061-661-1420');
INSERT INTO FOOD_FACTORY VALUES('FT20100002','(주)맛있는김치','충청남도 아산시 탕정면 탕정면로 4
85','041-241-5421');
INSERT INTO FOOD_FACTORY VALUES('FT20100003','(주)맛있는음료','강원도 원주시 문막읍 문막공단길 1
54','033-232-7630');
INSERT INTO FOOD_FACTORY VALUES('FT20100004','(주)맛있는국','강원도 평창군 봉평면 진조길 227-3
5','033-323-6640');
INSERT INTO FOOD_FACTORY VALUES('FT20110001','(주)맛있는밥','경기도 화성시 팔탄면 가재리 34번
지','031-661-1532');
INSERT INTO FOOD_FACTORY VALUES('FT20110002','(주)맛있는과자','광주광역시 북구 하서로 222','06
2-211-7759'); FACTORY_ID
```

```
SELECT FACTORY_ID
      ,FACTORY_NAME
      ,ADDRESS
  FROM FOOD_FACTORY
 WHERE ADDRESS LIKE '%강원도%'
 ORDER BY FACTORY_ID;
```

> 상반기에 판매된 아이스크림의 맛을 총주문량을 기준으로 내림차순 정렬하고 총주문량이 같다면 출하 번호를 기준으로 오름차순 정렬하여 조회하는 SQL 문을 작성해주세요.

```
SQL> select flavor
from first_half
order by total_order desc, shipment_id;   2    3

FLAVOR
--------------------------------------------------
chocolate
melon
white_chocolate
strawberry
mango
orange
pineapple
vanila
caramel
peach
watermelon
mint_chocolate

12 rows selected.
```



```
order by total_order desc, shipment_id;    2    3


FLAVOR

------------------------------------------------
--
chocolate
melon
white_chocolate
strawberry
mango
orange
pineapple
vanila
caramel
peach
watermelon
mint_chocolate

12 rows selected.

SQL> ^C
SQL> 
```

> `FOOD_PRODUCT` 테이블에서 식품분류별로 가격이 제일 비싼 식품의 분류, 가격, 이름을 조회하는 SQL문을 작성해주세요. 이때 식품분류가 '과자', '국', '김치', '식용유'인 경우만 출력시켜 주시고 결과는 식품 가격을 기준으로 내림차순 정렬해주세요. - MYSQL

