CS 325 Week 5-1

more clauses for the select statement's where clause

NOW -- how about some SQL?

- DEMO some of the features discussed in SQL Reading Packet 3, WITH the understanding that you WILL read this packet and it has MORE details about these features;

NOW -- how about some SQL?

- DEMO some of the features discussed in SQL Reading Packet 3, WITH the understanding that you WILL read this packet and it has MORE details about these features;
- a few points about SQL:
  - it is NOT case-sensitve EXCEPT within quotes!

NOW -- how about some SQL?

- DEMO some of the features discussed in SQL Reading Packet 3, WITH the understanding that you WILL read this packet and it has MORE details about these features;
- a few points about SQL:
  - it is NOT case-sensitve EXCEPT within quotes!
  - I tend to type SQL in (mostly) lowercase, but I am fine if you use upper-case or mixed case
    - it is nice if you follow a consistent pattern... 8-)

NOW -- how about some SQL?

- DEMO some of the features discussed in SQL Reading Packet 3, WITH the understanding that you WILL read this packet and it has MORE details about these features;
- a few points about SQL:
  - it is NOT case-sensitve EXCEPT within quotes!
  - I tend to type SQL in (mostly) lowercase, but I am fine if you use upper-case or mixed case
    - it is nice if you follow a consistent pattern... 8-)
  - NOW: some more BASIC select statement SYNTAX and SEMANTICS:
  - (we'll be ADDING to this in the next few weeks!)

NOW -- how about some SQL?

- SYNTAX:
- angle and square brackets below are NOT part of the syntax,
  - angle brackets are just surrounding a description
  - square brackets are just surrounding OPTIONAL parts

select [distinct] <one or more expressions, separated by commas>

from <1 or more table expressions, separated by commas>

[where <boolean expr>];

NOW -- how about some SQL?

- SYNTAX:
- angle and square brackets below are NOT part of the syntax,
  - angle brackets are just surrounding a description
  - square brackets are just surrounding OPTIONAL parts

select [distinct] <one or more expressions, separated by commas>

from <1 or more table expressions, separated by commas>

[where <boolean expr>];

NOW -- how about some SQL?

- SEMANTICS: [conceptually!! actual algorithms try to be more efficient!]

NOW -- how about some SQL?

- SEMANTICS: [conceptually!! actual algorithms try to be more efficient!]
  - STEP 1: take the CARTESIAN PRODUCT of the table expressions in the FROM clause
  - STEP 2: take a relation SELECTION of the rows resulting from STEP 1 for which the WHERE clause <boolean expr> is TRUE
  - STEP 3: take a projection of the columns/expressions in the SELECT clause, in the order they appear, JUST from the rows resulting from STEP 2 ...and only remove duplicate rows from the result IF DISTINCT is in the SELECT clause

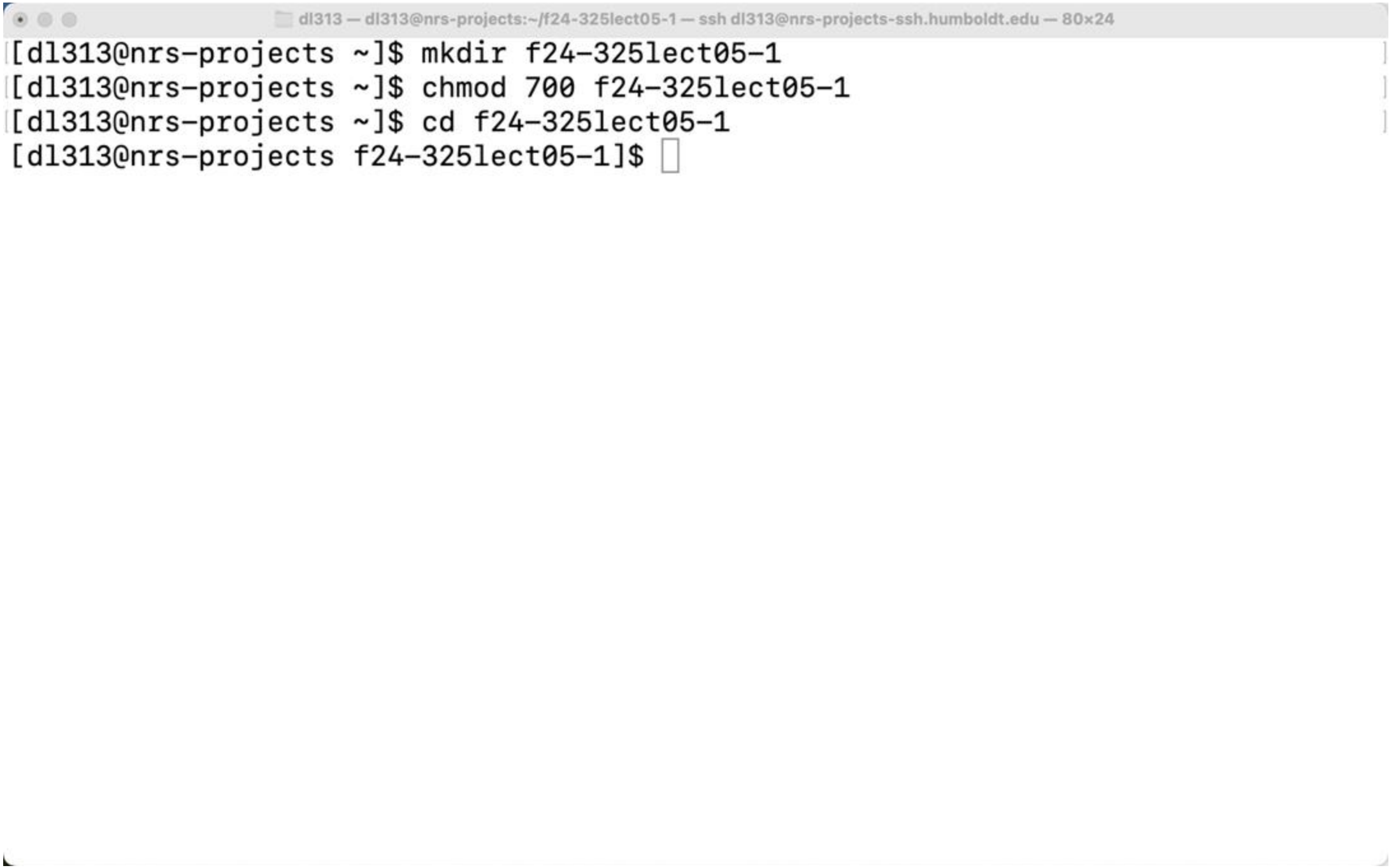
NOW -- how about some SQL?

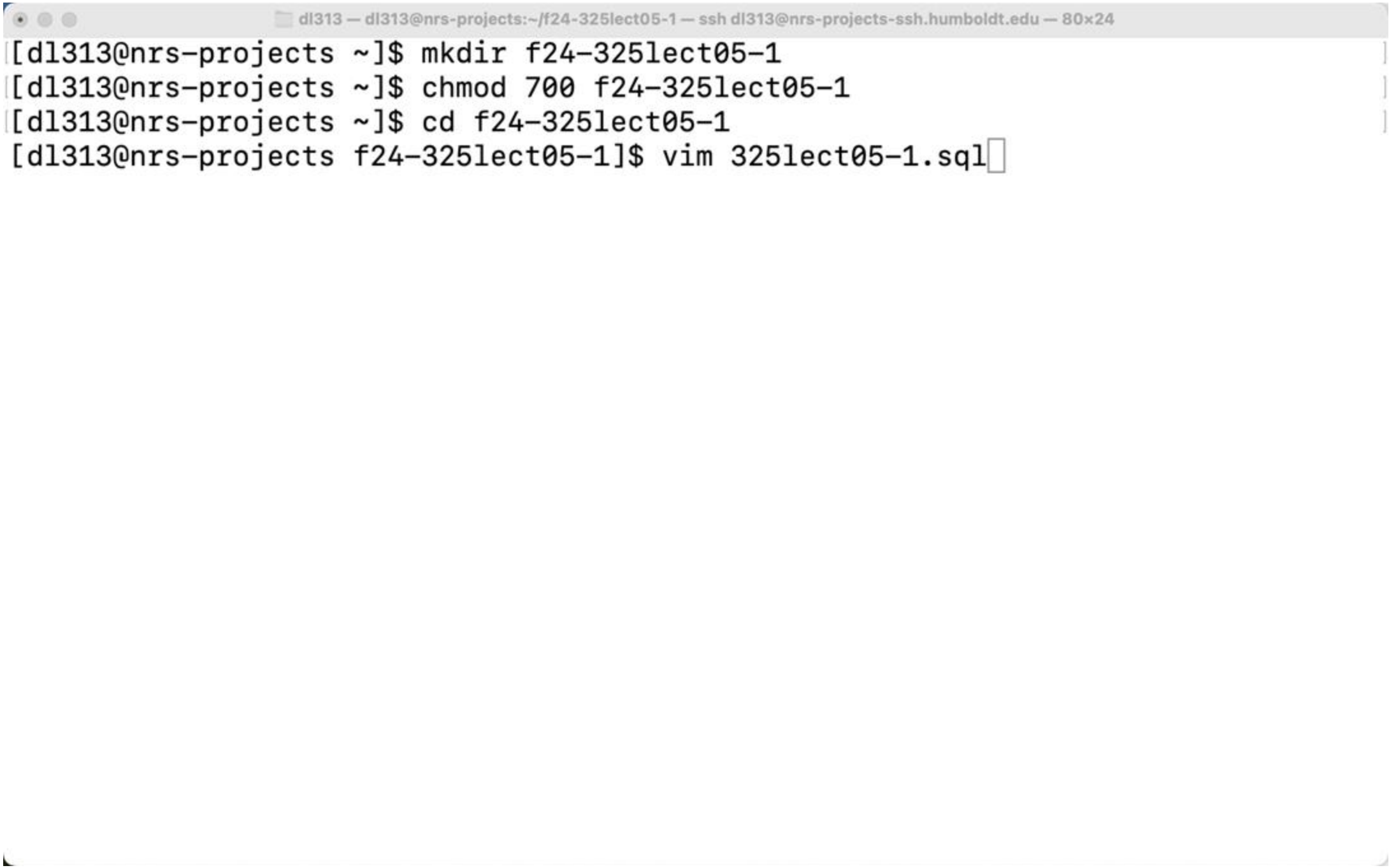
- SEMANTICS: [conceptually!! actual algorithms try to be more efficient!]
  - STEP 1: take the CARTESIAN PRODUCT of the table expressions in the FROM clause
  - STEP 2: take a relation SELECTION of the rows resulting from STEP 1 for which the WHERE clause <boolean expr> is TRUE
  - STEP 3: take a projection of the columns/expressions in the SELECT clause, in the order they appear, JUST from the rows resulting from STEP 2 ...and only remove duplicate rows from the result IF DISTINCT is in the SELECT clause
- Top-to-Bottom Writing, a different way of Execution

Question

What is the correct execution order of the clauses in a typical SQL SELECT query?

