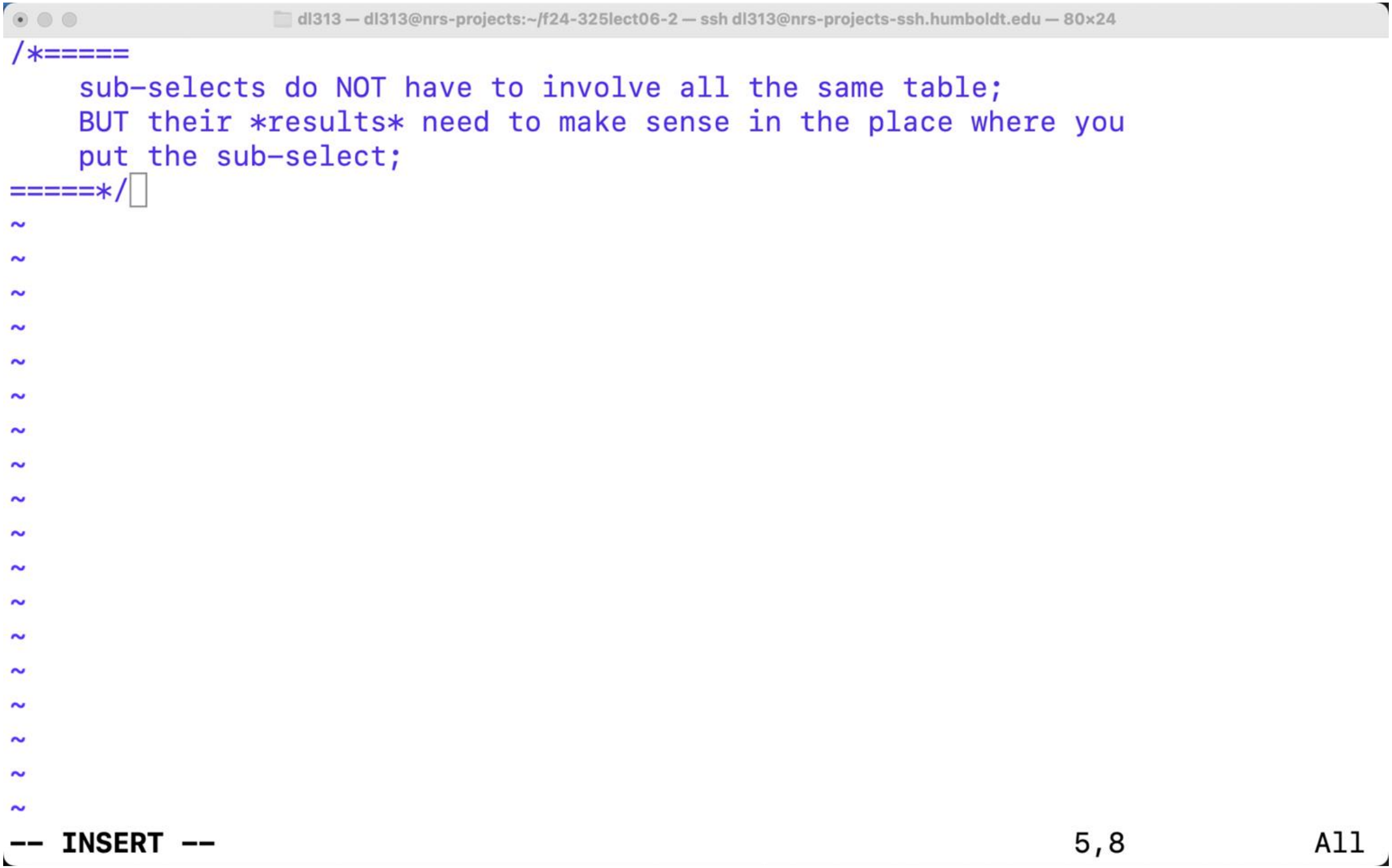
CS 325 Week 6-2

SQL sub-selects (SQL Reading Packet 4)

Review for Exam 1


dl313 — dl313@nrs-projects:~/f24-325lect06-2 — ssh dl313@nrs-projects-ssh.humboldt.edu — 80×24
[dl313@nrs-projects ~]$ mkdir f24-325lect06-2
[dl313@nrs-projects ~]$ chmod 700 f24-325lect06-2
[dl313@nrs-projects ~]$ cd f24-325lect06-2
[dl313@nrs-projects f24-325lect06-2]$ vim 325lect06-2.sql


dl313 — dl313@nrs-projects:~/f24-325lect06-2 — ssh dl313@nrs-projects-ssh.humboldt.edu — 80×24
/*=====
sub-selects do NOT have to involve all the same table;
BUT their *results* need to make sense in the place where you
put the sub-select;
=====*/
-- INSERT --
5,8
All

dl313 — dl313@nrs-projects:~/f24-325lect06-2 — ssh dl313@nrs-projects-ssh.humboldt.edu — 80×24

```
/*=====
    sub-selects do NOT have to involve all the same table;
    BUT their *results* need to make sense in the place where you
    put the sub-select;
=====*/

prompt =====
prompt I want to know which employees work in Dallas.
prompt =====

prompt =====
prompt I *could* start by asking: what departments are in Dallas?
prompt =====

select dept_num
from   dept
where  dept_loc = 'Dallas';
~
~
~
~
~
~
-- INSERT --                                          17,28         All
```

dl313 — dl313@nrs-projects:~/f24-325lect06-2 — ssh dl313@nrs-projects-ssh.humboldt.edu — 80×24

```
SQL> @ 325lect06-2.sql
```

dl313 — dl313@nrs-projects:~/f24-325lect06-2 — ssh dl313@nrs-projects-ssh.humboldt.edu — 80×24

```
SQL> @ 325lect06-2.sql
=====
I want to know which employees work in Dallas.
=====
=====
I *could* start by asking: what departments are in Dallas?
=====

DEP
---
200

SQL>
```

dl313 — dl313@nrs-projects:~/f24-325lect06-2 — ssh dl313@nrs-projects-ssh.humboldt.edu — 80×24

```
/*=====
    sub-selects do NOT have to involve all the same table;
    BUT their *results* need to make sense in the place where you
    put the sub-select;
=====*/

prompt =====
prompt I want to know which employees work in Dallas.
prompt =====

prompt =====
prompt I *could* start by asking: what departments are in Dallas?
prompt =====

select dept_num
from   dept
where  dept_loc = 'Dallas';

prompt =====
prompt then I could use this select as a sub-select in another
prompt query, to find out the last names of employees who work
prompt in Dallas:
prompt =====
-- INSERT --                                          23,13         All
```

dl313 — dl313@nrs-projects:~/f24-325lect06-2 — ssh dl313@nrs-projects-ssh.humboldt.edu — 80×24

```
select dept_num
from   dept
where  dept_loc = 'Dallas';

prompt =====
prompt then I could use this select as a sub-select in another
prompt query, to find out the last names of employees who work
prompt in Dallas:
prompt =====

select empl_last_name
from   empl
where  dept_num IN (select dept_num
                    from   dept
                    where  dept_loc = 'Dallas');
```

