Query 1:

```
 store |  hol_sales
-------+-------------
    33 | 2.62594e+06
    20 | 2.24903e+07
(2 rows)
```

Query 2:

```
 store
-------
    34
    43
(2 rows)
```

Query 3:

```
 count
-------
     8
(1 row)
```

Query 4:

The result has 36 rows. Showing results for "A".

```
 months | type |     sum     |  %contribution
--------+------+-------------+------------------
      1 | A    | 2.14176e+08 | 4.94517291870296
      2 | A    | 3.66506e+08 | 8.46237778111938
      3 | A    | 3.80773e+08 |  8.7917834357331
      4 | A    | 4.16181e+08 | 9.60933018828221
      5 | A    | 3.59085e+08 | 8.29102708191177
      6 | A    | 3.99448e+08 | 9.22297486205232
      7 | A    | 4.17243e+08 | 9.63384622468555
      8 | A    | 3.94862e+08 | 9.11709212455745
      9 | A    | 3.73119e+08 |  8.6150486806721
     10 | A    | 3.77133e+08 | 8.70773098700509
     11 | A    | 2.64721e+08 | 6.11222225453493
     12 | A    | 3.67763e+08 | 8.49139641617382
...
(36 rows)
```

Graduate students are expected to show month names.

```
 months | type |     sum     |  %contribution
--------+------+-------------+------------------
 Jan    | A    | 2.14176e+08 | 4.94517291870296
 Feb    | A    | 3.66506e+08 | 8.46237778111938
```

```
 Mar    | A    | 3.80773e+08 | 8.7917834357331
 Apr    | A    | 4.16181e+08 | 9.60933018828221
 May    | A    | 3.59085e+08 | 8.29102708191177
 Jun    | A    | 3.99448e+08 | 9.22297486205232
 Jul    | A    | 4.17243e+08 | 9.63384622468555
 Aug    | A    | 3.94862e+08 | 9.11709212455745
 Sep    | A    | 3.73119e+08 | 8.6150486806721
 Oct    | A    | 3.77133e+08 | 8.70773098700509
 Nov    | A    | 2.64721e+08 | 6.11222225453493
 Dec    | A    | 3.67763e+08 | 8.49139641617382
...
(36 rows)
```

Query 5:

Be careful to test this one, as the sample data results in 0 rows. For example, create a test "Sales" table where the result would be true for a couple of stores.

```
 store
-------
(0 rows)
```

Query 6:

```
    attribute     | corr_sign |      correlation
------------------+-----------+-----------------------
 Temperature      | -         |  -0.00231244659998809
 FuelPrice        | -         | -0.000120295860528548
 CPI              | -         |   -0.0209213356051743
 UnemploymentRate | -         |   -0.0258637151104456
(4 rows)
```

Query 7:

```
 dept |        avg
------+--------------------
   94 | 0.0304081355131024
   95 |  0.069525075952212
   40 | 0.0441973058713807
   92 | 0.0730967313879066
   91 | 0.0313699985724977
   93 | 0.0254024084353863
   90 | 0.0449520747280783
   38 | 0.0727544338338905
    2 | 0.0410644138852755
   72 | 0.0420093146144659
(10 rows)
```

0.030408154626558

Query 8:

```
 dept |    normsales
------+------------------
   92 | 4128.35283184452
   38 | 4080.21098287073
   95 |  3879.8351126117
   90 | 2567.52589305854
   40 | 2400.34807233329
    2 | 2232.72935979053
   72 | 2191.77409403543
   91 | 1791.72819385294
   94 | 1747.77832661447
   13 | 1620.50955989047
(10 rows)
```

Query 9:

There are 10 departments x 33 months of data:

```
 dept |  yr  | mo | monthlysales | contribution | cumulative_sales
------+------+----+--------------+--------------+-----------------
    2 | 2010 |  2 |  7.65827e+06 |         2.73 |       7658270.00
    2 | 2010 |  3 |  7.54055e+06 |         2.69 |      15198800.00
    2 | 2010 |  4 |  9.65966e+06 |         3.44 |      24858500.00
    2 | 2010 |  5 |  7.75584e+06 |         2.76 |      32614300.00
    2 | 2010 |  6 |  8.02598e+06 |         2.86 |      40640300.00
    2 | 2010 |  7 |  1.02527e+07 |         3.65 |      50893000.00
    2 | 2010 |  8 |   8.4787e+06 |         3.02 |      59371700.00
    2 | 2010 |  9 |  7.87961e+06 |         2.81 |      67251300.00
...
   95 | 2012 |  9 |  1.27289e+07 |         2.83 |     437004000.00
   95 | 2012 | 10 |   1.2316e+07 |         2.74 |     449320000.00
(330 rows)
```

Query 10:

Here a partial output. There are 15 rows.

```
  yr  | qtr | store_a_sales | store_b_sales
------+-----+---------------+---------------
 2010 |   1 |   2.38155e+08 |   1.11851e+08
 2010 |   2 |   3.90788e+08 |    1.8321e+08
 2010 |   3 |   3.82692e+08 |   1.78504e+08
 2010 |   4 |    4.5379e+08 |   2.16412e+08
 2010 |     |   1.46542e+09 |    6.8997e+08
 2011 |   1 |    3.4185e+08 |   1.53904e+08
...
 2012 |   4 |    1.1864e+08 |   5.39716e+07
 2012 |     |   1.28737e+09 |     5.866e+08
(15 rows)
```

Part 2:

If you still stuck on how to do the sampling without fetching the entire table, here is a hint.

Think about how you'd do it if one asked for just one random sample. Using Knuth's algorithm, you can pretend to iterate through the table and stop whenever the "toss" selects the row. Let's say this row is row 5.

Then using the row_number construct (to add a pretend rowid), you can fetch row numbered 5 from the table.

assume my table was structured as:

rownum	c1	c2
1	A	1
2	B	5
3	A	6
…		
1000	A	2

Then you can fetch the 5th row simply by

```
SELECT c1, c2
from table
WHERE rownum = 5;
```

Now you have to extend this idea to getting a set of rows.