1) SELECT, FROM, WHERE
2) FROM, WHERE, SELECT
3) WHERE, SELECT, FROM
4) FROM, SELECT, WHERE


dl313 — dl313@nrs-projects:~/f24-325lect05-1 — ssh dl313@nrs-projects-ssh.humboldt.edu — 80×24
[dl313@nrs-projects ~]$ mkdir f24-325lect05-1
[dl313@nrs-projects ~]$ chmod 700 f24-325lect05-1
[dl313@nrs-projects ~]$ cd f24-325lect05-1
[dl313@nrs-projects f24-325lect05-1]$


dl313 — dl313@nrs-projects:~/f24-325lect05-1 — ssh dl313@nrs-projects-ssh.humboldt.edu — 80×24
[dl313@nrs-projects ~]$ mkdir f24-325lect05-1
[dl313@nrs-projects ~]$ chmod 700 f24-325lect05-1
[dl313@nrs-projects ~]$ cd f24-325lect05-1
[dl313@nrs-projects f24-325lect05-1]$ vim 325lect05-1.sql

f24-325lect05-1 — vim 325lect05-1.sql — 80×24

```
/*=====
    examples for Week 5 material
    (more detail: see SQL Reading Packet 3!!!!)
=====*/
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
-- INSERT --
```

f24-325lect05-1 — vim 325lect05-1.sql — 80×24

```
/*=====
    examples for Week 5 material
    (more detail: see SQL Reading Packet 3!!!!)
=====*/

/*=====
    SQL "GOTCHA" - how to select rows for which an attribute's value
    is nulli

    YOU CANNOT DO THIS WITH = !!!!!!!!!

    YOU MUST USE THE OPERATOR:   is null

    (likewise you can't use !=, you must use:   is not null
=====*/
~
~
~
~
~
~
~
~
-- INSERT --
```

f24-325lect05-1 — vim 325lect05-1.sql — 80×24

```
-- I want last names of employees whose commission attribute is null

prompt
prompt =====
prompt THIS DOES NOT WORK, there ARE empls whose commission is null
prompt =====

```

-- INSERT --

dl313 — dl313@nrs-projects:~/f24-325lect05-1 — ssh dl313@nrs-projects-ssh.humboldt.edu — 80×24

```
-- I want last names of employees whose commission attribute is null

prompt
prompt =====
prompt THIS DOES NOT WORK, there ARE empls whose commission is null
prompt =====

select empl_last_name
from   empl
where  commission = null;
```

"325lect05-1.sql" [New] 39L, 586C written 39,0-1 Bot

```
dl313 — dl313@nrs-projects:~/f24-325lect05-1 — ssh dl313@nrs-projects-ssh.humboldt.edu — 80×24
[dl313@nrs-projects f24-325lect05-1]$ sqlplus /

SQL*Plus: Release 19.0.0.0.0 - Production on Sun Sep 22 20:41:14 2024
Version 19.3.0.0.0

Copyright (c) 1982, 2019, Oracle.  All rights reserved.

Last Successful login time: Sun Sep 22 2024 20:24:21 -07:00

Connected to:
Oracle Database 19c Enterprise Edition Release 19.0.0.0.0 - Production
Version 19.3.0.0.0

SQL> @ 325lect05-1.sql
```

```
dl313 — dl313@nrs-projects:~/f24-325lect05-1 — ssh dl313@nrs-projects-ssh.humboldt.edu — 80×24
[dl313@nrs-projects f24-325lect05-1]$ sqlplus /

SQL*Plus: Release 19.0.0.0.0 - Production on Sun Sep 22 20:41:14 2024
Version 19.3.0.0.0

Copyright (c) 1982, 2019, Oracle.  All rights reserved.

Last Successful login time: Sun Sep 22 2024 20:24:21 -07:00

Connected to:
Oracle Database 19c Enterprise Edition Release 19.0.0.0.0 - Production
Version 19.3.0.0.0

SQL> @ 325lect05-1.sql

=====
THIS DOES NOT WORK, there ARE empls whose commission is null
=====

no rows selected

SQL>
```

```
dl313 — dl313@nrs-projects:~/f24-325lect05-1 — ssh dl313@nrs-projects-ssh.humboldt.edu — 80×24
[dl313@nrs-projects f24-325lect05-1]$ sqlplus /

SQL*Plus: Release 19.0.0.0.0 - Production on Sun Sep 22 20:41:14 2024
Version 19.3.0.0.0

Copyright (c) 1982, 2019, Oracle.  All rights reserved.

Last Successful login time: Sun Sep 22 2024 20:24:21 -07:00

Connected to:
Oracle Database 19c Enterprise Edition Release 19.0.0.0.0 - Production
Version 19.3.0.0.0

SQL> @ 325lect05-1.sql

=====
THIS DOES NOT WORK, there ARE empls whose commission is null
=====

no rows selected

SQL> select *
  2  from empl;
```

dl313 — dl313@nrs-projects:~/f24-325lect05-1 — ssh dl313@nrs-projects-ssh.humboldt.edu — 80×24

```
SQL> select *
  2  from empl;

EMPL EMPL_LAST_NAME  JOB_TITLE  MGR  HIREDATE      SALARY COMMISSION DEP
---- --------------- ---------- ---- --------- ---------- ---------- ---
7839 King            President       17-NOV-11       5000            500
7566 Jones           Manager    7839 02-APR-12       2975            200
7698 Blake           Manager    7839 01-MAY-13       2850            300
7782 Raimi           Manager    7839 09-JUN-12       2450            100
7902 Ford            Analyst    7566 03-DEC-12       3000            200
7369 Smith           Clerk      7902 17-DEC-12        800            200
7499 Michaels        Sales      7698 20-FEB-18       1600        300 300
7521 Ward            Sales      7698 22-FEB-19       1250        500 300
7654 Martin          Sales      7698 28-SEP-18       1250       1400 300
7788 Scott           Analyst    7566 09-NOV-18       3000            200
7844 Turner          Sales      7698 08-SEP-19       1500          0 300

EMPL EMPL_LAST_NAME  JOB_TITLE  MGR  HIREDATE      SALARY COMMISSION DEP
---- --------------- ---------- ---- --------- ---------- ---------- ---
7876 Adams           Clerk      7788 23-SEP-18       1100            400
7900 James           Clerk      7698 03-DEC-17        950            300
7934 Miller          Clerk      7782 23-JAN-16       1300            100
```

dl313 — dl313@nrs-projects:~/f24-325lect05-1 — ssh dl313@nrs-projects-ssh.humboldt.edu — 80×24

```
-- I want last names of employees whose commission attribute is null

prompt
prompt =====
prompt THIS DOES NOT WORK, there ARE empls whose commission is null
prompt =====

select empl_last_name
from   empl
where  commission = null;

prompt =====
prompt use IS NULL to get empls whose commission is null
prompt =====

select empl_last_name
from   empl
where  commission is null;
```

-- INSERT (paste) -- 34,27 72%

dl313 — dl313@nrs-projects:~/f24-325lect05-1 — ssh dl313@nrs-projects-ssh.humboldt.edu — 80×24

```
EMPL EMPL_LAST_NAME   JOB_TITLE   MGR  HIREDATE       SALARY COMMISSION DEP
---- ---------------- ---------- ---- --------- ---------- ---------- ---
7839 King             President       17-NOV-11        5000                500
7566 Jones            Manager    7839 02-APR-12        2975                200
7698 Blake            Manager    7839 01-MAY-13        2850                300
7782 Raimi            Manager    7839 09-JUN-12        2450                100
7902 Ford             Analyst    7566 03-DEC-12        3000                200
7369 Smith            Clerk      7902 17-DEC-12         800                200
7499 Michaels         Sales      7698 20-FEB-18        1600         300    300
7521 Ward             Sales      7698 22-FEB-19        1250         500    300
7654 Martin           Sales      7698 28-SEP-18        1250        1400    300
7788 Scott            Analyst    7566 09-NOV-18        3000                200
7844 Turner           Sales      7698 08-SEP-19        1500           0    300

EMPL EMPL_LAST_NAME   JOB_TITLE   MGR  HIREDATE       SALARY COMMISSION DEP
---- ---------------- ---------- ---- --------- ---------- ---------- ---
7876 Adams            Clerk      7788 23-SEP-18        1100                400
7900 James            Clerk      7698 03-DEC-17         950                300
7934 Miller           Clerk      7782 23-JAN-16        1300                100

14 rows selected.

SQL> @ 325lect05-1.sql
```

dl313 — dl313@nrs-projects:~/f24-325lect05-1 — ssh dl313@nrs-projects-ssh.humboldt.edu — 80×24

```
=====

no rows selected

=====
use IS NULL to get empls whose commission is null
=====

EMPL_LAST_NAME
---------------
King
Jones
Blake
Raimi
Ford
Smith
Scott
Adams
James
Miller

10 rows selected.

SQL>
```

dl313 — dl313@nrs-projects:~/f24-325lect05-1 — ssh dl313@nrs-projects-ssh.humboldt.edu — 80×24

```
EMPL EMPL_LAST_NAME   JOB_TITLE   MGR  HIREDATE       SALARY COMMISSION DEP
---- ---------------- ---------- ---- --------- ---------- ---------- ---
7839 King             President       17-NOV-11        5000                500
7566 Jones            Manager    7839 02-APR-12        2975                200
7698 Blake            Manager    7839 01-MAY-13        2850                300
7782 Raimi            Manager    7839 09-JUN-12        2450                100
7902 Ford             Analyst    7566 03-DEC-12        3000                200
7369 Smith            Clerk      7902 17-DEC-12         800                200
7499 Michaels         Sales      7698 20-FEB-18        1600         300    300
7521 Ward             Sales      7698 22-FEB-19        1250         500    300
7654 Martin           Sales      7698 28-SEP-18        1250        1400    300
7788 Scott            Analyst    7566 09-NOV-18        3000                200
7844 Turner           Sales      7698 08-SEP-19        1500           0    300

EMPL EMPL_LAST_NAME   JOB_TITLE   MGR  HIREDATE       SALARY COMMISSION DEP
---- ---------------- ---------- ---- --------- ---------- ---------- ---
7876 Adams            Clerk      7788 23-SEP-18        1100                400
7900 James            Clerk      7698 03-DEC-17         950                300
7934 Miller           Clerk      7782 23-JAN-16        1300                100

14 rows selected.

SQL> @ 325lect05-1.sql
```

dl313 — dl313@nrs-projects:~/f24-325lect05-1 — ssh dl313@nrs-projects-ssh.humboldt.edu — 80×24

```
prompt =====
prompt THIS DOES NOT WORK, there ARE empls whose commission is NOT null
prompt =====

select empl_last_name
from   empl
where  commission != null;

prompt =====
prompt use IS NOT NULL to get empls whose commission is NOT null
prompt =====

select empl_last_name, commission
from   empl
where  commission is not null;
```

-- INSERT -- 52,1 89%

```
Adams
James
Miller

10 rows selected.

=====
THIS DOES NOT WORK, there ARE empls whose commission is NOT null
=====

no rows selected

=====
use IS NOT NULL to get empls whose commission is NOT null
=====

EMPL_LAST_NAME  COMMISSION
--------------- ----------
Michaels               300
Ward                   500
Martin                1400
Turner                   0

SQL>
```

dl313 — dl313@nrs-projects:~/f24-325lect05-1 — ssh dl313@nrs-projects-ssh.humboldt.edu — 80×24

```
prompt =====
prompt THIS DOES NOT WORK, there ARE empls whose commission is NOT null
prompt =====

select empl_last_name
from   empl
where  commission != null;

prompt =====
prompt use IS NOT NULL to get empls whose commission is NOT null
prompt =====

select empl_last_name, commission
from   empl
where  commission is not null;
```

-- INSERT -- 52,1 89%

Question

In SQL, if you want to retrieve rows where an employee's commission is not null, which statement would you use?

1) where commission = null
2) where commission is not null
3) where commission != null
4) where commission == null

dl313 — dl313@nrs-projects:~/f24-325lect05-1 — ssh dl313@nrs-projects-ssh.humboldt.edu — 80×24

```
/*=====
the IN predicate

    attrib IN (comma-sep'd list of values)

    this will be true for rows whose value of attrib IS
    one of those within (comma-sep'd list of values)
======*/
```

-- INSERT -- 69,1 92%

dl313 — dl313@nrs-projects:~/f24-325lect05-1 — ssh dl313@nrs-projects-ssh.humboldt.edu — 80×24

```
/*=====
the IN predicate

    attrib IN (comma-sep'd list of values)

    this will be true for rows whose value of attrib IS
    one of those within (comma-sep'd list of values)
======*/

prompt =====
prompt want this info about empls who are Analysts or Managers
prompt =====

select empl_last_name, job_title, salary
from   empl
where  job_title IN ('Analyst', 'Manager');
```

-- INSERT (paste) -- 67,44 83%

```
James
Miller

10 rows selected.

=====
THIS DOES NOT WORK, there ARE empls whose commission is NOT null
=====

no rows selected

=====
use IS NOT NULL to get empls whose commission is NOT null
=====

EMPL_LAST_NAME  COMMISSION
--------------- ----------
Michaels               300
Ward                   500
Martin                1400
Turner                   0

SQL> @ 325lect05-1.sql
```

dl313 — dl313@nrs-projects:~/f24-325lect05-1 — ssh dl313@nrs-projects-ssh.humboldt.edu — 80×24

```
=====
use IS NOT NULL to get empls whose commission is NOT null
=====

EMPL_LAST_NAME  COMMISSION
--------------- ----------
Michaels               300
Ward                   500
Martin                1400
Turner                   0

=====
want this info about empls who are Analysts or Managers
=====

EMPL_LAST_NAME  JOB_TITLE      SALARY
--------------- ---------- ----------
Jones           Manager          2975
Blake           Manager          2850
Raimi           Manager          2450
Ford            Analyst          3000
Scott           Analyst          3000

SQL>
```

dl313 — dl313@nrs-projects:~/f24-325lect05-1 — ssh dl313@nrs-projects-ssh.humboldt.edu — 80×24

```
/*=====
    we FREQUENTLY do further projections from equi-joins!

    if I just want the empl_last_name and dept_name from
    the equi-join of empl and dept,
    JUST project those columns!
=====*/
```