-- INSERT -- 31,1 92%

dl313 — dl313@nrs-projects:~/f24-325lect06-2 — ssh dl313@nrs-projects-ssh.humboldt.edu — 80×24

```
select dept_num
from   dept
where  dept_loc = 'Dallas';

prompt =====
prompt then I could use this select as a sub-select in another
prompt query, to find out the last names of employees who work
prompt in Dallas:
prompt =====

select empl_last_name
from   empl
where  dept_num IN (select dept_num
                    from   dept
                    where  dept_loc = 'Dallas');
```

-- INSERT -- 31,1 92%

dl313 — dl313@nrs-projects:~/f24-325lect06-2 — ssh dl313@nrs-projects-ssh.humboldt.edu — 80×24

```
SQL> @ 325lect06-2.sql
```

dl313 — dl313@nrs-projects:~/f24-325lect06-2 — ssh dl313@nrs-projects-ssh.humboldt.edu — 80×24

```
I want to know which employees work in Dallas.
=====
=====
I *could* start by asking: what departments are in Dallas?
=====

DEP
---
200

=====
then I could use this select as a sub-select in another
query, to find out the last names of employees who work
in Dallas:
=====

EMPL_LAST_NAME
---------------
Scott
Jones
Ford
Smith

SQL>
```

dl313 — dl313@nrs-projects:~/f24-325lect06-2 — ssh dl313@nrs-projects-ssh.humboldt.edu — 80×24

```
/*=====
    examples of some common errors...
=====*/

prompt =====
prompt you have to be careful where you use aggregate function calls;
prompt THIS causes an error:
prompt =====
```

-- INSERT -- 47,1 93%

dl313 — dl313@nrs-projects:~/f24-325lect06-2 — ssh dl313@nrs-projects-ssh.humboldt.edu — 80×24

```
/*=====
    examples of some common errors...
=====*/

prompt =====
prompt you have to be careful where you use aggregate function calls;
prompt THIS causes an error:
prompt =====

select empl_last_name
from   empl
where  job_title = 'Clerk'
       and salary = max(salary);
```

-- INSERT -- 43,33 85%

dl313 — dl313@nrs-projects:~/f24-325lect06-2 — ssh dl313@nrs-projects-ssh.humboldt.edu — 80×24

```
SQL> @ 325lect06-2.sql
```

dl313 — dl313@nrs-projects:~/f24-325lect06-2 — ssh dl313@nrs-projects-ssh.humboldt.edu — 80×24

```
=====
then I could use this select as a sub-select in another
query, to find out the last names of employees who work
in Dallas:
=====

EMPL_LAST_NAME
---------------
Scott
Jones
Ford
Smith

=====
you have to be careful where you use aggregate function calls
THIS causes an error:
=====
       and salary = max(salary)
                    *
ERROR at line 4:
ORA-00934: group function is not allowed here


SQL>
```

dl313 — dl313@nrs-projects:~/f24-325lect06-2 — ssh dl313@nrs-projects-ssh.humboldt.edu — 80×24

```
prompt =====

select empl_last_name
from   empl
where  job_title = 'Clerk'
       and salary = max(salary);

prompt =====
prompt what if you want to know the names of Clerks making
prompt more than the salary of the average-paid Clerk?
prompt =====

select empl_last_name, job_title, salary
from   empl
where  job_title = 'Clerk'
       and salary > (select avg(salary)
                     from   empl
                     where  job_title = 'Clerk');
```

-- INSERT -- 55,50 80%

dl313 — dl313@nrs-projects:~/f24-325lect06-2 — ssh dl313@nrs-projects-ssh.humboldt.edu — 80×24

```
SQL> @ 325lect06-2.sql
```

```
Ford
Smith

=====
you have to be careful where you use aggregate function calls
THIS causes an error:
=====
       and salary = max(salary)
                    *
ERROR at line 4:
ORA-00934: group function is not allowed here



=====
what if you want to know the names of Clerks making
more than the salary of the average-paid Clerk?
=====

EMPL_LAST_NAME  JOB_TITLE      SALARY
--------------- ---------- ----------
Adams           Clerk            1100
Miller          Clerk            1300

SQL>
```

dl313 — dl313@nrs-projects:~/f24-325lect06-2 — ssh dl313@nrs-projects-ssh.humboldt.edu — 80×24

```
prompt =====

select empl_last_name, job_title, salary
from   empl
where  job_title = 'Clerk'
       and salary > (select avg(salary)
                     from   empl
                     where  job_title = 'Clerk');

prompt =====
prompt be careful; you need both instances of job_title = 'Clerk' here!
prompt =====

select empl_last_name, job_title, salary
from   empl
where  salary > (select avg(salary)
                 from   empl
                 where  job_title = 'Clerk');
```

-- INSERT -- 65,46 83%

dl313 — dl313@nrs-projects:~/f24-325lect06-2 — ssh dl313@nrs-projects-ssh.humboldt.edu — 80×24

```
SQL> @ 325lect06-2.sql
```

dl313 — dl313@nrs-projects:~/f24-325lect06-2 — ssh dl313@nrs-projects-ssh.humboldt.edu — 80×24

```
be careful; you need both instances of job_title = 'Clerk' here!
=====

EMPL_LAST_NAME  JOB_TITLE      SALARY
--------------- ---------- ----------
King            President        5000
Jones           Manager          2975
Blake           Manager          2850
Raimi           Manager          2450
Ford            Analyst          3000
Michaels        Sales            1600
Ward            Sales            1250
Martin          Sales            1250
Scott           Analyst          3000
Turner          Sales            1500
Adams           Clerk            1100

EMPL_LAST_NAME  JOB_TITLE      SALARY
--------------- ---------- ----------
Miller          Clerk            1300

12 rows selected.

SQL>
```

```
prompt be careful; you need both instances of job_title = 'Clerk' here!
prompt =====

select empl_last_name, job_title, salary
from   empl
where  salary > (select avg(salary)
                 from   empl
                 where  job_title = 'Clerk');

prompt =====
prompt be careful; you need both instances of job_title = 'Clerk' here!
prompt =====

select empl_last_name, job_title, salary
from   empl
where  job_title = 'Clerk'
       and salary > (select avg(salary)
                     from   empl);
```

dl313 — dl313@nrs-projects:~/f24-325lect06-2 — ssh dl313@nrs-projects-ssh.humboldt.edu — 80×24

```
SQL> @ 325lect06-2.sql
```

```
Jones             Manager              2975
Blake             Manager              2850
Raimi             Manager              2450
Ford              Analyst              3000
Michaels          Sales                1600
Ward              Sales                1250
Martin            Sales                1250
Scott             Analyst              3000
Turner            Sales                1500
Adams             Clerk                1100

EMPL_LAST_NAME  JOB_TITLE      SALARY
--------------- ---------- ----------
Miller          Clerk              1300

12 rows selected.

=====
be careful; you need both instances of job_title = 'Clerk' here!
=====

no rows selected

SQL>
```

dl313 — dl313@nrs-projects:~/f24-325lect06-2 — ssh dl313@nrs-projects-ssh.humboldt.edu — 80×24

```
/*=====
    BE CAREFUL:
    = < > <= >= != <>
    ...can only have ONE value for EACH of their operands!

    (why it sometimes matters whether you use = or IN for a sub-select!
    IN can have a set of values for its right-hand-side operand,
    BUT = can only have ONE value for its right-hand-side operand)
=====*/
```