```
SELECT CATEGORY, PRICE AS MAX_PRICE, PRODUCT_NAME
FROM FOOD_PRODUCT
WHERE CATEGORY IN ('과자', '국', '김치', '식용유')
```

```
AND PRICE IN (SELECT MAX(PRICE) FROM FOOD_PRODUCT GROUP BY CATEGORY)
GROUP BY CATEGORY
ORDER BY MAX_PRICE DESC;
```

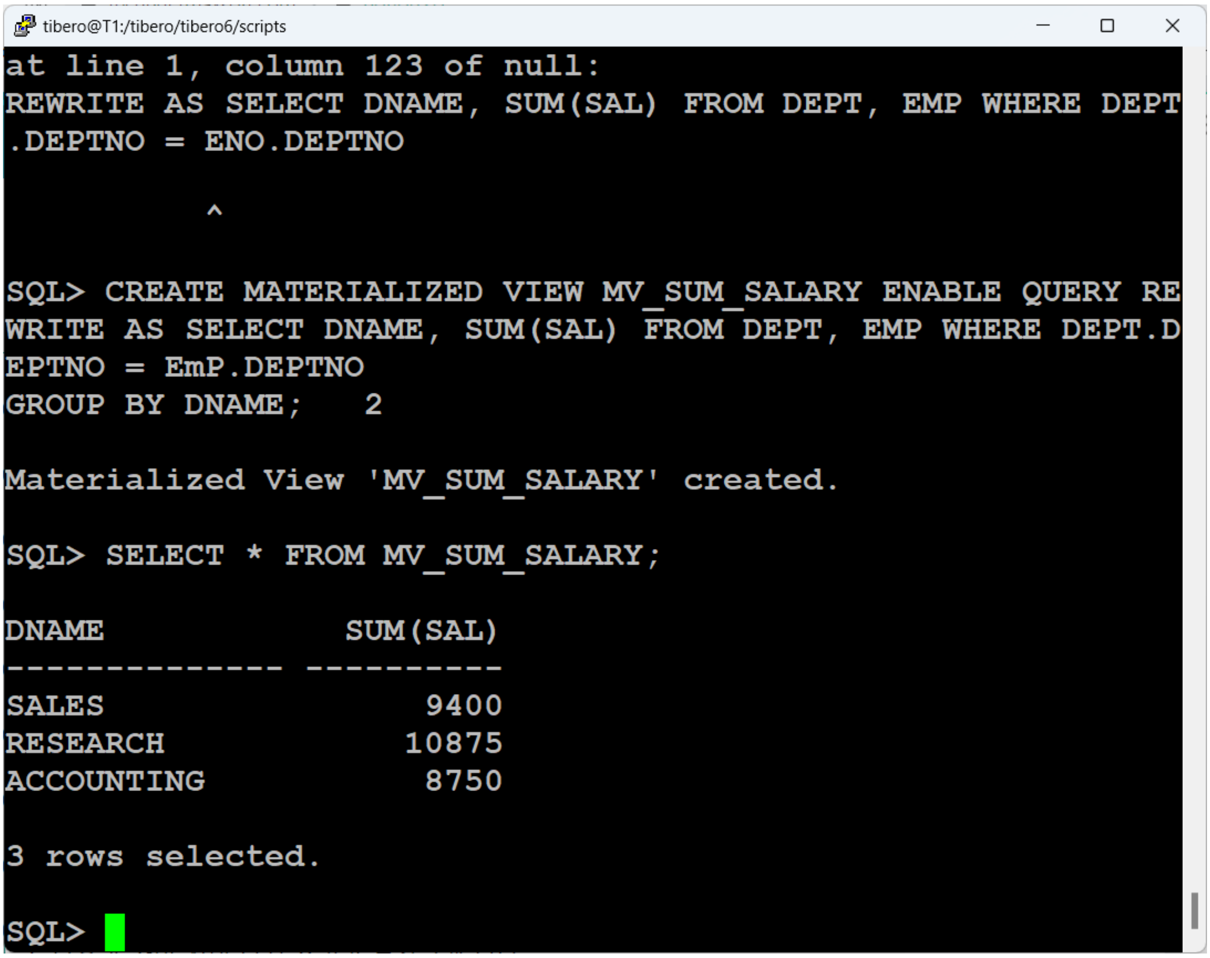
실체화뷰

```
SQL> CREATE MATERIALIZED VIEW MV_SUM_SALARY ENABLE QUERY REWRITE AS SELECT DNAME, SUM
(SAL) FROM DEPT, EMP WHERE DEPT.DEPTNO = EmP.DEPTNO
GROUP BY DNAME;   2

Materialized View 'MV_SUM_SALARY' created.

SQL> SELECT * FROM MV_SUM_SALARY;

DNAME            SUM(SAL)
-------------- ----------
SALES                9400
RESEARCH            10875
ACCOUNTING           8750