-- INSERT -- 85,1 84%

```
=====
use IS NOT NULL to get empls whose commission is NOT null
=====

EMPL_LAST_NAME  COMMISSION
--------------- ----------
Michaels               300
Ward                   500
Martin                1400
Turner                   0

=====
want this info about empls who are Analysts or Managers
=====

EMPL_LAST_NAME  JOB_TITLE      SALARY
--------------- ---------- ----------
Jones           Manager          2975
Blake           Manager          2850
Raimi           Manager          2450
Ford            Analyst          3000
Scott           Analyst          3000

SQL> describe empl
```

dl313 — dl313@nrs-projects:~/f24-325lect05-1 — ssh dl313@nrs-projects-ssh.humboldt.edu — 80×24

```
want this info about empls who are Analysts or Managers
=====

EMPL_LAST_NAME  JOB_TITLE      SALARY
--------------- ---------- ----------
Jones           Manager          2975
Blake           Manager          2850
Raimi           Manager          2450
Ford            Analyst          3000
Scott           Analyst          3000

SQL> describe empl
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 EMPL_NUM                                  NOT NULL CHAR(4)
 EMPL_LAST_NAME                            NOT NULL VARCHAR2(15)
 JOB_TITLE                                          VARCHAR2(10)
 MGR                                                CHAR(4)
 HIREDATE                                  NOT NULL DATE
 SALARY                                             NUMBER(6,2)
 COMMISSION                                         NUMBER(6,2)
 DEPT_NUM                                           CHAR(3)

SQL>
```

```
want this info about empls who are Analysts or Managers
=====

EMPL_LAST_NAME  JOB_TITLE      SALARY
--------------- ---------- ----------
Jones           Manager          2975
Blake           Manager          2850
Raimi           Manager          2450
Ford            Analyst          3000
Scott           Analyst          3000

SQL> describe empl
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 EMPL_NUM                                  NOT NULL CHAR(4)
 EMPL_LAST_NAME                            NOT NULL VARCHAR2(15)
 JOB_TITLE                                          VARCHAR2(10)
 MGR                                                CHAR(4)
 HIREDATE                                  NOT NULL DATE
 SALARY                                             NUMBER(6,2)
 COMMISSION                                         NUMBER(6,2)
 DEPT_NUM                                           CHAR(3)

SQL> describe dept
```

dl313 — dl313@nrs-projects:~/f24-325lect05-1 — ssh dl313@nrs-projects-ssh.humboldt.edu — 80×24

```
Raimi                Manager              2450
Ford                 Analyst              3000
Scott                Analyst              3000

SQL> describe empl
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 EMPL_NUM                                  NOT NULL CHAR(4)
 EMPL_LAST_NAME                            NOT NULL VARCHAR2(15)
 JOB_TITLE                                          VARCHAR2(10)
 MGR                                                CHAR(4)
 HIREDATE                                  NOT NULL DATE
 SALARY                                             NUMBER(6,2)
 COMMISSION                                         NUMBER(6,2)
 DEPT_NUM                                           CHAR(3)

SQL> describe dept
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 DEPT_NUM                                  NOT NULL CHAR(3)
 DEPT_NAME                                 NOT NULL VARCHAR2(15)
 DEPT_LOC                                  NOT NULL VARCHAR2(15)

SQL>
```

```
/*=====
    we FREQUENTLY do further projections from equi-joins!

    if I just want the empl_last_name and dept_name from
    the equi-join of empl and dept,
    JUST project those columns!
=====*/

prompt =====
prompt projecting JUST empl_last_name and dept_name from the equi-join
prompt of empl and dept
prompt =====

select empl_last_name, dept_name
from   empl, dept
where  empl.dept_num = dept.dept_num;
```

dl313 — dl313@nrs-projects:~/f24-325lect05-1 — ssh dl313@nrs-projects-ssh.humboldt.edu — 80×24

```
Raimi                Manager              2450
Ford                 Analyst              3000
Scott                Analyst              3000

SQL> describe empl
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 EMPL_NUM                                  NOT NULL CHAR(4)
 EMPL_LAST_NAME                            NOT NULL VARCHAR2(15)
 JOB_TITLE                                          VARCHAR2(10)
 MGR                                                CHAR(4)
 HIREDATE                                  NOT NULL DATE
 SALARY                                             NUMBER(6,2)
 COMMISSION                                         NUMBER(6,2)
 DEPT_NUM                                           CHAR(3)

SQL> describe dept
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 DEPT_NUM                                  NOT NULL CHAR(3)
 DEPT_NAME                                 NOT NULL VARCHAR2(15)
 DEPT_LOC                                  NOT NULL VARCHAR2(15)

SQL> @ 325lect05-1.sql
```

```
EMPL_LAST_NAME  DEPT_NAME
--------------- ---------------
Miller          Accounting
Raimi           Accounting
Scott           Research
Jones           Research
Ford            Research
Smith           Research
Martin          Sales
Ward            Sales
Blake           Sales
Michaels        Sales
James           Sales

EMPL_LAST_NAME  DEPT_NAME
--------------- ---------------
Turner          Sales
Adams           Operations
King            Management

14 rows selected.

SQL>
```

Question

Which SQL statement correctly uses the IN operator to find employees whose job title is either 'Analyst' or 'Manager’?

A) select * from empl where job_title = 'Analyst' or 'Manager'
B) select * from empl where job_title IN ('Analyst', 'Manager')
C) select * from empl where job_title IS ('Analyst', 'Manager')
D) select * from empl where job_title = 'Analyst' AND job_title = 'Manager'

dl313 — dl313@nrs-projects:~/f24-325lect05-1 — ssh dl313@nrs-projects-ssh.humboldt.edu — 80×24

```
/*=====
    BUT!
    the SCOPE of a SELECT statement, if you will,
    is the table(s) from its FROM clause....

    IF an attribute has the same name in MORE than one table expr
    in the FROM clause, you MUST precede it by the name of the table
    you want it from;
=====*/
```

-- INSERT -- 102,1 87%

dl313 — dl313@nrs-projects:~/f24-325lect05-1 — ssh dl313@nrs-projects-ssh.humboldt.edu — 80×24

```
/*=====
    BUT!
    the SCOPE of a SELECT statement, if you will,
    is the table(s) from its FROM clause....

    IF an attribute has the same name in MORE than one table expr
    in the FROM clause, you MUST precede it by the name of the table
    you want it from;
=====*/

prompt ========
prompt WILL GET ERROR!
prompt dept_num is in dept and empl, MUST say which table''s version you want
prompt ========

select empl_last_name, dept_num, dept_name
from   empl, dept
where  empl.dept_num = dept.dept_num;
```

-- INSERT -- 103,38 82%

dl313 — dl313@nrs-projects:~/f24-325lect05-1 — ssh dl313@nrs-projects-ssh.humboldt.edu — 80×24

```
EMPL_LAST_NAME  DEPT_NAME
--------------- ---------------
Miller          Accounting
Raimi           Accounting
Scott           Research
Jones           Research
Ford            Research
Smith           Research
Martin          Sales
Ward            Sales
Blake           Sales
Michaels        Sales
James           Sales

EMPL_LAST_NAME  DEPT_NAME
--------------- ---------------
Turner          Sales
Adams           Operations
King            Management

14 rows selected.

SQL> @ 325lect05-1.sql
```

```
Ward            Sales
Blake           Sales
Michaels        Sales
James           Sales

EMPL_LAST_NAME  DEPT_NAME
--------------- ---------------
Turner          Sales
Adams           Operations
King            Management

14 rows selected.

========
WILL GET ERROR!
dept_num is in dept and empl, MUST say which table''s version you want
========
select empl_last_name, dept_num, dept_name
                       *
ERROR at line 1:
ORA-00918: column ambiguously defined


SQL>
```

dl313 — dl313@nrs-projects:~/f24-325lect05-1 — ssh dl313@nrs-projects-ssh.humboldt.edu — 80×24

```
prompt ========
prompt WILL GET ERROR!
prompt dept_num is in dept and empl, MUST say which table''s version you want
prompt ========

select empl_last_name, dept_num, dept_name
from   empl, dept
where  empl.dept_num = dept.dept_num;

prompt =====
prompt indicate which table''s version of dept_num you want, and that works:
prompt =====

select empl_last_name, empl.dept_num, dept_name
from   empl, dept
where  empl.dept_num = dept.dept_num;
```

-- INSERT -- 113,1 84%

```
Ward                 Sales
Blake                Sales
Michaels             Sales
James                Sales

EMPL_LAST_NAME  DEPT_NAME
--------------- ---------------
Turner          Sales
Adams           Operations
King            Management

14 rows selected.

========
WILL GET ERROR!
dept_num is in dept and empl, MUST say which table''s version you want
========
select empl_last_name, dept_num, dept_name
                       *
ERROR at line 1:
ORA-00918: column ambiguously defined


SQL> @ 325lect05-1.sql
```

```
=====
indicate which table''s version of dept_num you want, and that works:
=====

EMPL_LAST_NAME  DEP DEPT_NAME
--------------- --- --------------
Miller          100 Accounting
Raimi           100 Accounting
Scott           200 Research
Jones           200 Research
Ford            200 Research
Smith           200 Research
Martin          300 Sales
Ward            300 Sales
Blake           300 Sales
Michaels        300 Sales
James           300 Sales

EMPL_LAST_NAME  DEP DEPT_NAME
--------------- --- --------------
Turner          300 Sales
Adams           400 Operations
King            500 Management
```

dl313 — dl313@nrs-projects:~/f24-325lect05-1 — ssh dl313@nrs-projects-ssh.humboldt.edu — 80×24

```
EMPL_LAST_NAME  DEP DEPT_NAME
--------------- --- ---------------
Miller          100 Accounting
Raimi           100 Accounting
Scott           200 Research
Jones           200 Research
Ford            200 Research
Smith           200 Research
Martin          300 Sales
Ward            300 Sales
Blake           300 Sales
Michaels        300 Sales
James           300 Sales

EMPL_LAST_NAME  DEP DEPT_NAME
--------------- --- ---------------
Turner          300 Sales
Adams           400 Operations
King            500 Management

14 rows selected.

SQL>
```

dl313 — dl313@nrs-projects:~/f24-325lect05-1 — ssh dl313@nrs-projects-ssh.humboldt.edu — 80×24

```
/*=====
    AND OR NOT - are supported by SQL

    can use these in the WHERE clause
    (because you can only HAVE ONE WHERE clause in a SELECT statement!)

    select blah
    from   blah
    where  (bool_expr)
           AND (bool_expr);

    select blah
    from   blah
    where  (bool_expr)
           OR (bool_expr);

    (and CAN have as "big" an expression as you want built
    from these...

    where (bool_expr)
          AND (bool_expr)
          AND (bool_expr)
          ... etc.! )
-- INSERT --                                          130,3         82%
```

dl313 — dl313@nrs-projects:~/f24-325lect05-1 — ssh dl313@nrs-projects-ssh.humboldt.edu — 80×24

```
/*=====
    For example, you want to have further selection
    of rows from an equi-join, you will typically indicate
    this within a SQL SELECT statement by using logical AND
    within the WHERE clause.
=====*/
~
~
~
~
~
                                                    11,59         All
```

dl313 — dl313@nrs-projects:~/f24-325lect05-1 — ssh dl313@nrs-projects-ssh.humboldt.edu — 80×24

```
/*=====
    For example, you want to have further selection
    of rows from an equi-join, you will typically indicate
    this within a SQL SELECT statement by using logical AND
    within the WHERE clause.
=====*/

prompt =====
prompt Say you would like employee last names,
prompt the name of their departments, and their department
prompt location only for employees hired since December 1st 2012.
prompt =====

select empl_last_name, dept_name, dept_loc, hiredate
from   empl, dept
where  empl.dept_num = dept.dept_num
       AND hiredate > '01-dec-2012';
~
~
~
~
~
~
                                                    11,59         All
```

dl313 — dl313@nrs-projects:~/f24-325lect05-1 — ssh dl313@nrs-projects-ssh.humboldt.edu — 80×24

```
SQL> @ test2.sql
=====
Say you would like employee last names,
the name of their departments, and their department
location only for employees hired since 12-1-1991.
=====

EMPL_LAST_NAME  DEPT_NAME      DEPT_LOC        HIREDATE
--------------- --------------- --------------- ---------
Miller          Accounting      New York        23-JAN-16
Ford            Research        Dallas          03-DEC-12
Scott           Research        Dallas          09-NOV-18
Smith           Research        Dallas          17-DEC-12
Turner          Sales           Chicago         08-SEP-19
Ward            Sales           Chicago         22-FEB-19
Michaels        Sales           Chicago         20-FEB-18
Blake           Sales           Chicago         01-MAY-13
Martin          Sales           Chicago         28-SEP-18
James           Sales           Chicago         03-DEC-17
Adams           Operations      Boston          23-SEP-18

11 rows selected.

SQL>
```

dl313 — dl313@nrs-projects:~/f24-325lect05-1 — ssh dl313@nrs-projects-ssh.humboldt.edu — 80×24

```
prompt =====
prompt So, instead of doing this using the predicate IN,
prompt =====

select empl_last_name, job_title, salary
from   empl
where  job_title IN ('Analyst', 'Manager');

prompt =====
prompt We can also use OR for that,
prompt =====

select empl_last_name, job_title, salary
from   empl
where  job_title = 'Analyst'
       or job_title = 'Manager';
~
~
~
~
~
~
~
~
"test.sql" [New] 16L, 360C written                          16,32         All
```

dl313 — dl313@nrs-projects:~/f24-325lect05-1 — ssh dl313@nrs-projects-ssh.humboldt.edu — 80×24