-- INSERT -- 94,1 Bot

dl313 — dl313@nrs-projects:~/f24-325lect06-2 — ssh dl313@nrs-projects-ssh.humboldt.edu — 80×24

```
/*=====
    BE CAREFUL:
    = < > <= >= != <>
    ...can only have ONE value for EACH of their operands!

    (why it sometimes matters whether you use = or IN for a sub-select!
    IN can have a set of values for its right-hand-side operand,
    BUT = can only have ONE value for its right-hand-side operand)
=====*/

prompt =====
prompt error you will see if you use a subquery that returns more than
prompt one row with the = operator:
prompt =====

select min(salary)
from   empl
where  job_title = 'Sales';
```

"325lect06-2.sql" 106L, 2384C written 94,27 91%

dl313 — dl313@nrs-projects:~/f24-325lect06-2 — ssh dl313@nrs-projects-ssh.humboldt.edu — 80×24

```
SQL> @ 325lect06-2.sql
```

dl313 — dl313@nrs-projects:~/f24-325lect06-2 — ssh dl313@nrs-projects-ssh.humboldt.edu — 80×24

```
Adams                Clerk                 1100

EMPL_LAST_NAME  JOB_TITLE           SALARY
--------------- ---------- ----------
Miller               Clerk                 1300

12 rows selected.

=====
be careful; you need both instances of job_title = 'Clerk' here!
=====

no rows selected

=====
error you will see if you use a subquery that returns more than
one row with the = operator:
=====

MIN(SALARY)
-----------
       1250

SQL>
```

dl313 — dl313@nrs-projects:~/f24-325lect06-2 — ssh dl313@nrs-projects-ssh.humboldt.edu — 80×24

```
prompt =====
prompt error you will see if you use a subquery that returns more than
prompt one row with the = operator:
prompt =====

select min(salary)
from   empl
where  job_title = 'Sales';

select mgr
from   empl
where  job_title = 'Sales'
       and salary = (select min(salary)
                     from   empl
                     where  job_title = 'Sales');
```

"325lect06-2.sql" 113L, 2558C written 101,49 94%

dl313 — dl313@nrs-projects:~/f24-325lect06-2 — ssh dl313@nrs-projects-ssh.humboldt.edu — 80×24

```
SQL> @ 325lect06-2.sql
```

dl313 — dl313@nrs-projects:~/f24-325lect06-2 — ssh dl313@nrs-projects-ssh.humboldt.edu — 80×24

```
12 rows selected.

=====
be careful; you need both instances of job_title = 'Clerk' here!
=====

no rows selected

=====
error you will see if you use a subquery that returns more than
one row with the = operator:
=====

MIN(SALARY)
-----------
       1250


MGR
----
7698
7698

SQL>
```

dl313 — dl313@nrs-projects:~/f24-325lect06-2 — ssh dl313@nrs-projects-ssh.humboldt.edu — 80×24

```
prompt =====

select min(salary)
from   empl
where  job_title = 'Sales';

select mgr
from   empl
where  job_title = 'Sales'
       and salary = (select min(salary)
                     from   empl
                     where  job_title = 'Sales');

select empl_last_name
from   empl
where  empl_num = (select mgr
                   from   empl
                   where  job_title = 'Sales'
                          and salary = (select min(salary)
                                        from   empl
                                        where  job_title = 'Sales'));


-- INSERT --                                                  102,1         89%
```

dl313 — dl313@nrs-projects:~/f24-325lect06-2 — ssh dl313@nrs-projects-ssh.humboldt.edu — 80×24

```
SQL> @ 325lect06-2.sql
```

```
no rows selected

=====
error you will see if you use a subquery that returns more than
one row with the = operator:
=====

MIN(SALARY)
-----------
       1250


MGR
----
7698
7698

where  empl_num = (select mgr
                   *
ERROR at line 3:
ORA-01427: single-row subquery returns more than one row


SQL>
```

dl313 — dl313@nrs-projects:~/f24-325lect06-2 — 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

14 rows selected.
```

dl313 — dl313@nrs-projects:~/f24-325lect06-2 — ssh dl313@nrs-projects-ssh.humboldt.edu — 80×24

```
prompt =====
prompt but if I use IN instead of = here, this query works:
prompt =====

select empl_last_name
from   empl
where  empl_num IN (select mgr
                    from   empl
                    where  job_title = 'Sales'
                           and salary = (select min(salary)
                                         from   empl
                                         where  job_title = 'Sales'));
```

"325lect06-2.sql" 133L, 3295C written 129,0-1 Bot

dl313 — dl313@nrs-projects:~/f24-325lect06-2 — ssh dl313@nrs-projects-ssh.humboldt.edu — 80×24

```
SQL> @ 325lect06-2.sql
```

```
-----------
       1250


MGR
----
7698
7698

where  empl_num = (select mgr
                   *
ERROR at line 3:
ORA-01427: single-row subquery returns more than one row


=====
but if I use IN instead of = here, this query works:
=====

EMPL_LAST_NAME
---------------
Blake

SQL>
```

dl313 — dl313@nrs-projects:~/f24-325lect06-2 — ssh dl313@nrs-projects-ssh.humboldt.edu — 80×24

```
/*======
   another place you can put a sub-select:
   ...in a FROM clause!

   you can put names of tables, or EXPRESSIONS whose values are tabular,
   in a FROM clause! SO, a select results in tabular data,
   so a sub-select CAN be in a FROM clause
=====*/
```

-- INSERT -- 146,1 Bot

dl313 — dl313@nrs-projects:~/f24-325lect06-2 — ssh dl313@nrs-projects-ssh.humboldt.edu — 80×24

```
/*======
   another place you can put a sub-select:
   ...in a FROM clause!

   you can put names of tables, or EXPRESSIONS whose values are tabular,
   in a FROM clause! SO, a select results in tabular data,
   so a sub-select CAN be in a FROM clause
=====*/

prompt =====
prompt rather silly example of a sub-select in a FROM clause:
prompt =====

select empl_last_name, dept_name
from   (select *
        from   empl e, dept d
        where  e.dept_num = d.dept_num)
where  dept_name = 'Operations';
```

"325lect06-2.sql" 154L, 3801C written 142,32 94%

dl313 — dl313@nrs-projects:~/f24-325lect06-2 — ssh dl313@nrs-projects-ssh.humboldt.edu — 80×24

```
SQL> @ 325lect06-2.sql
```

```
where  empl_num = (select mgr
                   *
ERROR at line 3:
ORA-01427: single-row subquery returns more than one row


=====
but if I use IN instead of = here, this query works:
=====

EMPL_LAST_NAME
---------------
Blake

=====
rather silly example of a sub-select in a FROM clause:
=====

EMPL_LAST_NAME  DEPT_NAME
--------------- --------------
Adams           Operations

SQL>
```

```
   you can put names of tables, or EXPRESSIONS whose values are tabular,
   in a FROM clause! SO, a select results in tabular data,
   so a sub-select CAN be in a FROM clause
=====*/

prompt =====
prompt rather silly example of a sub-select in a FROM clause:
prompt =====

select empl_last_name, dept_name
from   (select *
        from   empl e, dept d
        where  e.dept_num = d.dept_num)
where  dept_name = 'Operations';

select *
from   empl e, dept d
where  e.dept_num = d.dept_num;
```

dl313 — dl313@nrs-projects:~/f24-325lect06-2 — ssh dl313@nrs-projects-ssh.humboldt.edu — 80×24

```
SQL> @ 325lect06-2.sql
```

dl313 — dl313@nrs-projects:~/f24-325lect06-2 — ssh dl313@nrs-projects-ssh.humboldt.edu — 80×24

```
7499 Michaels        Sales       7698 20-FEB-18        1600          300 300 300
Sales           Chicago

7900 James           Clerk       7698 03-DEC-17         950              300 300
Sales           Chicago

7844 Turner          Sales       7698 08-SEP-19        1500            0 300 300
Sales           Chicago


EMPL EMPL_LAST_NAME  JOB_TITLE   MGR  HIREDATE       SALARY COMMISSION DEP DEP
---- --------------- ---------- ---- --------- ---------- ---------- --- ---
DEPT_NAME       DEPT_LOC
--------------- --------------
7876 Adams           Clerk       7788 23-SEP-18        1100              400 400
Operations      Boston

7839 King            President        17-NOV-11        5000              500 500
Management      New York


14 rows selected.

