```
6.830 problem set 1
geofft

----Problem 1---
6.830=> select name from papers where lower(name) like ’%coffee%’ limit 10;
                                                        name
--------------------------------------------------------------------------------------------------------------------
 Comparing commodity prices in electronic and traditional auctions: empirical evidence from Indian coffee auctions.
 PARALLEL-TCOFFEE: A parallel multiple sequence aligner.
 HGA-COFFEE : Aligning Multiple Sequences by Hybrid Genetic Algorithm.
 Vista: interactive coffee-corner display.
 ITIL capacity management: More than charts over coffee.
 Neurobiology 101: A One Semester in the Neurosciences Instead of a Coffee Break.
 Mobiles can’t kiss and hug, so lets meet over coffee.
 CoffeeStrainer - Statically Checking Structural Constraints on Java Programs.
 CoffeeStrainer: Statically-Checked Constraints on the Definition and Use of Types in Java.
 Effects of the Office Environment on Health and Productivity 1: Effects of Coffee Corner Position.
(10 rows)

----Problem 2---
6.830=> select authors.name, papers.name from authors join paperauths on authors.id = paperauths.authid join papers on pap
erauths.paperid = papers.id join venue on papers.venue = venue.id where lower(papers.name) like ’%coffee%’ and venue.name=
’Commun. ACM’;
         name         |                                    name
----------------------+----------------------------------------------------------------------------
 Asjad M. Khan        | Wake up and smell the coffee: evaluation methodology for the 21st century.
 Rotem Bentzur        | Wake up and smell the coffee: evaluation methodology for the 21st century.
 Daniel Feinberg      | Wake up and smell the coffee: evaluation methodology for the 21st century.
 Amer Diwan           | Wake up and smell the coffee: evaluation methodology for the 21st century.
 Daniel von Dincklage | Wake up and smell the coffee: evaluation methodology for the 21st century.
 Han Lee              | Wake up and smell the coffee: evaluation methodology for the 21st century.
 Ben Wiedermann       | Wake up and smell the coffee: evaluation methodology for the 21st century.
 Thomas VanDrunen     | Wake up and smell the coffee: evaluation methodology for the 21st century.
 Maria Jump           | Wake up and smell the coffee: evaluation methodology for the 21st century.
 Samuel Z. Guyer      | Wake up and smell the coffee: evaluation methodology for the 21st century.
 Darko Stefanovic     | Wake up and smell the coffee: evaluation methodology for the 21st century.
 Stephen M. Blackburn | Wake up and smell the coffee: evaluation methodology for the 21st century.
 J. Eliot B. Moss     | Wake up and smell the coffee: evaluation methodology for the 21st century.
 Robin Garner         | Wake up and smell the coffee: evaluation methodology for the 21st century.
 Daniel Frampton      | Wake up and smell the coffee: evaluation methodology for the 21st century.
 Chris Hoffmann       | Wake up and smell the coffee: evaluation methodology for the 21st century.
 Aashish Phansalkar   | Wake up and smell the coffee: evaluation methodology for the 21st century.
 Kathryn S. McKinley  | Wake up and smell the coffee: evaluation methodology for the 21st century.
 Antony L. Hosking    | Wake up and smell the coffee: evaluation methodology for the 21st century.
 Martin Hirzel        | Wake up and smell the coffee: evaluation methodology for the 21st century.
(20 rows)
```

```
----Problem 3---
select papers.name from papers, paperauths join authors on paperauths.authid = authors.id where ...

----Problem 4---
6.830=> select venue.name from venue join papers on venue.id = papers.venue join paperauths on papers.id = paperauths.pape
rid where paperauths.authid = (select authors.id from authors where authors.name = ’David J. DeWitt’) group by venue.name
order by count(*) desc limit 10;
           name
---------------------------
 SIGMOD Conference
 VLDB
 SIGMOD Record
 ICDE
 VLDB J.
 PDIS
 IEEE Trans. Software Eng.
 ACM Trans. Database Syst.
 CoRR
 POS
(10 rows)

----Problem 5---
6.830=> select n, count(*) from (select count(*) as n from paperauths as p1 join paperauths as p2 on p1.paperid=p2.paperid
 and p1.authid<p2.authid group by p1.authid, p2.authid) as subquery group by n order by n desc;
  n  |  count
-----+---------
 317 |       1
 201 |       2
 138 |       1
 123 |       1
 121 |       1
 118 |       1
 117 |       2
 116 |       1
 110 |       1
 107 |       1
  99 |       1
  98 |       2
  97 |       3
  94 |       1
  92 |       2
  90 |       1
  89 |       2
  88 |       3
  83 |       1
  82 |       1
  80 |       2
```

```
78 |       3
76 |       3
75 |       3
74 |       1
73 |       3
72 |       1
71 |       3
70 |       2
69 |       1
68 |       1
67 |       6
66 |       7
65 |       3
64 |       2
63 |       4
62 |       6
61 |       5
60 |       5
59 |       6
58 |       9
57 |       9
56 |       7
55 |       3
54 |      11
53 |      11
52 |       7
51 |       8
50 |      11
49 |       8
48 |      12
47 |      13
46 |      18
45 |      31
44 |      31
43 |      28
42 |      32
41 |      30
40 |      40
39 |      35
38 |      41
37 |      38
36 |      64
35 |      44
34 |      58
33 |      70
32 |      98
31 |      79
```

```
  30 |     116
  29 |     116
  28 |     116
  27 |     157
  26 |     159
  25 |     177
  24 |     244
  23 |     280
  22 |     329
  21 |     413
  20 |     442
  19 |     510
  18 |     604
  17 |     734
  16 |     965
  15 |    1178
  14 |    1429
  13 |    1790
  12 |    2376
  11 |    3159
  10 |    4138
   9 |    5651
   8 |    8110
   7 |   11724
   6 |   17758
   5 |   28528
   4 |   50835
   3 |  102624
   2 |  286250
   1 | 1422107
(98 rows)
```

---Problem 6---

```
create local temp table coauth as select p1.authid as id1, p2.authid as id2, count(*) as n from paperauths as p1 join paperauths as p2 on p1.paperid=p2.paperid and p1.authid<p2.authid group by p1.authid, p2.authid;
select p1.authid, ..., pn.authid from paperauths as p1 join ... join paperauths as pn on pk.paperid=p{k+1}.paperid and pk.authid<p{k+1}.authid where (select n from paperauths where id1 = p1.authid and id2 = p2.authid) >= m and (select n from paperauths where id1 = p1.authid and id2 = p3.authid) >= m and ... and (select n from paperauths where id1 = p{n-1}.authid and id2 = pn.authid) >= m
```

SQL doesn’t provide a way to take the row [p1, p2, ..., pn] and transpose it for the self-join, so we need to list the n^2 where clauses manually. If we could transpose it into a table authsubset, then we could join coauth with (authsubset as a1, authsubset as a2) on id1 = a1.id and id2 = a2.id, and ensure that count(*) where n<m is zero.