So, instead of doing this using the predicate IN,

=====

EMPL_LAST_NAME	JOB_TITLE	SALARY
Jones	Manager	2975
Blake	Manager	2850
Raimi	Manager	2450
Ford	Analyst	3000
Scott	Analyst	3000

=====

We can also use OR for that,

=====

EMPL_LAST_NAME	JOB_TITLE	SALARY
Jones	Manager	2975
Blake	Manager	2850
Raimi	Manager	2450
Ford	Analyst	3000
Scott	Analyst	3000

SQL>

Question

What happens if an attribute in a SELECT statement exists in more than one table in the FROM clause?
A) SQL will automatically pick the attribute from the first table listed.
B) SQL will return an error because the attribute is ambiguous.
C) SQL will randomly select one of the tables.
D) SQL will ignore the ambiguity and return both versions of the attribute.

dl313 — dl313@nrs-projects:~/f24-325lect05-1 — ssh dl313@nrs-projects-ssh.humboldt.edu — 80×24

```
/*=====
    Be careful when you combine AND and OR within the same SQL SELECT statement
    -- to make it perfectly clear what is being AND'ed and what is being OR'ed,
    you should use parentheses to make that explicitly clear.
=====*/
```

"test2.sql" 44L, 849C written 38,0-1 95%

dl313 — dl313@nrs-projects:~/f24-325lect05-1 — ssh dl313@nrs-projects-ssh.humboldt.edu — 80×24

```
/*=====
    Be careful when you combine AND and OR within the same SQL SELECT statement
    -- to make it perfectly clear what is being AND'ed and what is being OR'ed,
    you should use parentheses to make that explicitly clear.
=====*/

prompt =====
prompt For example, if I want the names and hiredates of only employees
prompt hired after March 1, 2015, who are also either sales people or
prompt make $1500 or more,
prompt =====

select empl_last_name, hiredate
from   empl
where  hiredate > '01-Mar-2015'
       and (job_title = 'Salesman'
            or salary >= 1500);
```

"test2.sql" 54L, 1189C written 38,31 64%

dl313 — dl313@nrs-projects:~/f24-325lect05-1 — ssh dl313@nrs-projects-ssh.humboldt.edu — 80×24

```
SQL> @ test2.sql
=====
For example, if I want the names and hiredates of only employees
hired after March 1, 2015, who are also either sales people or
make $1500 or more,
=====

EMPL_LAST_NAME  HIREDATE
--------------- ---------
Michaels        20-FEB-18
Scott           09-NOV-18
Turner          08-SEP-19

SQL>
```

dl313 — dl313@nrs-projects:~/f24-325lect05-1 — ssh dl313@nrs-projects-ssh.humboldt.edu — 80×24

```
    Be careful when you combine AND and OR within the same SQL SELECT statement
    -- to make it perfectly clear what is being AND'ed and what is being OR'ed,
    you should use parentheses to make that explicitly clear.
=====*/

prompt =====
prompt For example, if I want the names and hiredates of only employees
prompt hired after March 1, 2015, who are also either sales people or
prompt make $1500 or more,
prompt =====

select empl_last_name, hiredate
from   empl
where  hiredate > '01-Mar-2015'
       and (job_title = 'Salesman'
            or salary >= 1500);
```

-- VISUAL BLOCK -- 11x1 38,1 70%

dl313 — dl313@nrs-projects:~/f24-325lect05-1 — ssh dl313@nrs-projects-ssh.humboldt.edu — 80×24

```
    Be careful when you combine AND and OR within the same SQL SELECT statement
    -- to make it perfectly clear what is being AND'ed and what is being OR'ed,
    you should use parentheses to make that explicitly clear.
=====*/

--prompt =====
prompt For example, if I want the names and hiredates of only employees
prompt hired after March 1, 2015, who are also either sales people or
prompt make $1500 or more,
prompt =====

select empl_last_name, hiredate
from   empl
where  hiredate > '01-Mar-2015'
       and (job_title = 'Salesman'
            or salary >= 1500);
```

-- INSERT -- 28,3 70%

dl313 — dl313@nrs-projects:~/f24-325lect05-1 — ssh dl313@nrs-projects-ssh.humboldt.edu — 80×24

```
    Be careful when you combine AND and OR within the same SQL SELECT statement
    -- to make it perfectly clear what is being AND'ed and what is being OR'ed,
    you should use parentheses to make that explicitly clear.
=====*/

--prompt =====
--prompt For example, if I want the names and hiredates of only employees
--prompt hired after March 1, 2015, who are also either sales people or
--prompt make $1500 or more,
--prompt =====
--
--select empl_last_name, hiredate
--from   empl
--where  hiredate > '01-Mar-2015'
--       and (job_title = 'Salesman'
--            or salary >= 1500);
```

28,1 70%

dl313 — dl313@nrs-projects:~/f24-325lect05-1 — ssh dl313@nrs-projects-ssh.humboldt.edu — 80×24

```
/*======
    As an example of the logical NOT operator,
======*/

prompt =====
prompt For example, if I want select those employee
prompt rows for employees who are not sales people:
prompt =====

select *
from   empl
where  not job_title = 'Sales';
```

"test2.sql" 73L, 1467C written 63,0-1 96%

dl313 — dl313@nrs-projects:~/f24-325lect05-1 — ssh dl313@nrs-projects-ssh.humboldt.edu — 80×24

```
SQL> @ test2.sql
=====
For example, if I want select those employee
rows for employees who are not sales people:
=====

EMPL EMPL_LAST_NAME  JOB_TITLE  MGR  HIREDATE       SALARY COMMISSION DEP
---- --------------- ---------- ---- --------- ---------- ---------- ---
7839 King            President       17-NOV-11        5000                500
7566 Jones           Manager    7839 02-APR-12        2975                200
7698 Blake           Manager    7839 01-MAY-13        2850                300
7782 Raimi           Manager    7839 09-JUN-12        2450                100
7902 Ford            Analyst    7566 03-DEC-12        3000                200
7369 Smith           Clerk      7902 17-DEC-12         800                200
7788 Scott           Analyst    7566 09-NOV-18        3000                200
7876 Adams           Clerk      7788 23-SEP-18        1100                400
7900 James           Clerk      7698 03-DEC-17         950                300
7934 Miller          Clerk      7782 23-JAN-16        1300                100

10 rows selected.

SQL>
```

dl313 — dl313@nrs-projects:~/f24-325lect05-1 — ssh dl313@nrs-projects-ssh.humboldt.edu — 80×24

```
SQL> @ test2.sql
=====
For example, if I want select those employee
rows for employees who are not sales people:
=====

EMPL EMPL_LAST_NAME  JOB_TITLE  MGR  HIREDATE       SALARY COMMISSION DEP
---- --------------- ---------- ---- --------- ---------- ---------- ---
7839 King            President       17-NOV-11        5000                500
7566 Jones           Manager    7839 02-APR-12        2975                200
7698 Blake           Manager    7839 01-MAY-13        2850                300
7782 Raimi           Manager    7839 09-JUN-12        2450                100
7902 Ford            Analyst    7566 03-DEC-12        3000                200
7369 Smith           Clerk      7902 17-DEC-12         800                200
7788 Scott           Analyst    7566 09-NOV-18        3000                200
7876 Adams           Clerk      7788 23-SEP-18        1100                400
7900 James           Clerk      7698 03-DEC-17         950                300
7934 Miller          Clerk      7782 23-JAN-16        1300                100

10 rows selected.

SQL> clear screen;
```


dl313 — dl313@nrs-projects:~/f24-325lect05-1 — ssh dl313@nrs-projects-ssh.humboldt.edu — 80×24
SQL>

dl313 — dl313@nrs-projects:~/f24-325lect05-1 — ssh dl313@nrs-projects-ssh.humboldt.edu — 80×24

```
=====*/

prompt =====
prompt For example, if I want select those employee
prompt rows for employees who are not sales people:
prompt =====

select *
from   empl
where  not job_title = 'Sales';

prompt =====
prompt You can also use both <> or != as "not equal" operators,
prompt =====

select *
from   empl
where  job_title <> 'Sales';

select *
from empl
where job_title != 'Sales';

-- INSERT --                                                57,13         87%
```

```
=====
You can also use both <> or != as "not equal" operators,
=====

EMPL EMPL_LAST_NAME  JOB_TITLE  MGR  HIREDATE      SALARY COMMISSION DEP
---- --------------- ---------- ---- --------- ---------- ---------- ---
7839 King            President       17-NOV-11        5000            500
7566 Jones           Manager    7839 02-APR-12        2975            200
7698 Blake           Manager    7839 01-MAY-13        2850            300
7782 Raimi           Manager    7839 09-JUN-12        2450            100
7902 Ford            Analyst    7566 03-DEC-12        3000            200
7369 Smith           Clerk      7902 17-DEC-12         800            200
7788 Scott           Analyst    7566 09-NOV-18        3000            200
7876 Adams           Clerk      7788 23-SEP-18        1100            400
7900 James           Clerk      7698 03-DEC-17         950            300
7934 Miller          Clerk      7782 23-JAN-16        1300            100

10 rows selected.


EMPL EMPL_LAST_NAME  JOB_TITLE  MGR  HIREDATE      SALARY COMMISSION DEP
---- --------------- ---------- ---- --------- ---------- ---------- ---
7839 King            President       17-NOV-11        5000            500
7566 Jones           Manager    7839 02-APR-12        2975            200
```

dl313 — dl313@nrs-projects:~/f24-325lect05-1 — ssh dl313@nrs-projects-ssh.humboldt.edu — 80×24

```
/*=====
    BETWEEN operator CAN be used in WHERE clauses, also

    (we used to restrict an attribute's domain in a create table
    statement earlier this semester)
=====*/
```

-- INSERT -- 154,1 92%

dl313 — dl313@nrs-projects:~/f24-325lect05-1 — ssh dl313@nrs-projects-ssh.humboldt.edu — 80×24

```
/*=====
    BETWEEN operator CAN be used in WHERE clauses, also

    (we used to restrict an attribute's domain in a create table
    statement earlier this semester)
=====*/

prompt =====
prompt projecting JUST empl_last_name and salary for empls
prompt whose salary is between 1100 and 1600 inclusive:
prompt =====

select empl_last_name, salary
from   empl
where  salary BETWEEN 1100 and 1600;
```

-- INSERT -- 154,1 92%

dl313 — dl313@nrs-projects:~/f24-325lect05-1 — ssh dl313@nrs-projects-ssh.humboldt.edu — 80×24

```
EMPL_LAST_NAME  DEP DEPT_NAME
--------------- --- ---------------
Miller          100 Accounting
Raimi           100 Accounting
Scott           200 Research
Jones           200 Research
Ford            200 Research
Smith           200 Research
Martin          300 Sales
Ward            300 Sales
Blake           300 Sales
Michaels        300 Sales
James           300 Sales

EMPL_LAST_NAME  DEP DEPT_NAME
--------------- --- ---------------
Turner          300 Sales
Adams           400 Operations
King            500 Management

14 rows selected.

SQL> @ 325lect05-1.sql
```

dl313 — dl313@nrs-projects:~/f24-325lect05-1 — ssh dl313@nrs-projects-ssh.humboldt.edu — 80×24

```
--------------- --- ---------------
Turner          300 Sales
Adams           400 Operations
King            500 Management

14 rows selected.

=====
projecting JUST empl_last_name and salary for empls
whose salary is between 1100 and 1600 inclusive:
=====

EMPL_LAST_NAME      SALARY
--------------- ----------
Michaels              1600
Ward                  1250
Martin                1250
Turner                1500
Adams                 1100
Miller                1300

6 rows selected.

SQL>
```