SQL>
```

dl313 — dl313@nrs-projects:~/f24-325lect06-2 — ssh dl313@nrs-projects-ssh.humboldt.edu — 80×24

```
/*=====
   note, though: the outer select only SEES the columns as they
   are projected from the subquery in the FROM clause;
=====*/

prompt =====
prompt another rather silly example of a sub-select in a FROM clause:
prompt =====
```

-- INSERT -- 165,1 Bot

dl313 — dl313@nrs-projects:~/f24-325lect06-2 — ssh dl313@nrs-projects-ssh.humboldt.edu — 80×24

```
/*=====
   note, though: the outer select only SEES the columns as they
   are projected from the subquery in the FROM clause;
=====*/

prompt =====
prompt another rather silly example of a sub-select in a FROM clause:
prompt =====

select ename
from   (select empl_last_name ename, dept_name dname
        from   empl e, dept d
        where  e.dept_num = d.dept_num)
where   dname = 'Operations';
```

"325lect06-2.sql" 169L, 4201C written 157,29 97%

dl313 — dl313@nrs-projects:~/f24-325lect06-2 — ssh dl313@nrs-projects-ssh.humboldt.edu — 80×24

```
SQL> @ 325lect06-2.sql
```

```
but if I use IN instead of = here, this query works:
=====

EMPL_LAST_NAME
---------------
Blake

=====
rather silly example of a sub-select in a FROM clause:
=====

EMPL_LAST_NAME  DEPT_NAME
--------------- ---------------
Adams           Operations

=====
another rather silly example of a sub-select in a FROM clause:
=====

ENAME
---------------
Adams

SQL>
```

```
/*=====
   note, though: the outer select only SEES the columns as they
   are projected from the subquery in the FROM clause;
=====*/

prompt =====
prompt another rather silly example of a sub-select in a FROM clause:
prompt =====

select ename
from   (select empl_last_name ename, dept_name dname
        from   empl e, dept d
        where  e.dept_num = d.dept_num)
where   dname = 'Operations';

select empl_last_name ename, dept_name dname
from   empl e, dept d
where  e.dept_num = d.dept_num;
```

dl313 — dl313@nrs-projects:~/f24-325lect06-2 — ssh dl313@nrs-projects-ssh.humboldt.edu — 80×24

```
SQL> @ 325lect06-2.sql
```

dl313 — dl313@nrs-projects:~/f24-325lect06-2 — ssh dl313@nrs-projects-ssh.humboldt.edu — 80×24

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

ENAME          DNAME
-------------- --------------
Turner         Sales
Adams          Operations
King           Management

14 rows selected.

SQL>
```

dl313 — dl313@nrs-projects:~/f24-325lect06-2 — ssh dl313@nrs-projects-ssh.humboldt.edu — 80×24

```
prompt =====
prompt this causes an error; outer select only knows what is projected
prompt from subselect in its FROM clause:
prompt =====

select empl_last_name
from   (select empl_last_name ename, dept_name dname
        from   empl e, dept d
        where  e.dept_num = d.dept_num)
where   dname = 'Operations';
```

"325lect06-2.sql" 184L, 4617C written 184,0-1 Bot

dl313 — dl313@nrs-projects:~/f24-325lect06-2 — ssh dl313@nrs-projects-ssh.humboldt.edu — 80×24

```
SQL> @ 325lect06-2.sql
```

```
Ward            Sales
Blake           Sales
Michaels        Sales
James           Sales

ENAME           DNAME
--------------- ---------------
Turner          Sales
Adams           Operations
King            Management

14 rows selected.

=====
this causes an error; outer select only knows what is projected
from subselect in its FROM clause:
=====
select empl_last_name
       *
ERROR at line 1:
ORA-00904: "EMPL_LAST_NAME": invalid identifier


SQL>
```

dl313 — dl313@nrs-projects:~/f24-325lect06-2 — ssh dl313@nrs-projects-ssh.humboldt.edu — 80×24

```
prompt =====

select empl_last_name
from   (select empl_last_name ename, dept_name dname
        from   empl e, dept d
        where  e.dept_num = d.dept_num)
where   dname = 'Operations';

prompt =====
prompt this causes an error; outer select only knows what is projected
prompt from subselect in its FROM clause:
prompt =====

select ename
from   (select empl_last_name ename, dept_name dname
        from   empl e, dept d
        where  e.dept_num = d.dept_num)
where   dept_name = 'Operations';
```

"325lect06-2.sql" 193L, 4926C written 183,33 97%

dl313 — dl313@nrs-projects:~/f24-325lect06-2 — ssh dl313@nrs-projects-ssh.humboldt.edu — 80×24

```
SQL> @ 325lect06-2.sql
```

```
14 rows selected.

=====
this causes an error; outer select only knows what is projected
from subselect in its FROM clause:
=====
select empl_last_name
       *
ERROR at line 1:
ORA-00904: "EMPL_LAST_NAME": invalid identifier



=====
this causes an error; outer select only knows what is projected
from subselect in its FROM clause:
=====
where   dept_name = 'Operations'
        *
ERROR at line 5:
ORA-00904: "DEPT_NAME": invalid identifier


SQL>
```

dl313 — dl313@nrs-projects:~/f24-325lect06-2 — ssh dl313@nrs-projects-ssh.humboldt.edu — 80×24

```
SQL> select empl_last_name ename, dept_name dname
  2  from empl e, dept d
  3  where e.dept_num = d.dept_num;

ENAME           DNAME
--------------- ---------------
Miller          Accounting
Raimi           Accounting
Scott           Research
Jones           Research
Ford            Research
Smith           Research
Martin          Sales
Ward            Sales
Blake           Sales
Michaels        Sales
James           Sales

ENAME           DNAME
--------------- ---------------
Turner          Sales
Adams           Operations
King            Management
```

dl313 — dl313@nrs-projects:~/f24-325lect06-2 — ssh dl313@nrs-projects-ssh.humboldt.edu — 80×24

```
prompt =====

select ename
from   (select empl_last_name ename, dept_name dname
        from   empl e, dept d
        where  e.dept_num = d.dept_num)
where   dept_name = 'Operations';

prompt =====
prompt this causes an error; outer select only knows what is projected
prompt from subselect in its FROM clause:
prompt =====

select ename, salary
from   (select empl_last_name ename, dept_name dname
        from   empl e, dept d
        where  e.dept_num = d.dept_num)
where   dname = 'Operations';
```

"325lect06-2.sql" 202L, 5239C written 194,29 98%

dl313 — dl313@nrs-projects:~/f24-325lect06-2 — ssh dl313@nrs-projects-ssh.humboldt.edu — 80×24

```
SQL> @ 325lect06-2.sql
```

```
ORA-00904: "EMPL_LAST_NAME": invalid identifier


=====
this causes an error; outer select only knows what is projected
from subselect in its FROM clause:
=====
where   dept_name = 'Operations'
        *
ERROR at line 5:
ORA-00904: "DEPT_NAME": invalid identifier


=====
this causes an error; outer select only knows what is projected
from subselect in its FROM clause:
=====
select ename, salary
               *
ERROR at line 1:
ORA-00904: "SALARY": invalid identifier


SQL>
```

Exam 1 review

- Exam 1 date
- Exam 1 review guide and bonus points
- Exam 1 review and sample exam question

Exam 1 review

- Introduction to DBMSs
  - What is a DBMS? What is an example of a DBMS? What are some typical capabilities of a DBMS?