3 rows selected.
```

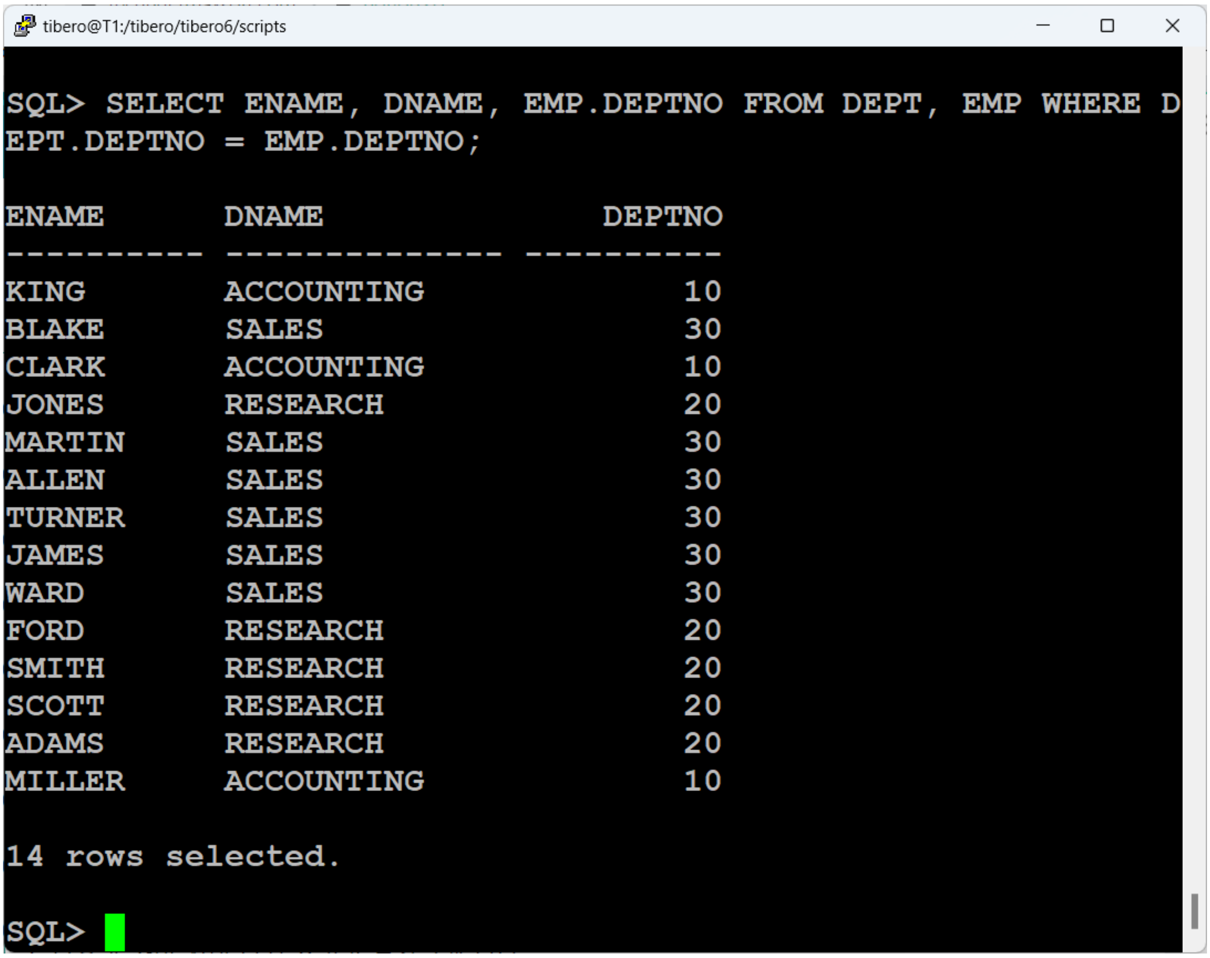
실체화뷰 예시2

```
SQL> CREATE MATERIALIZED VIEW MV_JOIN_DEPT_EMP ENABLE QUERY REWRITE AS
    SELECT ENAME, DNAME, DEPT.DEPTNO FROM DEPT, EMP WHERE DEPT.DEPTNO = EMP.DEPTNO;

Materialized View 'MV_JOIN_DEPT_EMP' created.