Question

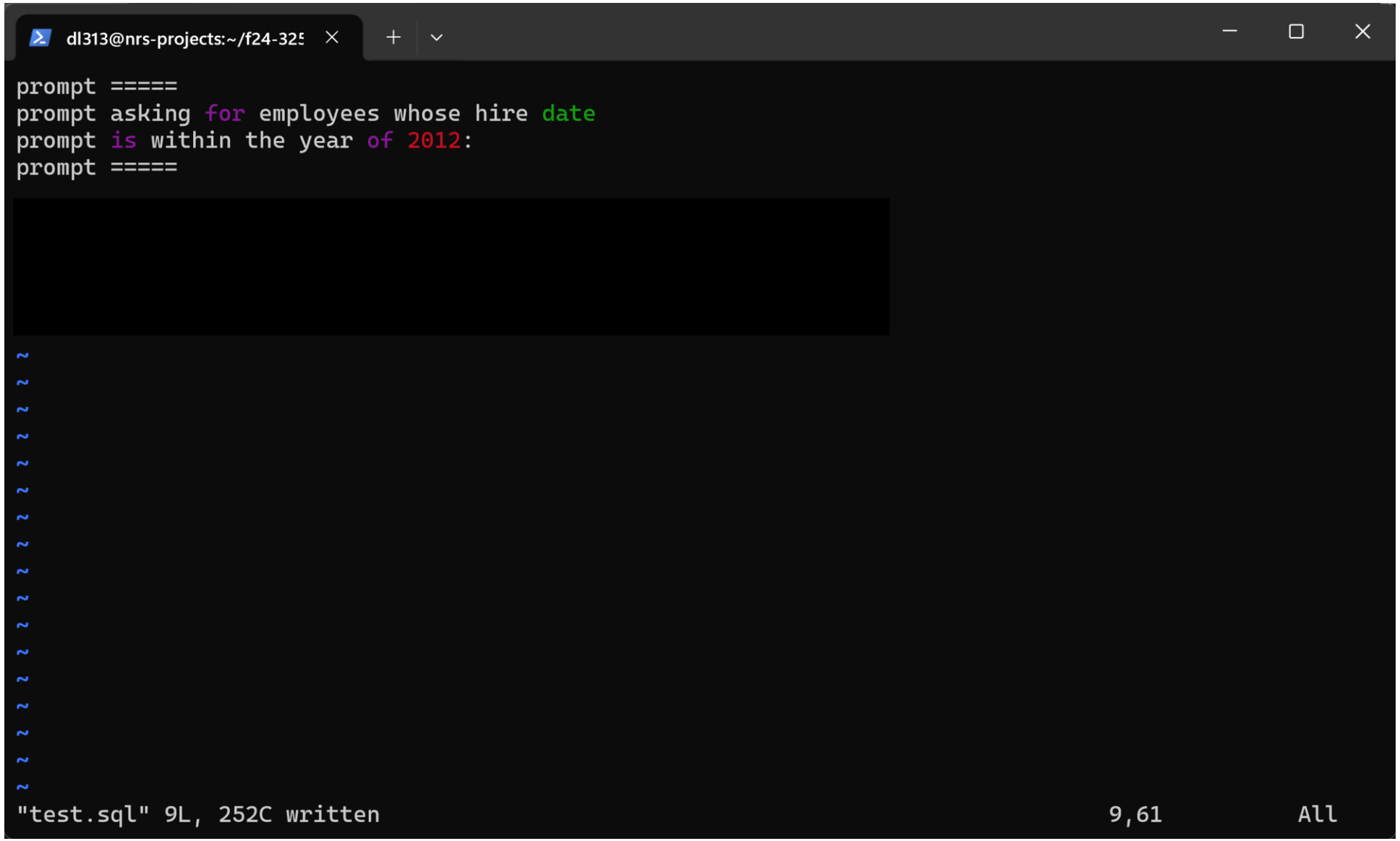
How do you select rows where a column is NOT equal to a specific value in SQL?

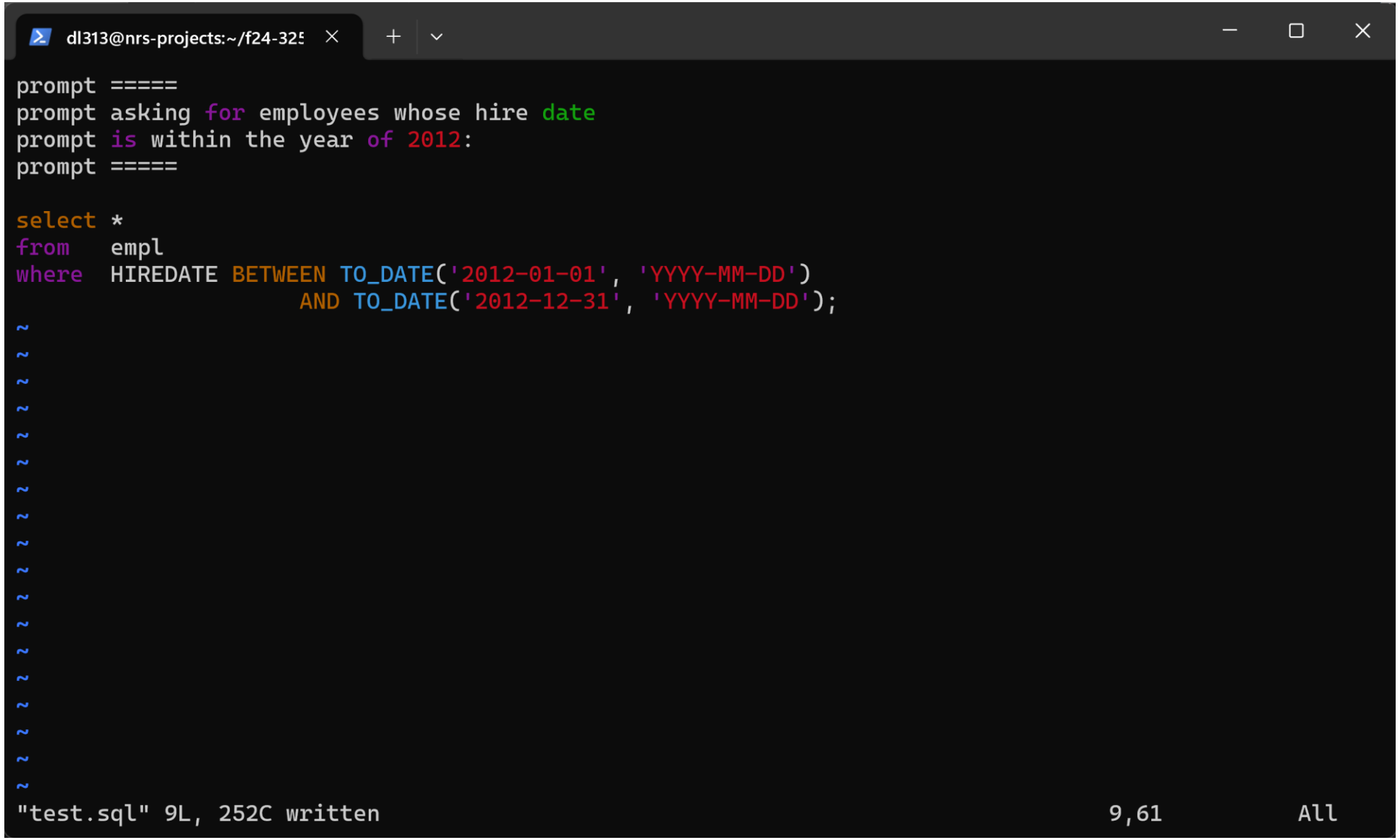
A) Use != or <>.

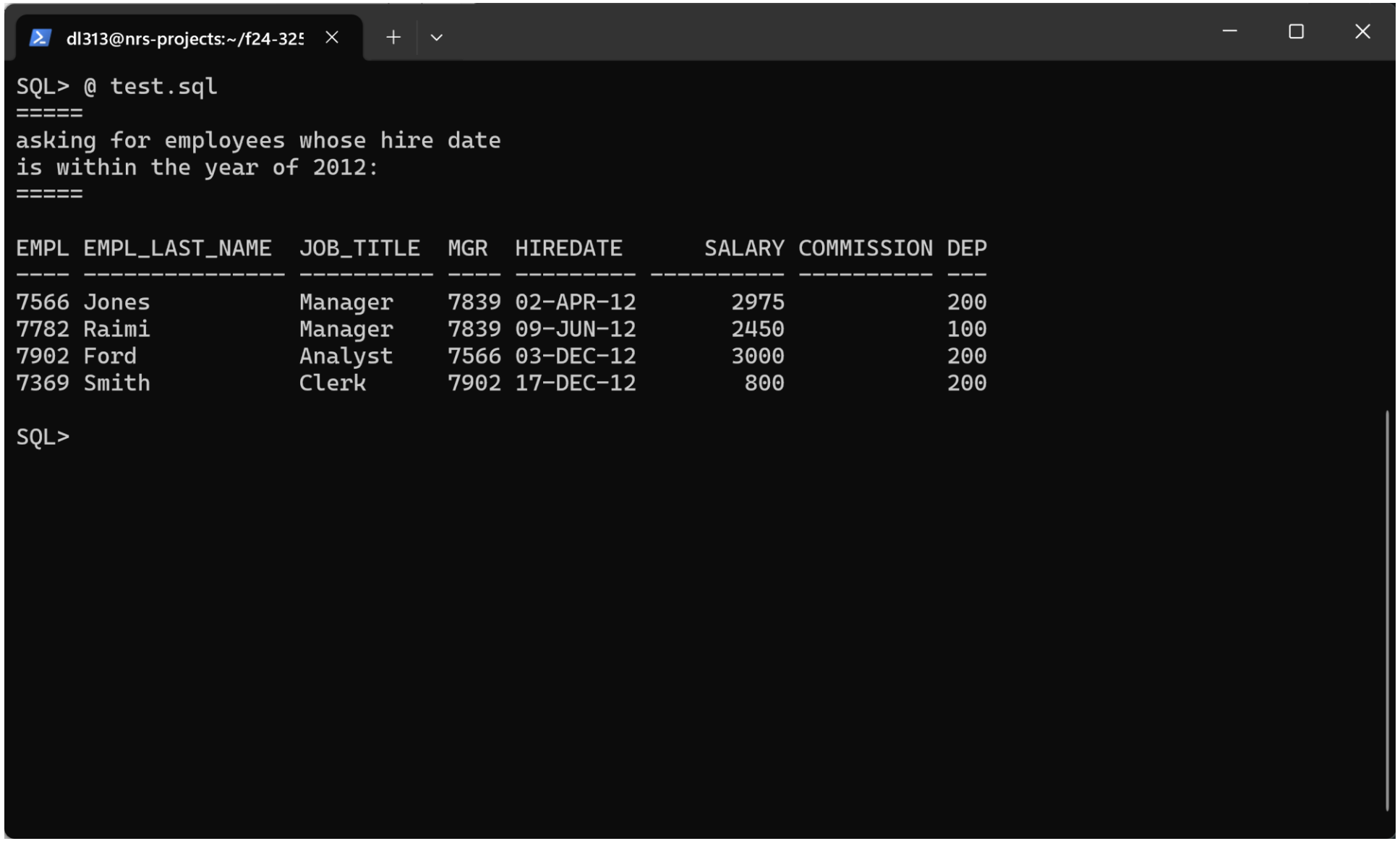
B) Use NOT IN.

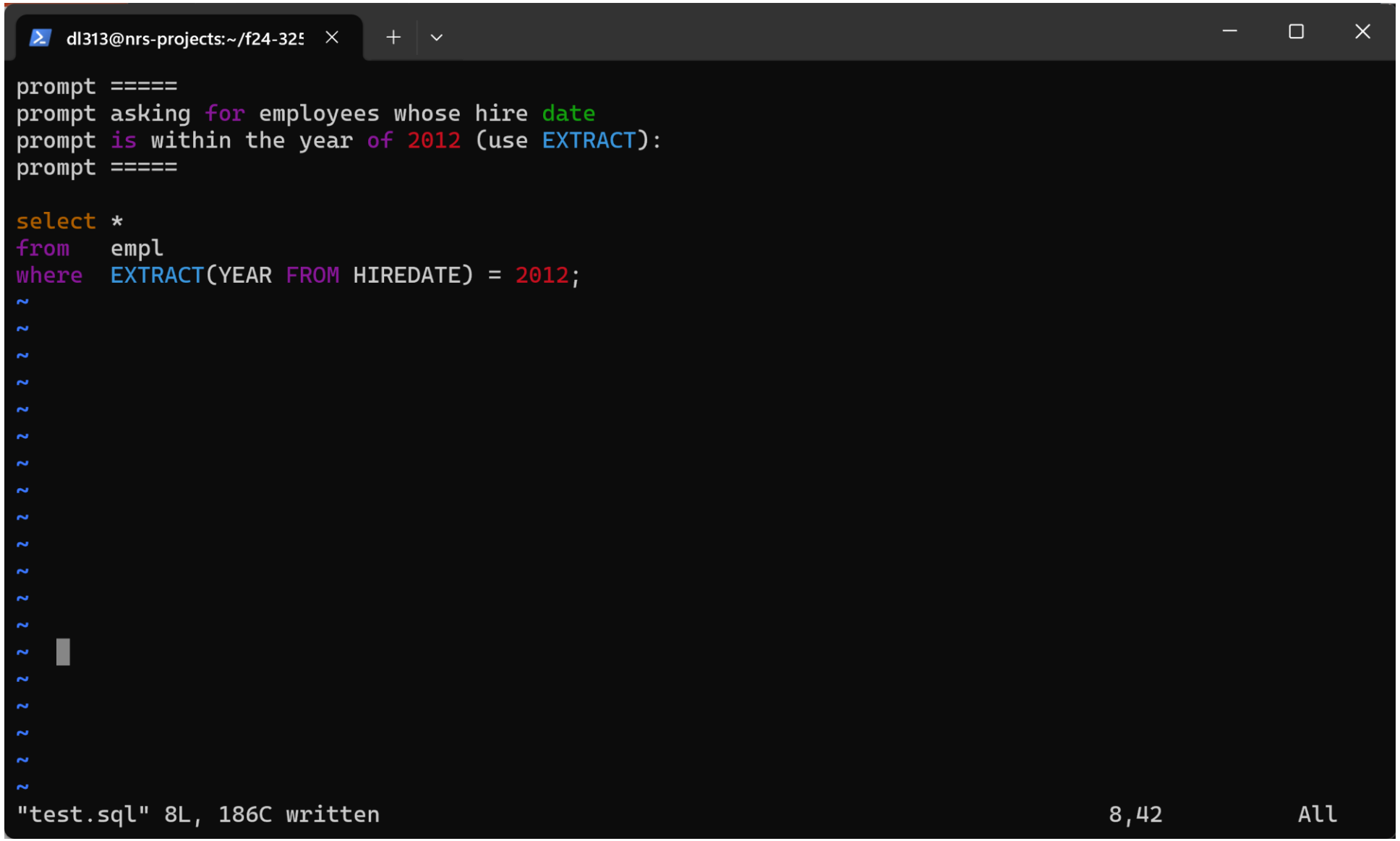
C) Use IS NOT NULL.