Exam 1 review

- Introduction to DBMSs
  - What is a DBMS? What is an example of a DBMS? What are some typical capabilities of a DBMS?
  - What is a database? What is a relational database?
    - the 4 main elements of a database? (user data, metadata, indexes, & application metadata)

Question

Which term below is the component of a database that makes it self-describing?

1. database management system (DBMS)
2. metadata
3. database schema/design
4. indexes

Exam 1 review

- Introduction to DBMSs
  - What is a DBMS? What is an example of a DBMS? What are some typical capabilities of a DBMS?
  - What is a database? What is a relational database?
    - the 4 main elements of a database? (user data, metadata, indexes, & application metadata)
  - What are some of the important limitations of file-processing systems?
    - What are some of the advantages of the database approach over these file-processing systems?

Exam 1 review

- Introduction to DBMSs
  - What is metadata?
  - What is a relation?
    - How do you represent a relation in relation structure form? ...or in tabular form? ...or in SQL create table statement form?
    - (possible question: here's a table/relation. Write a relation structure for it.)

Exam 1 review

- Introduction to DBMSs
  - What is metadata?
  - What is a relation?
    - How do you represent a relation in relation structure form? ...or in tabular form? ...or in SQL create table statement form?
    - (possible question: here's a table/relation. Write a relation structure for it.)
    - What are some of the important "restrictions“ or features of a relation?
    - (possible question: is the following a relation? Why not?)

Question

Based on the reading packets and course discussions, answer TRUE or FALSE:

A relation can have multiple rows with the same value for one of its non-primary-key attributes.

1. TRUE
2. FALSE

Exam 1 review

- Introduction to DBMSs
  - What is metadata?
  - What is a relation?
    - How do you represent a relation in relation structure form? ...or in tabular form? ...or in SQL create table statement form?
    - (possible question: here's a table/relation. Write a relation structure for it.)
    - What are some of the important "restrictions“ or features of a relation?
    - (possible question: is the following a relation? Why not?)
    - What is a primary key? How do you indicate a relation's primary key in relation structure form?

Exam 1 review

- Introduction to DBMSs
  - DBMS - database management system
    - What are some typical capabilities of a DBMS?
    - What is meant by DDL, DML, and DCL? What does each of these do?
    - What are some capabilities that a (high-end) DBMS might provide?

Exam 1 review

- Introduction to DBMSs
  - DBMS - database management system
    - What are some typical capabilities of a DBMS?
    - What is meant by DDL, DML, and DCL? What does each of these do?
    - What are some capabilities that a (high-end) DBMS might provide?
  - You should know that a database design/schema defines a database's structure, and typically includes:
    - its tables,
    - relationships,
    - domains, and
    - business rules
  - What are business rules?

Exam 1 review

- Introduction to DBMSs
  - How do you create tables in Oracle SQL? How do you define relationships between tables?

Exam 1 review

- Introduction to the relational model and relational operations
  - Who developed it? When? (Codd, 1970) Why was it first resisted?

Exam 1 review

- Introduction to the relational model and relational operations
  - Who developed it? When? (Codd, 1970) Why was it first resisted?
  - What is a relation? (including: What are the restrictions?)

Exam 1 review

- Introduction to the relational model and relational operations
  - Who developed it? When? (Codd, 1970) Why was it first resisted?
  - What is a relation? (including: What are the restrictions?)
    - Be comfortable with relation/table, tuple/row/record, attribute/column/field terminology
    - Single-valued cells, no duplicate rows, the order of rows/columns not important, the column entries all of same "kind"/from the same domain; must have a primary key

Exam 1 review

- Functional Dependencies and Key Definitions
  - What is a functional dependency? What does it mean for one attribute to be functionally dependent on another?
    - Understand the -> notation;
    - What is the determinant in A -> B ?

Exam 1 review

- Functional Dependencies and Key Definitions
  - What is a functional dependency? What does it mean for one attribute to be functionally dependent on another?
    - Understand the -> notation;
    - What is the determinant in A -> B ?
    - Given a relation in relation structure form, you should know what are some functional dependencies that you can assume?

Exam 1 review

- Functional Dependencies and Key Definitions
  - What is a functional dependency? What does it mean for one attribute to be functionally dependent on another?
    - Understand the -> notation;
    - What is the determinant in A -> B ?
    - Given a relation in relation structure form, you should know what are some functional dependencies that you can assume?
    - Does a determinant have to be a primary key? Why or why not?
    - Does a primary key have to be a determinant? Why or why not?
    - if (A, B) --> C, does it logically follow that A --> C and B --> C?
    - if A --> (B, C), does it logically follow that A --> B and A --> C?

Exam 1 review

- Functional Dependencies and Key Definitions
  - What is a superkey? ...a minimal key? ...a candidate key? ...a primary key?
    - How does one indicate a primary key in a relation structure?
    - How does one indicate a primary key in a SQL create table statement?
    - How many attributes may be in a primary key?

Question

Which term below is specifically defined as any set of attributes of a relation that uniquely determines a row of that relation?

1. primary key
2. candidate key
3. minimal key
4. superkey

Exam 1 review

- Relational Operations
  - The set-theoretic relational operations include union, difference, intersection, Cartesian product -- we have discussed only Cartesian product thus far;

Question

A relation ***Student*** has 3 rows, and a relation ***Class*** has 4 rows. How many rows are in the Cartesian product of stuff and nonsense?

1. 1
2. 7
3. 12
4. 0

Exam 1 review

- Relational Operations
  - The set-theoretic relational operations include union, difference, intersection, Cartesian product -- we have discussed only Cartesian product thus far;

Exam 1 review

- Relational Operations
  - The set-theoretic relational operations include union, difference, intersection, Cartesian product -- we have discussed only Cartesian product thus far;
  - The relation-theoretic relational operations include rename, selection, projection, equi-join/natural join/other joins, and division -- we have discussed only selection, projection, equi-join, and natural join so far;

Question

Which operation can be used to result in a relation containing ONLY the values for specified/requested attributes of a relation?

1. Selection
2. Projection
3. Natural Join
4. Equi-join

Exam 1 review

- Relational Operations
  - At this point, you are expected to know and understand selection, projection, equi-join, natural join, and Cartesian product;

Exam 1 review

- Relational Operations
  - At this point, you are expected to know and understand selection, projection, equi-join, natural join, and Cartesian product;
    - Expect to have to show that you can perform some or all of these operations on example relations; (note that I may describe the desired relational operation in words OR give it in SQL form);

Exam 1 review

- Relational Operations
  - At this point, you are expected to know and understand selection, projection, equi-join, natural join, and Cartesian product;
    - Expect to have to show that you can perform some or all of these operations on example relations; (note that I may describe the desired relational operation in words OR give it in SQL form);
    - What is an equi-join? How does it differ from a natural join?