SQL> SELECT ENAME, DNAME, EMP.DEPTNO FROM DEPT, EMP WHERE DEPT.DEPTNO = EMP.DEPTNO;

ENAME      DNAME              DEPTNO
---------- -------------- ----------
KING       ACCOUNTING             10
BLAKE      SALES                  30
CLARK      ACCOUNTING             10
JONES      RESEARCH               20
MARTIN     SALES                  30
ALLEN      SALES                  30
TURNER     SALES                  30
JAMES      SALES                  30
WARD       SALES                  30
FORD       RESEARCH               20
SMITH      RESEARCH               20
SCOTT      RESEARCH               20
ADAMS      RESEARCH               20
MILLER     ACCOUNTING             10

14 rows selected.
SQL> SELECT ENAME, DNAME, DEPTNO FROM MV_JOIN_DEPT_EMP;
```

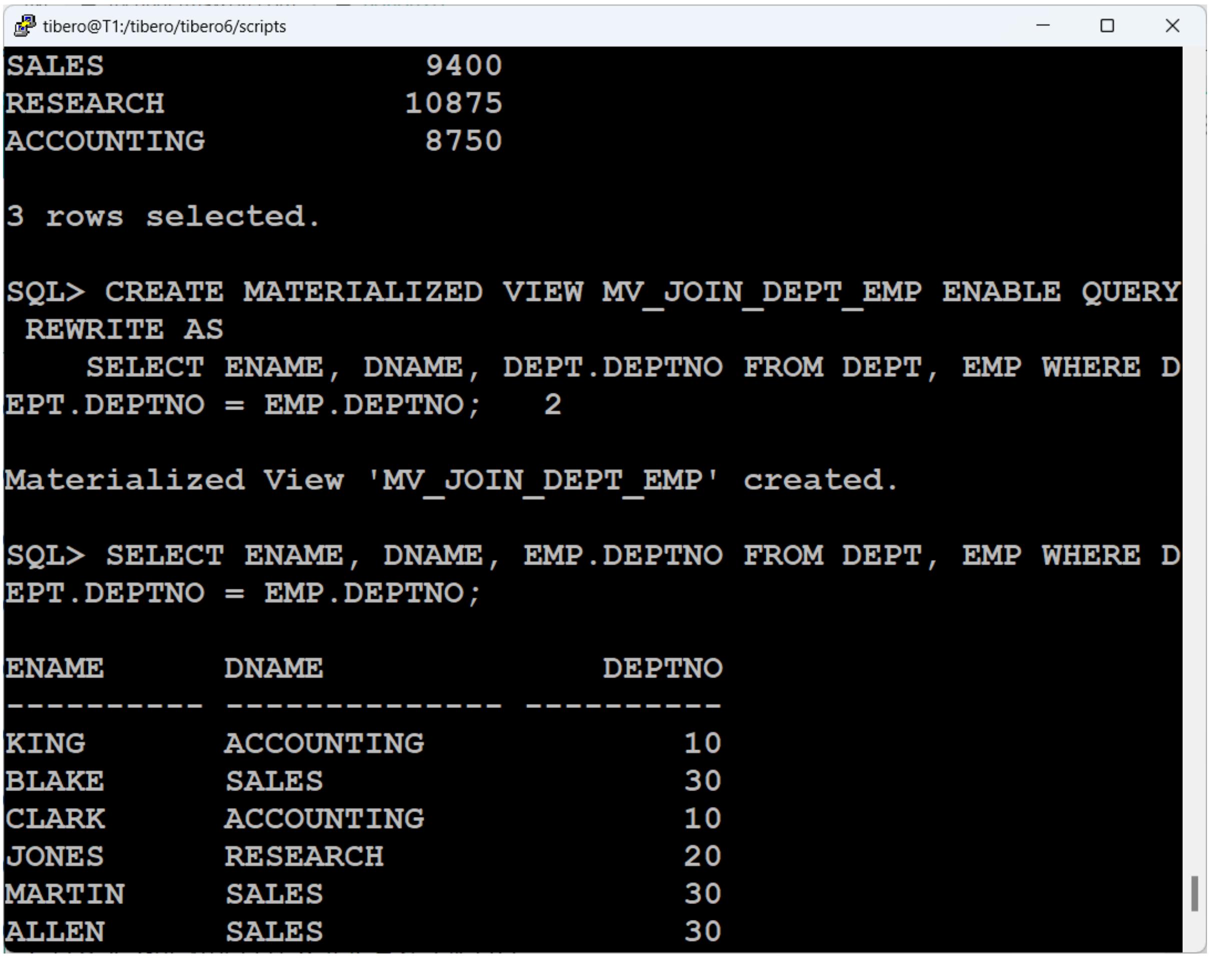
```
ENAME      DNAME          DEPTNO
---------- -------------- ----------
KING       ACCOUNTING             10
BLAKE      SALES                  30
CLARK      ACCOUNTING             10
JONES      RESEARCH               20
MARTIN     SALES                  30
ALLEN      SALES                  30
TURNER     SALES                  30
JAMES      SALES                  30
WARD       SALES                  30
FORD       RESEARCH               20
SMITH      RESEARCH               20
SCOTT      RESEARCH               20
ADAMS      RESEARCH               20
MILLER     ACCOUNTING             10