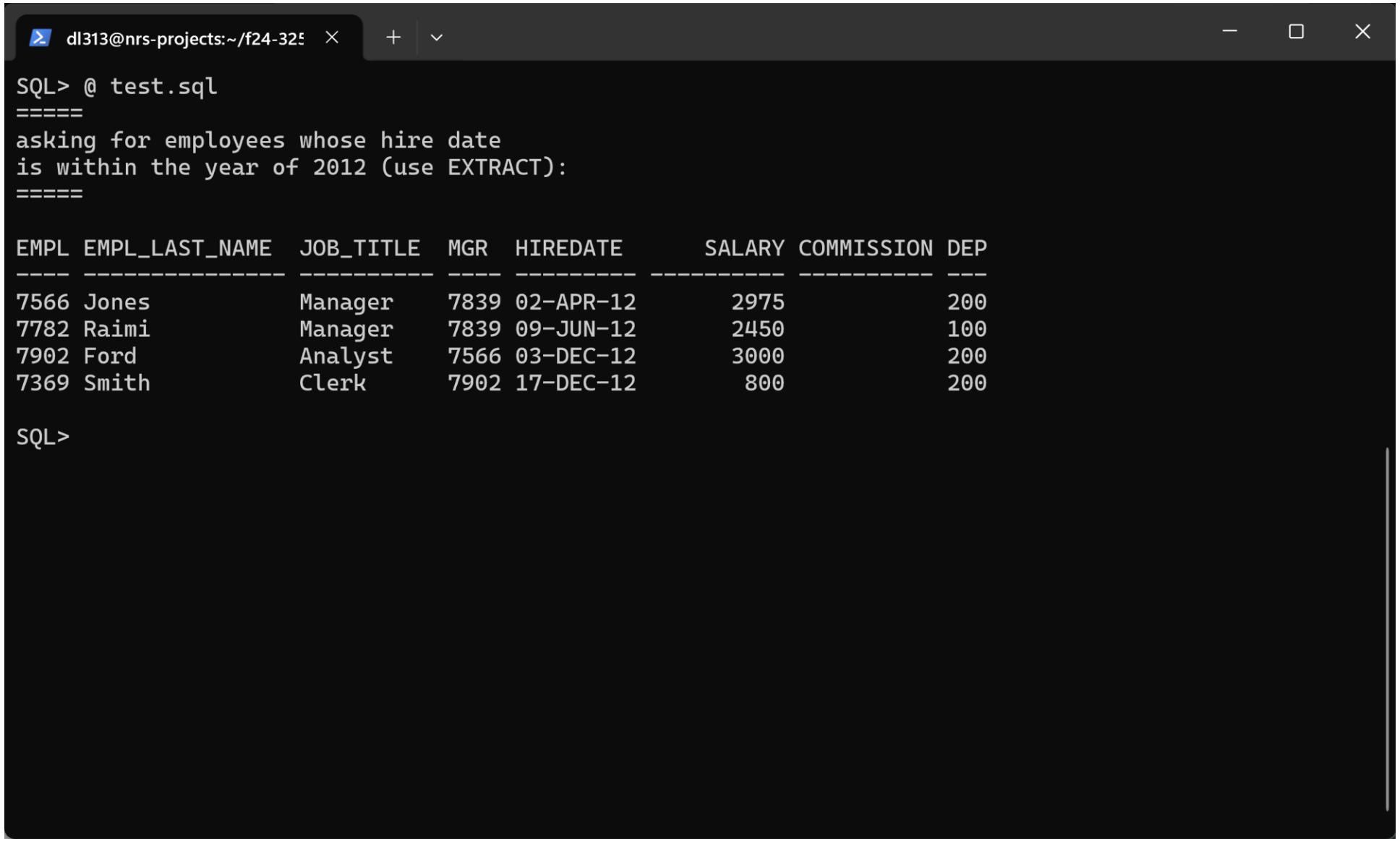
D) Use = followed by the value.

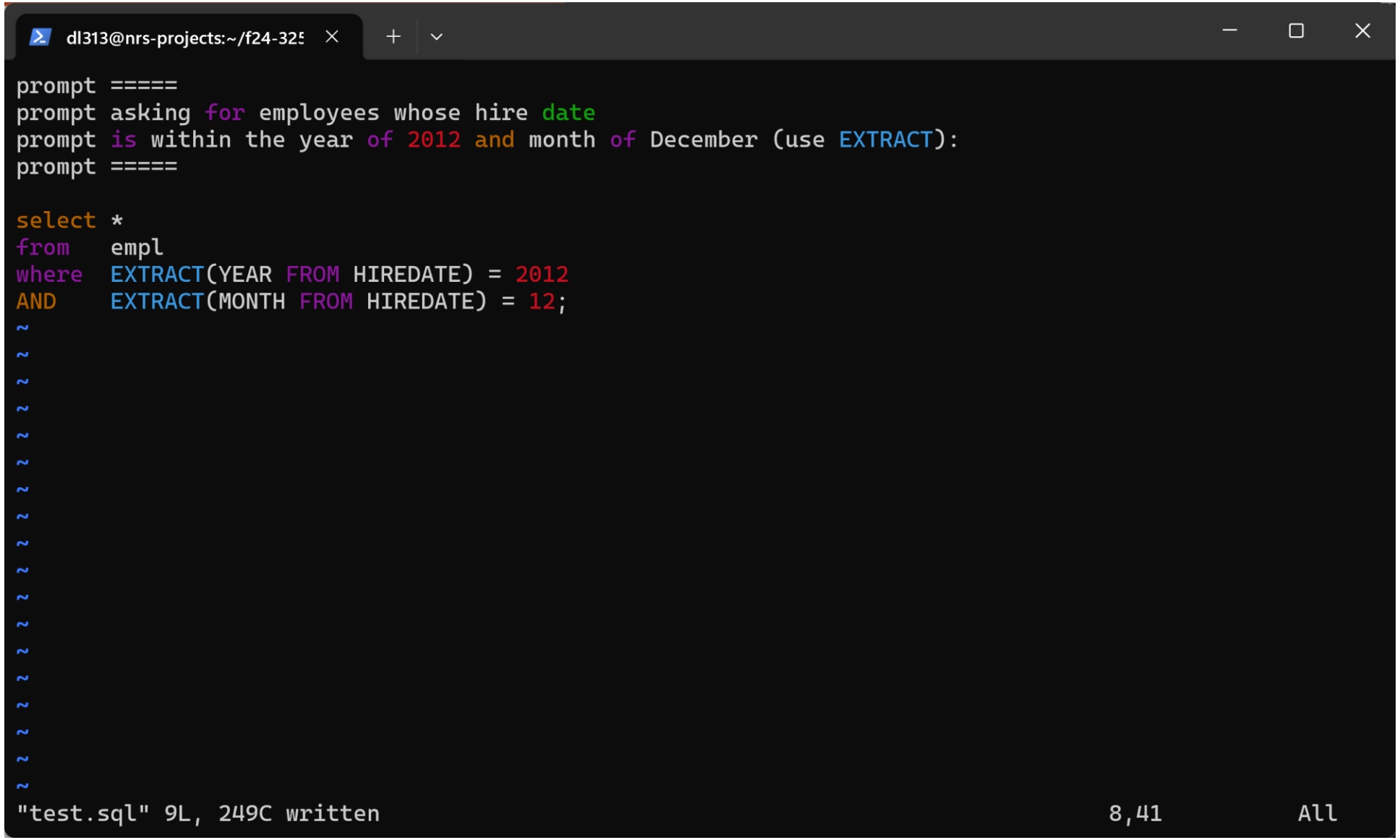

dl313@nrs-projects:~/f24-325
prompt =====
prompt asking for employees whose hire date
prompt is within the year of 2012:
prompt =====
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
"test.sql" 9L, 252C written
9,61
All

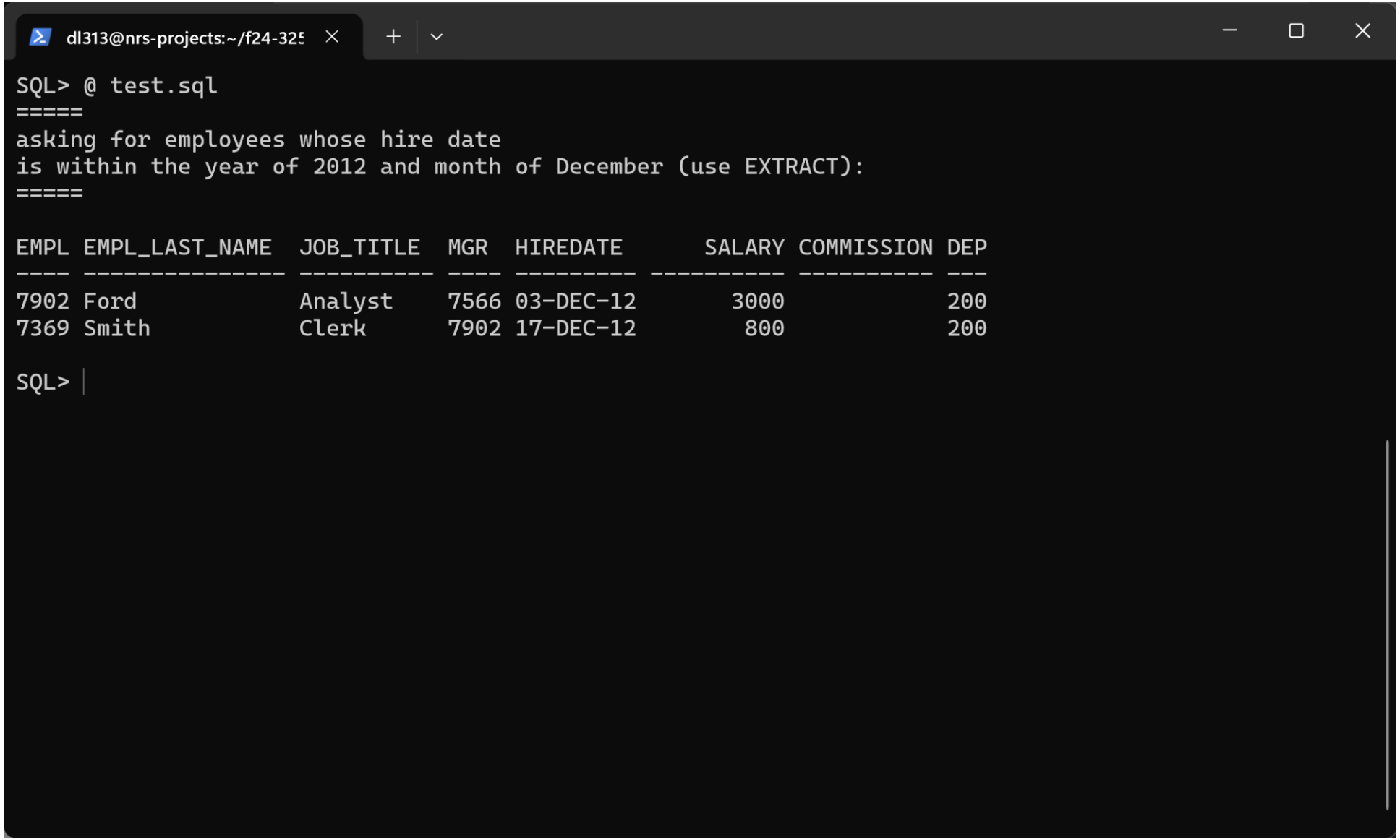

dl313@nrs-projects:~/f24-325
prompt =====
prompt asking for employees whose hire date
prompt is within the year of 2012:
prompt =====
select *
from empl
where HIREDATE BETWEEN TO_DATE('2012-01-01', 'YYYY-MM-DD')
AND TO_DATE('2012-12-31', 'YYYY-MM-DD');
"test.sql" 9L, 252C written
9,61
All