Exam 1 review

- Relational Operations
  - At this point, you are expected to know and understand selection, projection, equi-join, natural join, and Cartesian product;
    - Expect to have to show that you can perform some or all of these operations on example relations; (note that I may describe the desired relational operation in words OR give it in SQL form);
    - What is an equi-join? How does it differ from a natural join?
    - Know the relation_name.attribute_name notation;

Exam 1 review

- Relational Operations
  - At this point, you are expected to know and understand selection, projection, equi-join, natural join, and Cartesian product;
    - Expect to have to show that you can perform some or all of these operations on example relations; (note that I may describe the desired relational operation in words OR give it in SQL form);
    - What is an equi-join? How does it differ from a natural join?
    - Know the relation_name.attribute_name notation;
    - You should be able to express queries as combinations of relational algebra operations.

Question

Based on the reading packets and course discussions, answer TRUE or FALSE:

One should determine which tables the user wants, and then create a database model that corresponds to those tables.

1. FALSE
2. TRUE

Exam 1 review

- Introduction to the Entity-Relationship Model
  - What is (should be) the database development process? What is a database model? What are some of the general strategies for developing a database model?
    - the idea is that you come up with a database model before you come up with tables! Why should you develop a data model/database model before starting to create database tables?

Exam 1 review

- Introduction to the Entity-Relationship Model
  - What is (should be) the database development process? What is a database model? What are some of the general strategies for developing a database model?
    - the idea is that you come up with a database model before you come up with tables! Why should you develop a data model/database model before starting to create database tables?
    - there is more than one type of database model; we are focusing on the most commonly-used, the entity-relationship model

Question

Based on the reading packets and course discussions, answer TRUE or FALSE:

One should consider one entity class to be equivalent to one table/relation.

1. FALSE
2. TRUE

Exam 1 review

- Introduction to the Entity-Relationship Model
  - You are responsible for the entity-relationship diagram (ERD) notation given in class and in the course handouts; see also the guidelines mentioned in the Lab and Homework handout.
  - What are the elements of the E-R model? (entities, attributes, identifying attributes, relationships, cardinalities)
  - What is an entity class? How is it depicted in an ERD? What is/are an entity class's identifying attribute(s)?

Exam 1 review

- Introduction to the Entity-Relationship Model
  - You are responsible for the entity-relationship diagram (ERD) notation given in class and in the course handouts; see also the guidelines mentioned in the Lab and Homework handout.
  - What are the elements of the E-R model? (entities, attributes, identifying attributes, relationships, cardinalities)
  - What is an entity class? How is it depicted in an ERD? What is/are an entity class's identifying attribute(s)?
  - What is a relationship? How is it depicted in an ERD?

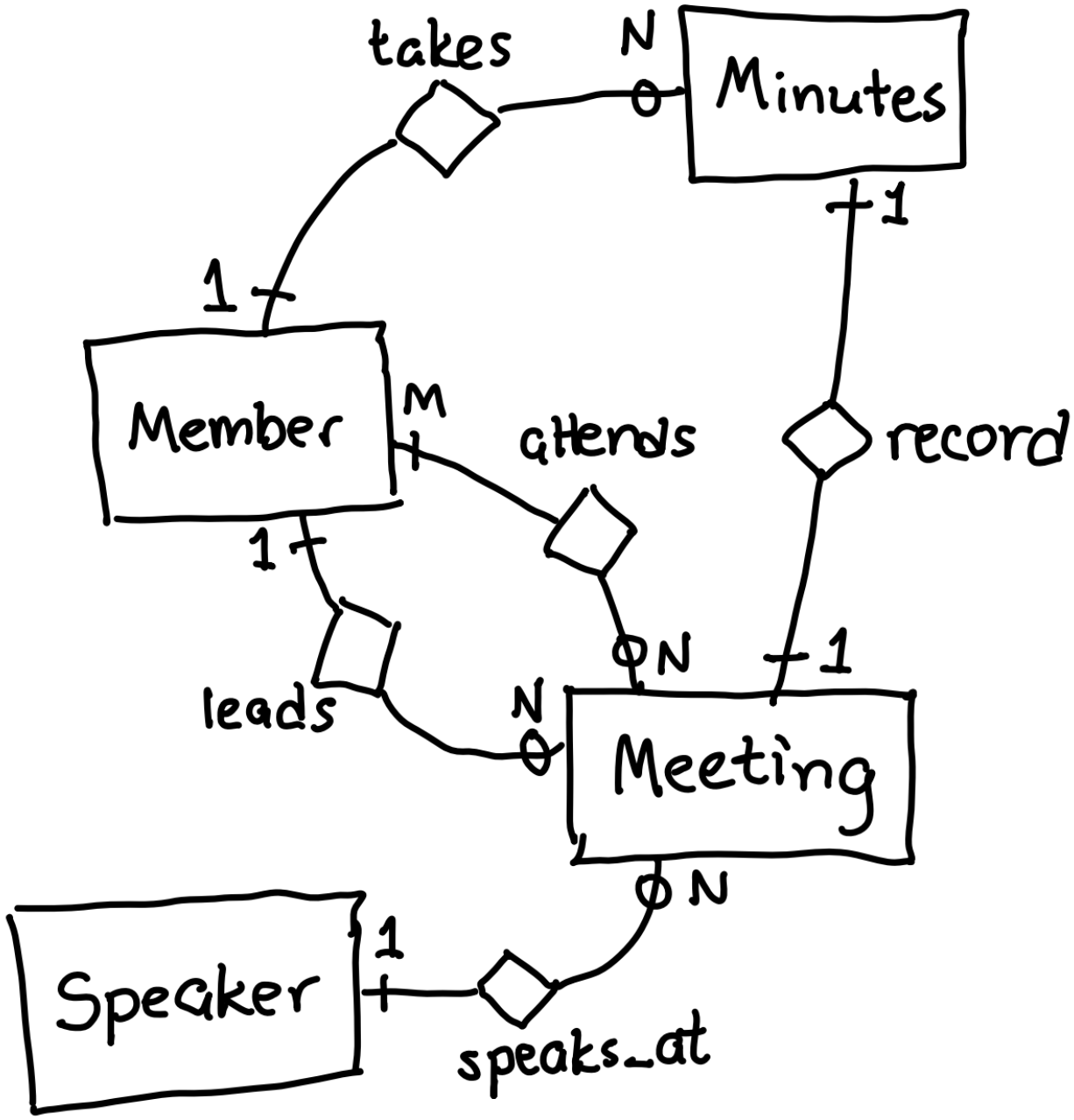
Exam 1 review

- Introduction to the Entity-Relationship Model
  - You are responsible for the entity-relationship diagram (ERD) notation given in class and in the course handouts; see also the guidelines mentioned in the Lab and Homework handout.
  - What are the elements of the E-R model? (entities, attributes, identifying attributes, relationships, cardinalities)
  - What is an entity class? How is it depicted in an ERD? What is/are an entity class's identifying attribute(s)?
  - What is a relationship? How is it depicted in an ERD?
  - What is an attribute? What is its domain? How is an attribute depicted in an ERD (according to our course ERD standards)?