14 rows selected.
```



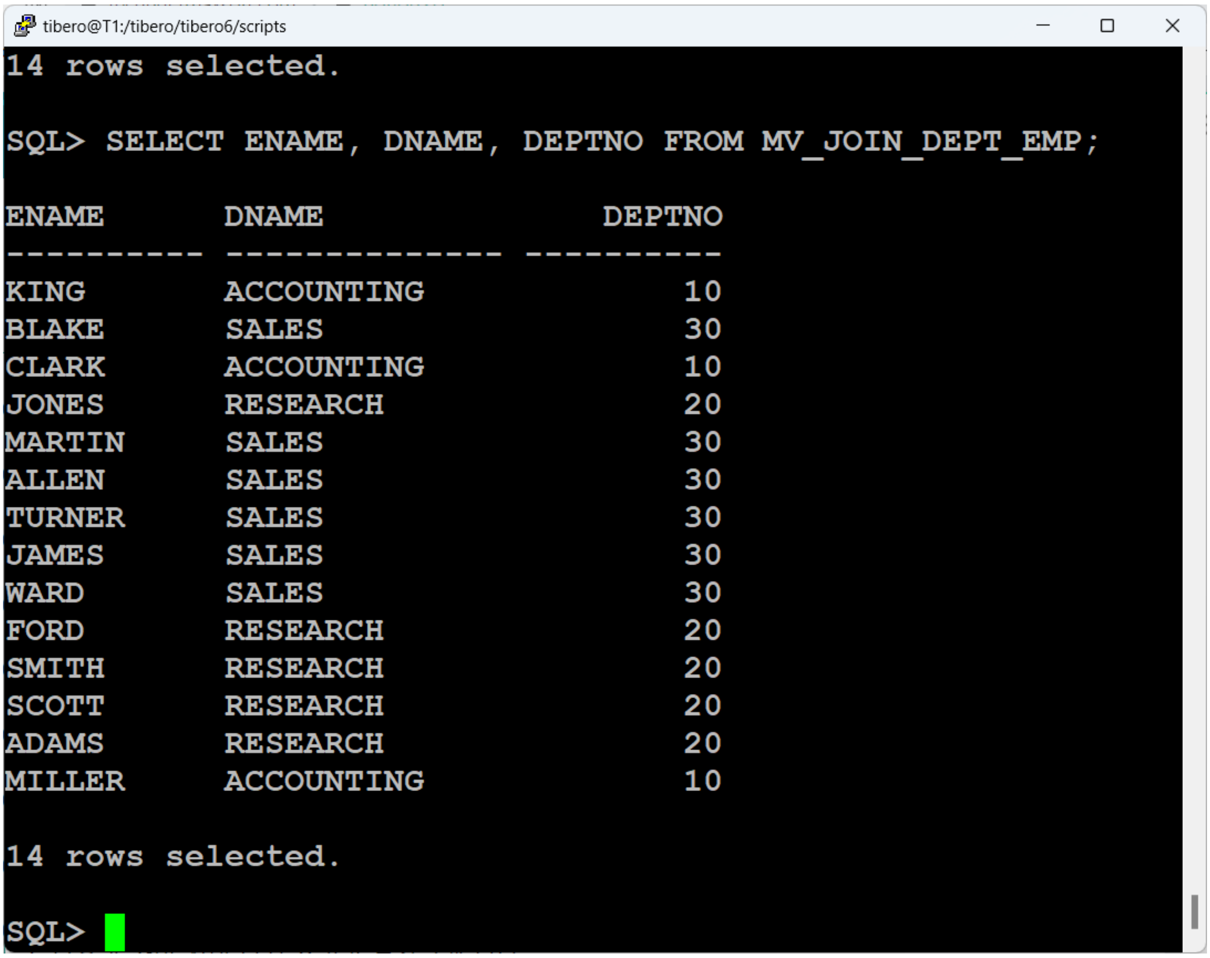
```
at line 1, column 123 of null:
REWRITE AS SELECT DNAME, SUM(SAL) FROM DEPT, EMP WHERE DEPT
.DEPTNO = ENO.DEPTNO