dl313@nrs-projects:~/f24-325
SQL> @ test.sql
=====
asking for employees whose hire date
is within the year of 2012:
=====
EMPL EMPL_LAST_NAME JOB_TITLE MGR HIREDATE SALARY COMMISSION DEP
---- --------------- ---------- ---- --------- ---------- ---------- ---
7566 Jones Manager 7839 02-APR-12 2975 200
7782 Raimi Manager 7839 09-JUN-12 2450 100
7902 Ford Analyst 7566 03-DEC-12 3000 200
7369 Smith Clerk 7902 17-DEC-12 800 200
SQL>


dl313@nrs-projects:~/f24-325
prompt =====
prompt asking for employees whose hire date
prompt is within the year of 2012 (use EXTRACT):
prompt =====
select *
from empl
where EXTRACT(YEAR FROM HIREDATE) = 2012;
"test.sql" 8L, 186C written
8,42
All


dl313@nrs-projects:~/f24-325
SQL> @ test.sql
=====
asking for employees whose hire date
is within the year of 2012 (use EXTRACT):
=====
EMPL EMPL_LAST_NAME JOB_TITLE MGR HIREDATE SALARY COMMISSION DEP
7566 Jones Manager 7839 02-APR-12 2975 200
7782 Raimi Manager 7839 09-JUN-12 2450 100
7902 Ford Analyst 7566 03-DEC-12 3000 200
7369 Smith Clerk 7902 17-DEC-12 800 200
SQL>


dl313@nrs-projects:~/f24-325
prompt =====
prompt asking for employees whose hire date
prompt is within the year of 2012 and month of December (use EXTRACT):
prompt =====
select *
from empl
where EXTRACT(YEAR FROM HIREDATE) = 2012
AND EXTRACT(MONTH FROM HIREDATE) = 12;
"test.sql" 9L, 249C written
8,41
All


dl313@nrs-projects:~/f24-325
SQL> @ test.sql
=====
asking for employees whose hire date
is within the year of 2012 and month of December (use EXTRACT):
=====
EMPL EMPL_LAST_NAME JOB_TITLE MGR HIREDATE SALARY COMMISSION DEP
---- --------------- ---------- ---- --------- ---------- ---------- ---
7902 Ford Analyst 7566 03-DEC-12 3000 200
7369 Smith Clerk 7902 17-DEC-12 800 200
SQL>

dl313 — dl313@nrs-projects:~/f24-325lect05-1 — ssh dl313@nrs-projects-ssh.humboldt.edu — 80×24

```
/*=====
    LIKE operator can select rows where an attribute matches
    a pattern!

    (see the SQL Reading Packet for options for these patterns)
    *   put the pattern in single quotes
    *   the pattern must match the entire attribute value to be true
    *   % matches 0 or more characters
    *   _ matches exactly one character

=====*/
```

-- INSERT -- 170,1 96%

dl313 — dl313@nrs-projects:~/f24-325lect05-1 — ssh dl313@nrs-projects-ssh.humboldt.edu — 80×24

```
    LIKE operator can select rows where an attribute matches
    a pattern!

    (see the SQL Reading Packet for options for these patterns)
    *   put the pattern in single quotes
    *   the pattern must match the entire attribute value to be true
    *   % matches 0 or more characters
    *   _ matches exactly one character

=====*/

prompt =====
prompt asking for employees whose empl_num ENDS with a 9:
prompt =====

select empl_num, empl_last_name
from   empl
where  empl_num like '%9';
```

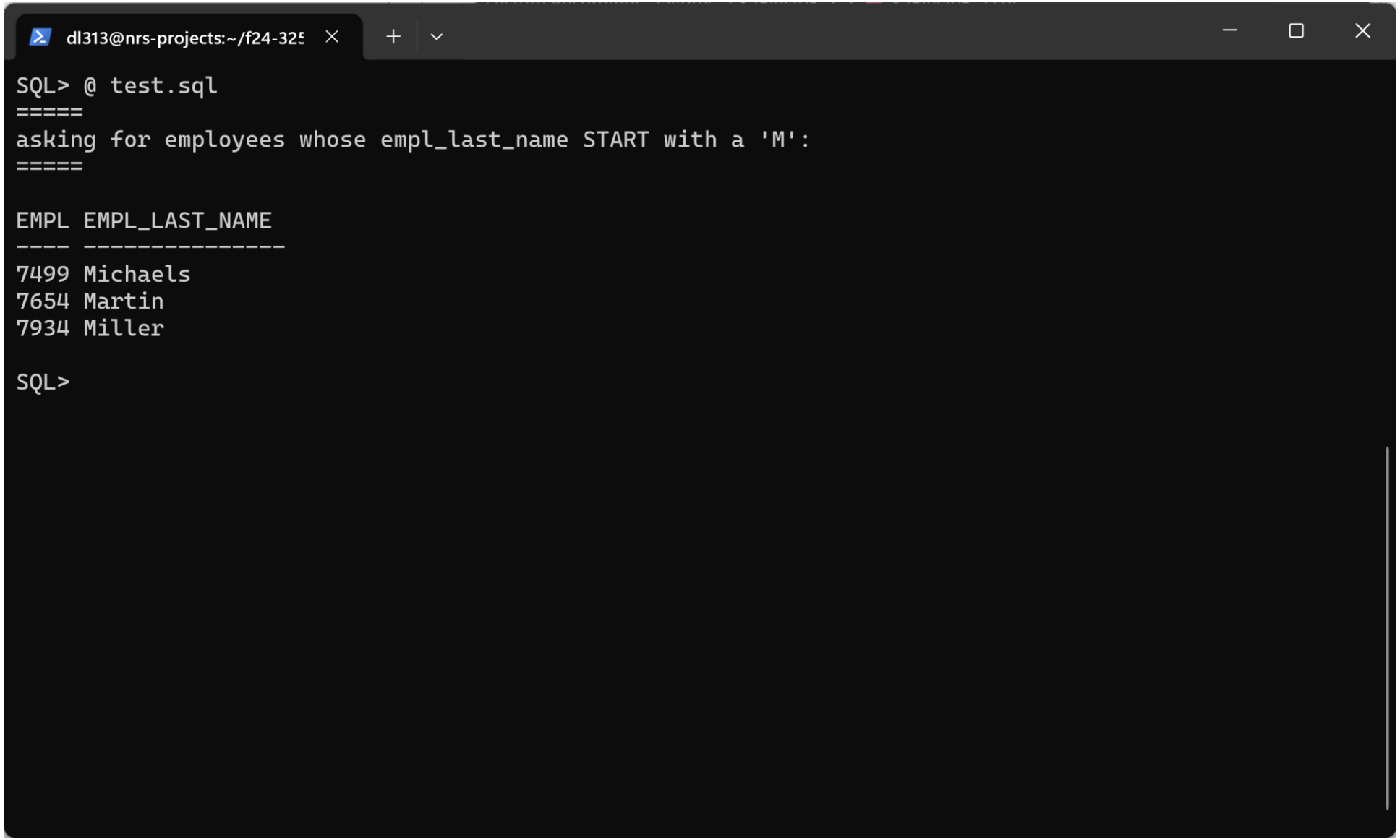
-- INSERT -- 172,27 94%

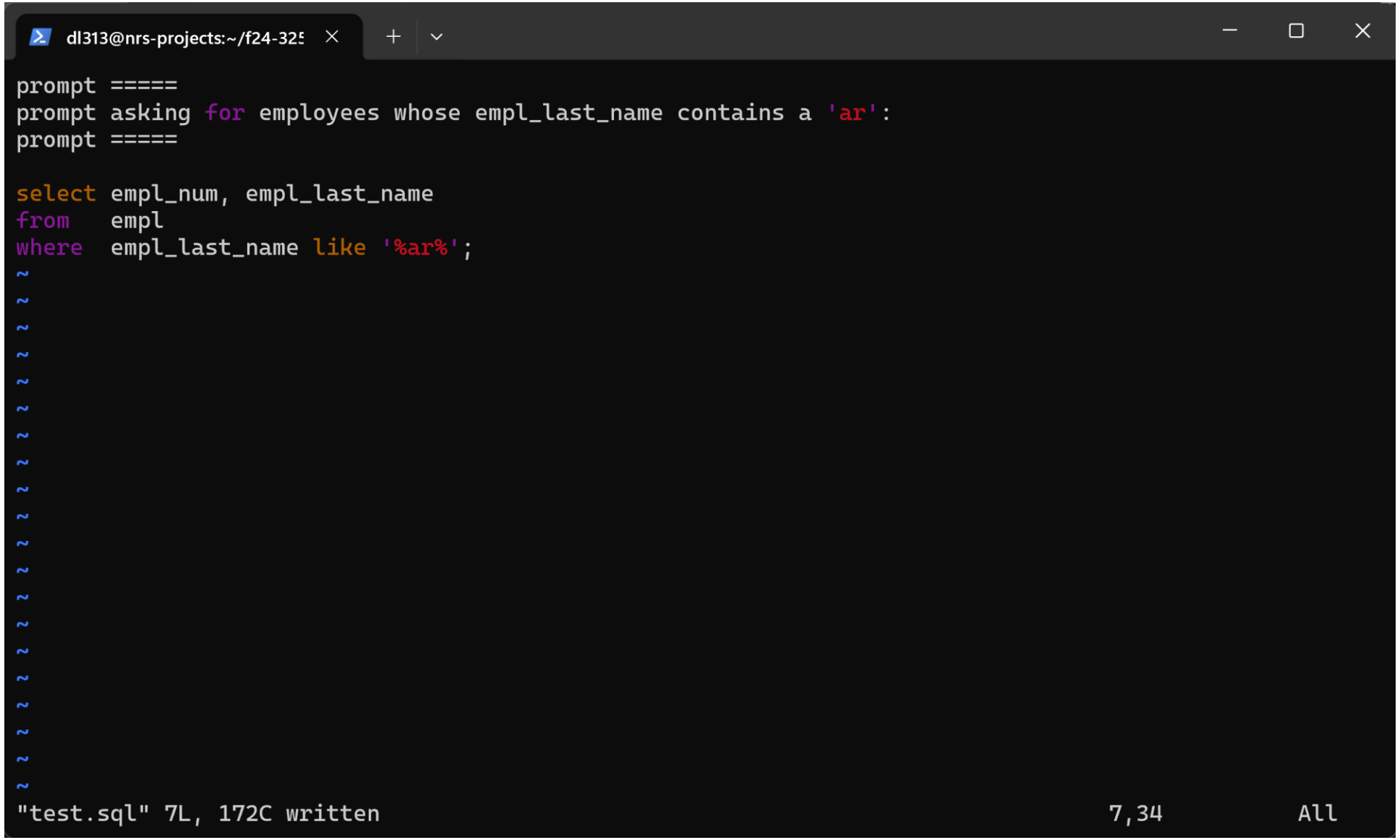
dl313 — dl313@nrs-projects:~/f24-325lect05-1 — ssh dl313@nrs-projects-ssh.humboldt.edu — 80×24