Question

Does a speaker instance have to speak-at any meeting?

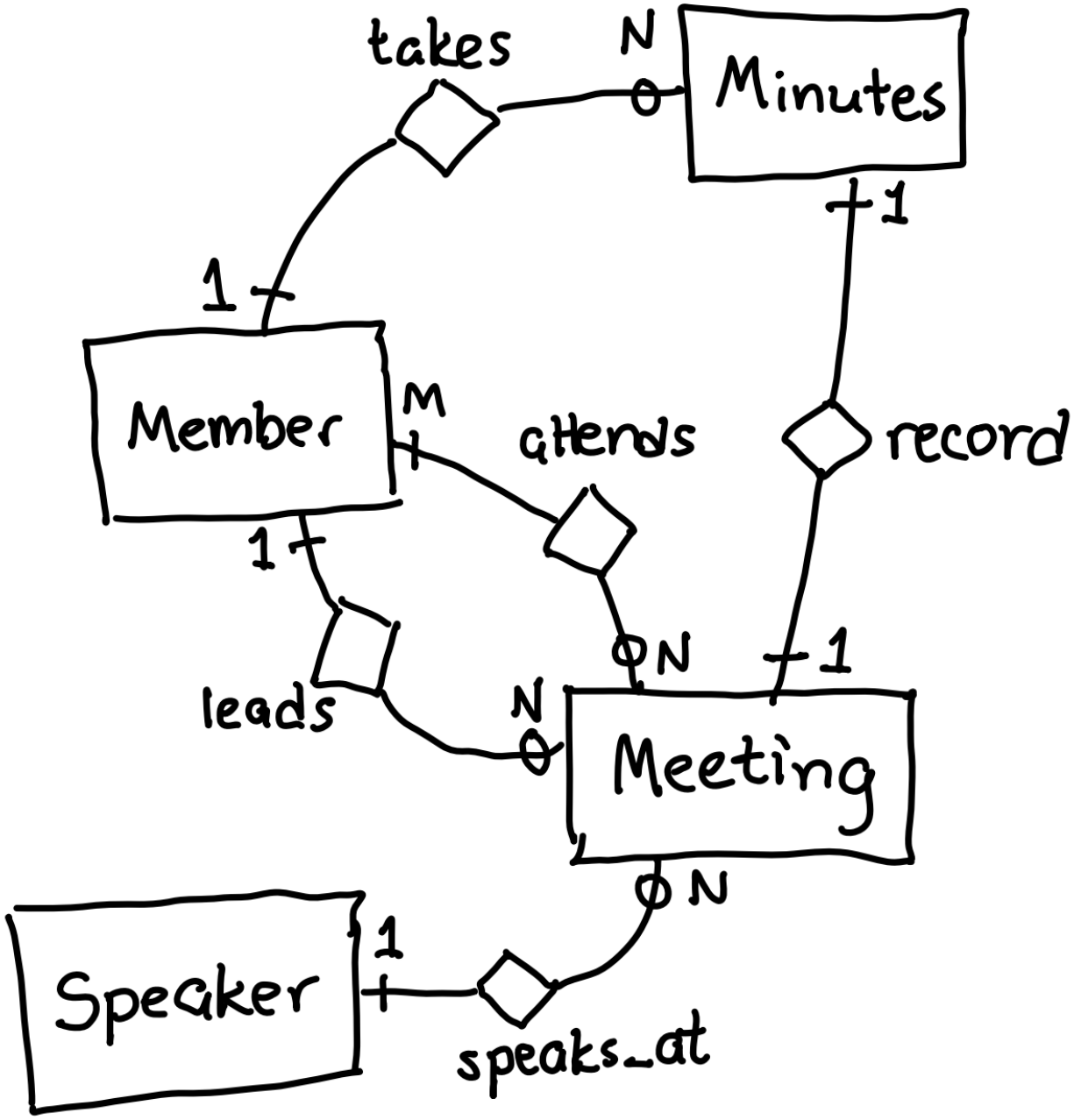
1. YES
2. NO



Question

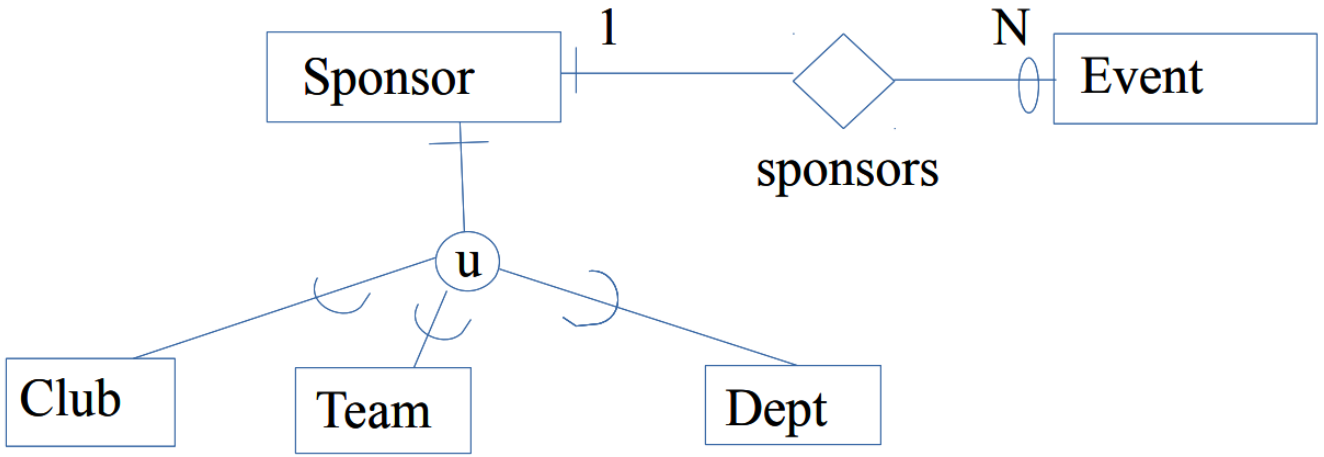
Can a minutes instance by more than one member?

1. YES
2. NO



Exam 1 review

- Introduction to the Entity-Relationship Model
  - What is an attribute? What is its domain? How is an attribute depicted in an ERD (according to our course ERD standards)?




```
Sponsor      Club          Team          Dept                Event
-------      --------      --------      ---------           -----
             CLUB_NUM      TEAM_CODE     DEPT_CODE           EVENT_NUM
             Club_Name     Sport         Dept_title          Event_title
             Is_active     Season        Office_num          Event_date
```

Exam 1 review

- Introduction to the Entity-Relationship Model
  - What is an attribute? What is its domain? How is an attribute depicted in an ERD (according to our course ERD standards)?
    - Make sure it is clear to you what is not an attribute in an ERD as well: attribute lists in an ERD should include no relationship-related information.

Exam 1 review

- Introduction to the Entity-Relationship Model
  - What is an attribute? What is its domain? How is an attribute depicted in an ERD (according to our course ERD standards)?
    - Make sure it is clear to you what is not an attribute in an ERD as well: attribute lists in an ERD should include no relationship-related information.
    - Only the relationship lines in the ERD show the relationships between entity classes!

