```
csc343h-dianeh=> \d
           List of relations
   Schema   |   Name   | Type  | Owner
------------+----------+-------+--------
 university | course   | table | dianeh
 university | offering | table | dianeh
 university | student  | table | dianeh
 university | took     | table | dianeh
(4 rows)

csc343h-dianeh=> select * from took;

-- It's very easy to aggregate across a whole column and produce the max.
csc343h-dianeh=> select max(grade) from took;
 max
-----
 100
(1 row)

-- Here we get the min of that same column.
csc343h-dianeh=> select min(grade) from took;
 min
-----
   0
(1 row)

-- Now we count how many oids there are.
csc343h-dianeh=> select count(oid) from took;
 count
-------
    54
(1 row)

-- There are a lot of duplicate oids, since typically more than one student
-- gets a grade in an offering of a course.  Here we count how many different
-- oids are in the table.
csc343h-dianeh=> select count(distinct oid) from took;
 count
-------
    23
(1 row)

-- We can also count simply the number of rows in the table.  In this table,
-- it's the same as the number of oids, which we counted above.
csc343h-dianeh=> select count(*) from took;
 count
-------
    54
(1 row)

-- We can combine aggregations into one table.
-- Here put all of the above aggregations together.  It doesn't matter that
-- they don't all pertain to the same column, and one isn't even about a single
-- column.  It DOES matter that we are dealing with like quantities: a single
-- max(grade), a single min(grade), etc.
csc343h-dianeh=> select max(grade), min(grade), count(distinct oid), count(*)
csc343h-dianeh-> from took;
 max | min | count | count
-----+-----+-------+-------
 100 |   0 |    23 |    54
(1 row)

-- But this is a problem.  Here we are combining sid (of which there are many)
-- with average grade (of which there is one).  That doesn't make sense.
```

```
csc343h-dianeh=> select sid, avg(grade) from took;
ERROR:  column "took.sid" must appear in the GROUP BY clause or be used in an aggregate function
LINE 1: select sid, avg(grade) from took;
               ^

-- Suppose we want each student and their own average grade.  In contrast to the
-- previous query, this would make sense: each student has one sid and one average grade.

-- Suggestion: before computing any aggregate values, let's use ORDER BY to see
-- the data organized in a way that would make calculating the aggregation
-- by hand easy.  This gives a feel for the data we are dealing with.
csc343h-dianeh=> select sid, grade from took order by sid;

  sid  | grade
-------+-------
   157 |    99
   157 |    82
   157 |    59
   157 |    72
   157 |    89
   157 |    39
   157 |    90
   157 |    98
   157 |    59
   157 |    71
   157 |    71
   157 |    91
   157 |    82
   157 |    62
   157 |    75
 11111 |    40
 11111 |     0
 11111 |    17
 11111 |    46
 11111 |    45
 98000 |    82
 98000 |    89
 98000 |    72
 . . . etc.

-- We want to collapse the sids down so that we see 157 once, 11111 once, etc.
-- "GROUP BY sid" will do that.
-- But then we'd better collapse the grades down too, so that there is only one
-- average grade value for 157, one for 11111 etc.  And that's exactly what we wanted to do:
-- we wanted the average per student.
-- So we make 2 changes to the ORDERY BY query above:
-- (1) Change "ORDER BY sid" to "GROUP BY sid".
-- (2) Instead of selecting grade, we select avg(grade).
csc343h-dianeh=> select sid, avg(grade) from took group by sid;
  sid  |         avg
-------+---------------------
 11111 | 29.6000000000000000
 98000 | 83.2000000000000000
 99132 | 76.2857142857142857
 99999 | 84.5833333333333333
   157 | 75.9333333333333333
(5 rows)

-- We can order this table too.  Here's one order.  Notice that "DESC" makes
-- the results go in descending order.
csc343h-dianeh=> select sid, avg(grade) from took group by sid order by sid desc;
  sid  |         avg
-------+---------------------
 99999 | 84.5833333333333333
 99132 | 76.2857142857142857
 98000 | 83.2000000000000000
 11111 | 29.6000000000000000
   157 | 75.9333333333333333
(5 rows)

-- Here we order by the aggregated column.
csc343h-dianeh=> select sid, avg(grade) from took group by sid order by avg(grade);
  sid  |         avg
-------+---------------------
 11111 | 29.6000000000000000
   157 | 75.9333333333333333
 99132 | 76.2857142857142857
 98000 | 83.2000000000000000
 99999 | 84.5833333333333333
(5 rows)

-- We can even order by something not in the SELECT.
-- QU: What will this do?  Ans: breaks because there is more than one OID per group.
csc343h-dianeh=> select sid, avg(grade) from took group by sid
csc343h-dianeh-> order by oid;
ERROR:  column "took.oid" must appear in the GROUP BY clause or be used in an aggregate function
LINE 2: order by oid;
                 ^

-- But this works because there is exactly one count(oid) per group:
csc343h-dianeh=> select sid, avg(grade) from took group by sid
order by count(oid);
  sid  |         avg
-------+---------------------
 11111 | 39.3333333333333333
 99132 | 76.2857142857142857
 99999 | 84.5833333333333333
 98000 | 83.2000000000000000
   157 | 75.9333333333333333
(5 rows)

-- We can order by and aggregate by multiple columns.
-- Order by 1 attribute:
csc343h-dianeh=> select * from offering order by dept;
 oid | cnum | dept | term  | instructor
-----+------+------+-------+------------
  11 |  200 | ANT  | 20089 | Zorich
  12 |  203 | ANT  | 20089 | Davies
  31 |  203 | ANT  | 20081 | Zorich
   4 |  320 | CSC  | 20089 | Jepson
   5 |  207 | CSC  | 20089 | Craig
   6 |  207 | CSC  | 20089 | Gries
   7 |  148 | CSC  | 20089 | Jepson
   8 |  148 | CSC  | 20089 | Chechik
etc.

-- Order by 2 attributes:
csc343h-dianeh=> select * from offering order by dept, cnum;
 oid | cnum | dept | term  | instructor
-----+------+------+-------+------------
  11 |  200 | ANT  | 20089 | Zorich
  31 |  203 | ANT  | 20081 | Zorich
  12 |  203 | ANT  | 20089 | Davies
   7 |  148 | CSC  | 20089 | Jepson
  27 |  148 | CSC  | 20081 | Jepson
   8 |  148 | CSC  | 20089 | Chechik
  28 |  148 | CSC  | 20081 | Miller
  25 |  207 | CSC  | 20081 | Craig
etc.
```

```
-- Group by the same 2 attributes:
-- This won't work (by now, you should know why):
csc343h-dianeh=> select * from offering group by dept, cnum;
ERROR:  column "offering.oid" must appear in the GROUP BY clause or be used in an aggregate function
LINE 1: select * from offering group by dept, cnum;
-- This works:
csc343h-dianeh=> select dept, cnum, count(cnum) from offering group by dept, cnum;
 dept | cnum | count
------+------+-------
 ENV  |  200 |     1
 ENG  |  235 |     2
 CSC  |  207 |     4
 EEB  |  216 |     1
 EEB  |  263 |     2
 HIS  |  296 |     1
 CSC  |  343 |     5
 HIS  |  220 |     2
 CSC  |  263 |     3
 ENG  |  205 |     2
 EEB  |  150 |     1
 ANT  |  203 |     2
 CSC  |  320 |     2
 CSC  |  148 |     4
 ENG  |  110 |     2
 ENV  |  320 |     1
 ANT  |  200 |     1
(17 rows)
```