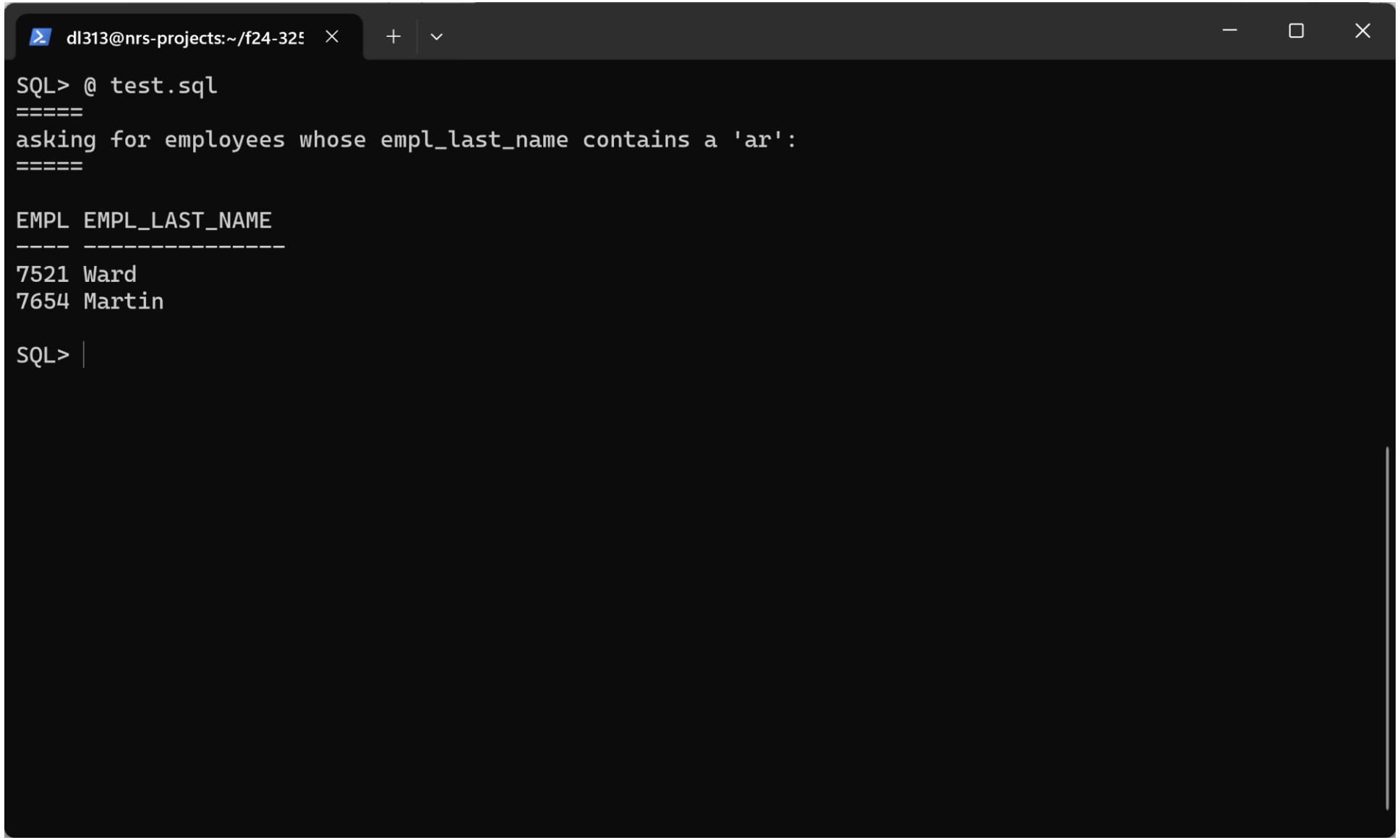
```
=====

EMPL_LAST_NAME       SALARY
--------------- ----------
Michaels               1600
Ward                   1250
Martin                 1250
Turner                 1500
Adams                  1100
Miller                 1300

6 rows selected.

=====
asking for employees whose empl_num ENDS with a 9:
=====

EMPL EMPL_LAST_NAME
---- ---------------
7839 King
7369 Smith
7499 Michaels

SQL>
```

dl313@nrs-projects:~/f24-325

```
prompt =====
prompt asking for employees whose empl_last_name START with a 'M':
prompt =====

select empl_num, empl_last_name
from   empl
where  empl_last_name like 'M%';
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
"test.sql" 7L, 171C written                                   7,32          All
```


dl313@nrs-projects:~/f24-325
SQL> @ test.sql
=====
asking for employees whose empl_last_name START with a 'M':
=====
EMPL EMPL_LAST_NAME
---- ---------------
7499 Michaels
7654 Martin
7934 Miller
SQL>


dl313@nrs-projects:~/f24-325
prompt =====
prompt asking for employees whose empl_last_name contains a 'ar':
prompt =====
select empl_num, empl_last_name
from empl
where empl_last_name like '%ar%';
"test.sql" 7L, 172C written
7,34
All


dl313@nrs-projects:~/f24-325
SQL> @ test.sql
=====
asking for employees whose empl_last_name contains a 'ar':
=====
EMPL EMPL_LAST_NAME
---- ---------------
7521 Ward
7654 Martin
SQL>

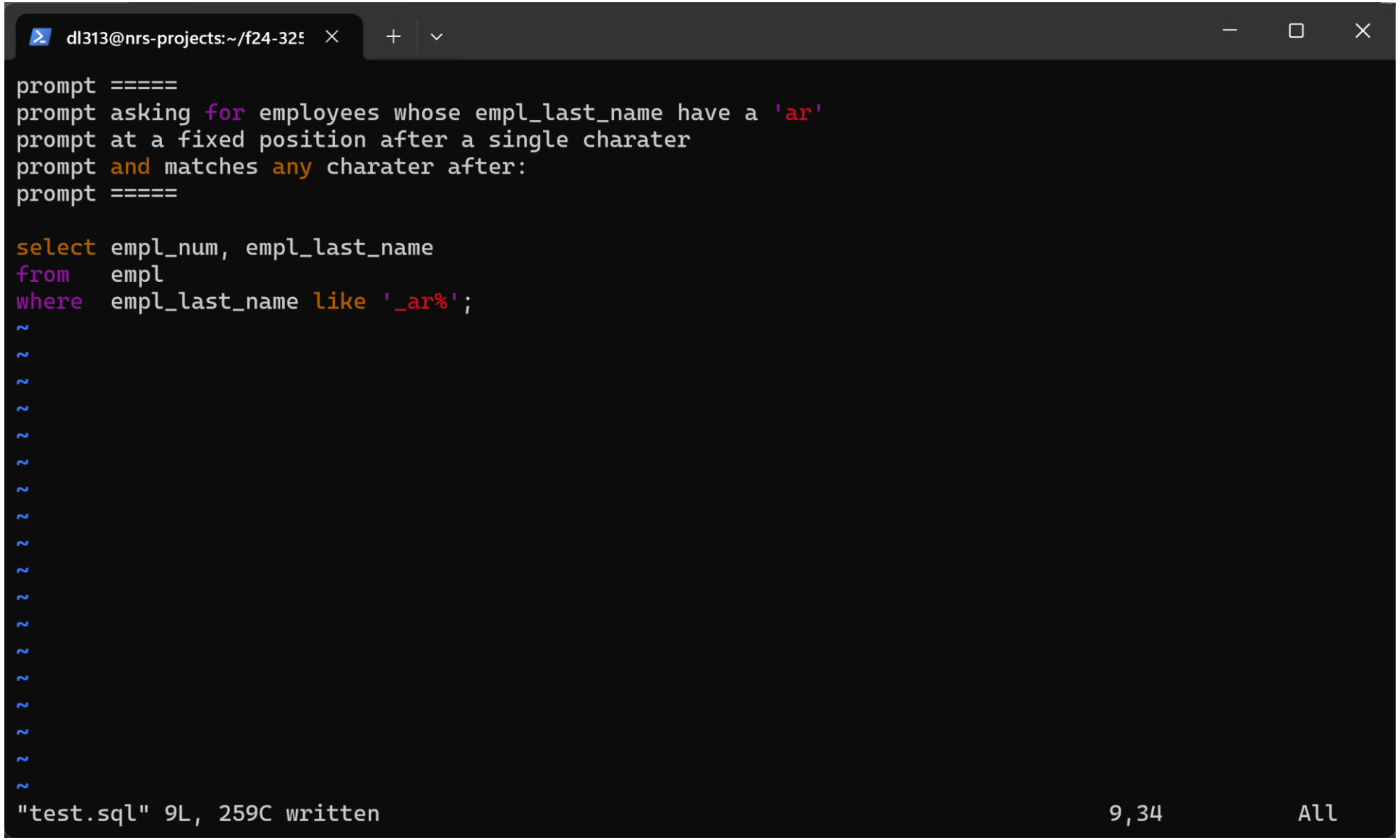
dl313 — dl313@nrs-projects:~/f24-325lect05-1 — ssh dl313@nrs-projects-ssh.humboldt.edu — 80×24

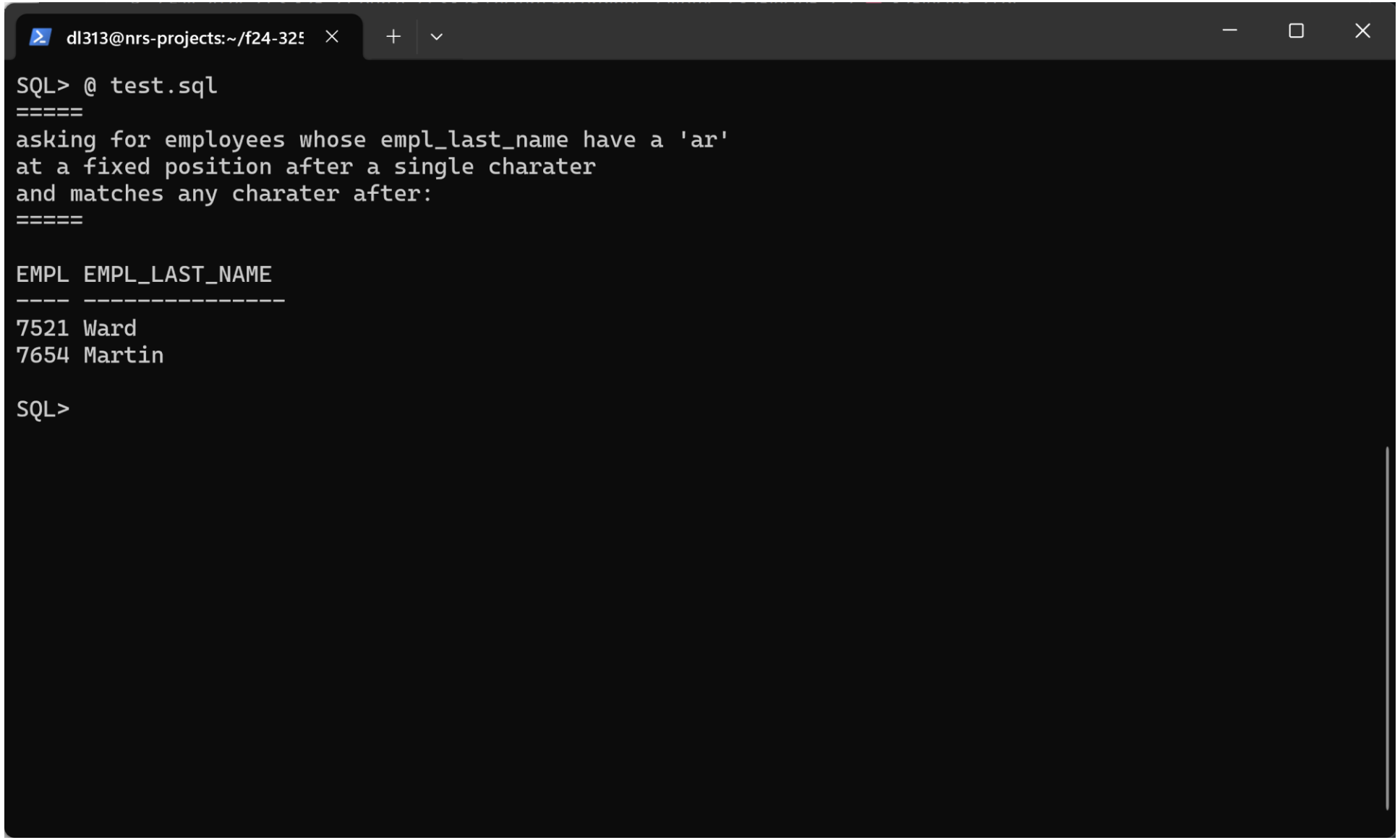
```
prompt =====
prompt asking for employees whose empl_num ENDS with a 9:
prompt =====

select empl_num, empl_last_name
from   empl
where  empl_num like '%9';

prompt =====
prompt asking for employees whose job_title is ONE letter followed by
prompt anager
prompt =====

select job_title, empl_last_name
from   empl
where  job_title like '_anager';
```

-- INSERT -- 182,1 94%

dl313 — dl313@nrs-projects:~/f24-325lect05-1 — ssh dl313@nrs-projects-ssh.humboldt.edu — 80×24

```
6 rows selected.

=====
asking for employees whose empl_num ENDS with a 9:
=====

EMPL EMPL_LAST_NAME
---- ---------------
7839 King
7369 Smith
7499 Michaels

=====
asking for employees whose job_title is ONE letter followed by
anager
=====

JOB_TITLE  EMPL_LAST_NAME
---------- ---------------
Manager    Jones
Manager    Blake
Manager    Raimi

SQL>
```


dl313@nrs-projects:~/f24-325
prompt =====
prompt asking for employees whose empl_last_name have a 'ar'
prompt at a fixed position after a single charater
prompt and matches any charater after:
prompt =====
select empl_num, empl_last_name
from empl
where empl_last_name like '_ar%';
"test.sql" 9L, 259C written
9,34
All


dl313@nrs-projects:~/f24-325
SQL> @ test.sql
=====
asking for employees whose empl_last_name have a 'ar'
at a fixed position after a single charater
and matches any charater after:
=====
EMPL EMPL_LAST_NAME
---- ---------------
7521 Ward
7654 Martin
SQL>

Question

Consider:

```
select  empl_last_name, dept_num, dept_name, dept_loc
from    empl, dept
where   empl.dept_num = dept.dept_num
        and     hiredate > '01-dec-1991';
```

Why won't this query work?

1. Because you need dept.dept_loc in the SELECT clause.
2. Because you need dept.dept_name in the SELECT clause.
3. Because you need dept.dept_num in the SELECT clause.
4. Because you need empl.empl_last_name in the SELECT clause.