Exam 1 review

- Introduction to the Entity-Relationship Model
  - What are maximum cardinalities of a relationship?
  - What are the possible maximum cardinality values (typically)? (one and many)
  - Based on the maximum cardinalities, what are the 3 (4) "kinds" of relationships? (1:1, 1:N, N:M)

Exam 1 review

- Introduction to the Entity-Relationship Model
  - What are maximum cardinalities of a relationship?
  - What are the possible maximum cardinality values (typically)? (one and many)
  - Based on the maximum cardinalities, what are the 3 (4) "kinds" of relationships? (1:1, 1:N, N:M)
  - Given the relevant information about a scenario, you should be able to determine which maximum cardinalities are appropriate for a relationship;

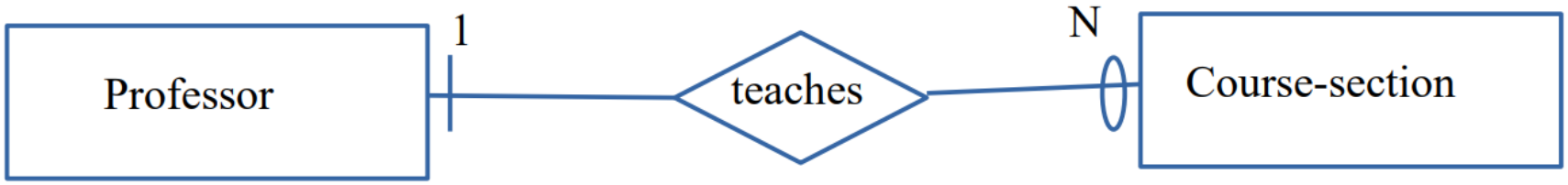
Exam 1 review

- Introduction to the Entity-Relationship Model
  - What are maximum cardinalities of a relationship?
  - What are the possible maximum cardinality values (typically)? (one and many)
  - Based on the maximum cardinalities, what are the 3 (4) "kinds" of relationships? (1:1, 1:N, N:M)
  - Given the relevant information about a scenario, you should be able to determine which maximum cardinalities are appropriate for a relationship;
  - According to our course ERD standards, how are maximum cardinalities depicted in an ERD? You should be able to read and understand an ERD's maximum cardinalities, and you should be able to create an ERD with appropriate maximum cardinalities;

Exam 1 review

- Introduction to the Entity-Relationship Model
  - What are minimum cardinalities of a relationship?
    - What are the possible minimum cardinality values (typically)? (0 and 1)
    - Given the pertinent information about a scenario, you should be able to determine which minimum cardinalities are appropriate for a relationship;
    - According to our course ERD standards, how are minimum cardinalities depicted in an ERD? You should be able to read and understand an ERD's minimum cardinalities, and you should be able to create an ERD with appropriate minimum cardinalities;

Question




```
Professor                    Course-section
-----------------            -----------------
PROF_SSN                     Course_name
Prof_lname                   COURSE_ID
Prof_email (MV)              Course_length
```

Consider the ERD, use it in answering this question, using only information provided in this diagram.

A course-section instance can have at most how many professor associated with it?

1. NONE
2. ONE
3. MANY

Exam 1 review

- Introduction to the Entity-Relationship Model
  - What is a supertype entity class? What is a subtype entity class?
    - How is a supertype/subtype relationship depicted in an ER diagram? (Remember to follow class style standards for these)
    - What is meant by having a d in the circle in depicting supertype/subtypes entity classes? ...having an o in that circle? ...having a u in that circle?
  - It is very likely that you will be asked questions about a given ERD, to see if you can read it correctly;
  - It is very likely that you will be asked to either draw or complete an ERD given scenario information;

Exam 1 review

- Basics of Oracle SQL and Oracle SQL*Plus
  - You will be required to read and write proper syntax SQL and SQL*Plus statements;
    - (By "read", I mean that I may give you a statement and ask you questions about it; I could also give you various table contents, and ask you what the results of running a given statement would be;)

Exam 1 review

- Basics of Oracle SQL and Oracle SQL*Plus
  - You will be required to read and write proper syntax SQL and SQL*Plus statements;
    - (By "read", I mean that I may give you a statement and ask you questions about it; I could also give you various table contents, and ask you what the results of running a given statement would be;)
  - How we start up SQL*Plus on nrs-projects?
  - How we create a table using SQL?
    - How we define attributes? What are some of the common data types?
    - How we define a table's primary key in SQL?
    - What is a foreign key? How we define a foreign key in SQL?
    - What kind of integrity checking do you get "automatically" in Oracle when you make an attribute a foreign key? ...a primary key?

Exam 1 review

- Basics of Oracle SQL and Oracle SQL*Plus
  - How we insert rows into a table? (know both variants)
    - Which version of insert do you need to use to make sure you get any default values for attributes that have them?

Exam 1 review

- Basics of Oracle SQL and Oracle SQL*Plus
  - How we insert rows into a table? (know both variants)
    - Which version of insert do you need to use to make sure you get any default values for attributes that have them?
  - How do you use a select statement to show the contents of a table?
  - What is a SQL script? How is it created? How is it run?
    - How can you write a SQL comment?

Exam 1 review

- Basics of Oracle SQL and Oracle SQL*Plus
  - You should be familiar with the SQL*Plus commands we have discussed so far, especially:
    - Which command can be used to list the column definitions for a table? (describe)
    - Which can be used to start spooling results to a file (and which can stop such spooling)? (spool file name and spool off)
    - Which can be used to execute a SQL script? (start)
    - Which can be used to output specified characters to the screen? (prompt)
  - How can you delete a table? (drop, probably with Cascade constraints)
  - What is the basic syntax and basic semantics of the SQL select statement?

Exam 1 review

- Writing "pure" relational operations using a SQL select statement
- How we write a relational selection using a SQL select statement?
- How can you write a relational Cartesian product using a SQL select statement?
- How can you write a relational equi-join using a SQL select statement? a relational natural join?
- How can these relational operations be combined within a single SQL select statement?
  - How can you express a query/question as a combination of relational algebra operations?
  - How can you express a query/question as a SQL select statement?

Exam 1 review

- More on the basic SQL select statement
  - Be familiar with the where clause possibilities discussed so far:
    - = < > <= >= <> !=
    - in
    - is null, is not null
    - and, or, not
    - between
    - like, %, _ (underscore)

Question

Write a SQL statement that will project just the empl_last_name from from the employee table just for those employees that have the letter 'M' anywhere in their last name.

select empl_last_name

from empl

_____

Exam 1 review

- More on the basic SQL select statement
  - aliases
    - What is a table alias (within the from clause)? what is a column alias (within the select clause)?
    - Why is a table alias useful? (2 reasons)
    - Why is a column alias useful?

Exam 1 review

- More on the basic SQL select statement
  - Computed columns
    - You should be able to read and write queries that project computed columns;
    - Make sure that you understand: whatever computations you choose to project from a select statement, projecting those computations does not change the contents of the database!

Exam 1 review

- More on the basic SQL select statement
  - Aggregate functions
    - avg, min, max, sum, count
    - Expect to have to read and/or write some of these;
    - Where can these be used within a select statement?
    - What effect do null values have with regard to these?
    - In a basic select statement (with just from and where clauses), either zero or how many rows will always be in the result of a select statement projecting an aggregate function?

Question

Consider the following query, assuming that currently there are 153 rows in the table empl, and that all empl instances must have a non-null salary:

```
select min(salary), max(salary)
from empl;
```

How many rows will be in the result?

1. 153
2. 2
3. cannot say for sure
4. 1