          ^

SQL> CREATE MATERIALIZED VIEW MV_SUM_SALARY ENABLE QUERY RE
WRITE AS SELECT DNAME, SUM(SAL) FROM DEPT, EMP WHERE DEPT.D
EPTNO = EmP.DEPTNO
GROUP BY DNAME;   2

Materialized View 'MV_SUM_SALARY' created.

SQL> SELECT * FROM MV_SUM_SALARY;

DNAME             SUM(SAL)
-------------- ----------
SALES                9400
RESEARCH            10875
ACCOUNTING           8750

3 rows selected.

SQL>
```



```
SQL> SELECT ENAME, DNAME, EMP.DEPTNO FROM DEPT, EMP WHERE D
EPT.DEPTNO = EMP.DEPTNO;

ENAME      DNAME              DEPTNO
---------- -------------- ----------
KING       ACCOUNTING             10
BLAKE      SALES                  30
CLARK      ACCOUNTING             10
JONES      RESEARCH               20
MARTIN     SALES                  30
ALLEN      SALES                  30
TURNER     SALES                  30
JAMES      SALES                  30
WARD       SALES                  30
FORD       RESEARCH               20
SMITH      RESEARCH               20
SCOTT      RESEARCH               20
ADAMS      RESEARCH               20
MILLER     ACCOUNTING             10

14 rows selected.

SQL> 
```



```
SALES                      9400
RESEARCH                  10875
ACCOUNTING                 8750

3 rows selected.

SQL> CREATE MATERIALIZED VIEW MV_JOIN_DEPT_EMP ENABLE QUERY
 REWRITE AS
    SELECT ENAME, DNAME, DEPT.DEPTNO FROM DEPT, EMP WHERE D
EPT.DEPTNO = EMP.DEPTNO;   2

Materialized View 'MV_JOIN_DEPT_EMP' created.

SQL> SELECT ENAME, DNAME, EMP.DEPTNO FROM DEPT, EMP WHERE D
EPT.DEPTNO = EMP.DEPTNO;

ENAME      DNAME                  DEPTNO
---------- -------------- ----------
KING       ACCOUNTING                 10
BLAKE      SALES                      30
CLARK      ACCOUNTING                 10
JONES      RESEARCH                   20
MARTIN     SALES                      30
ALLEN      SALES                      30
```



```
14 rows selected.

SQL> SELECT ENAME, DNAME, DEPTNO FROM MV_JOIN_DEPT_EMP;

ENAME      DNAME              DEPTNO
---------- -------------- ----------
KING       ACCOUNTING             10
BLAKE      SALES                  30
CLARK      ACCOUNTING             10
JONES      RESEARCH               20
MARTIN     SALES                  30
ALLEN      SALES                  30
TURNER     SALES                  30
JAMES      SALES                  30
WARD       SALES                  30
FORD       RESEARCH               20
SMITH      RESEARCH               20
SCOTT      RESEARCH               20
ADAMS      RESEARCH               20
MILLER     ACCOUNTING             10

14 rows selected.

SQL>
```