Exercise 2

a)

```
                        QUERY PLAN
-------------------------------------------------------------
 Seq Scan on customers  (cost=0.00..721.00 rows=995 width=156)
   Filter: ((country)::text = 'Japan'::text)
(2 rows)
```

Exercise 3

b)

```
                                QUERY PLAN
-----------------------------------------------------------------------------
 Seq Scan on customers  (cost=0.00..721.00 rows=995 width=156)
   Filter: ((country)::text = 'Japan'::text)
(2 rows)
```

d)

```
                                    QUERY PLAN
-----------------------------------------------------------------------------------------
 Index Scan using customers_country on customers
(cost=0.00..56.66 rows=995 width=156)
   Index Cond: ((country)::text = 'Japan'::text)
(2 rows)
```

Exercise 4

b)

```
                              QUERY PLAN
-----------------------------------------------------------------------
 Hash Join  (cost=733.44..1004.41 rows=597 width=8)
   Hash Cond: (o.customerid = c.customerid)
   ->  Seq Scan on orders o  (cost=0.00..220.00 rows=12000
width=12)
   ->  Hash  (cost=721.00..721.00 rows=995 width=4)
         ->  Seq Scan on customers c  (cost=0.00..721.00 rows=995
width=4)
               Filter: ((country)::text = 'Japan'::text)
(6 rows)
```

c)

```
                              QUERY PLAN
-----------------------------------------------------------------------
 Merge Join  (cost=1803.59..1874.53 rows=597 width=8)
   Merge Cond: (c.customerid = o.customerid)
   ->  Sort  (cost=770.54..773.03 rows=995 width=4)
         Sort Key: c.customerid
```

```
        ->  Seq Scan on customers c  (cost=0.00..721.00 rows=995
width=4)
              Filter: ((country)::text = 'Japan'::text)
   ->  Sort  (cost=1033.04..1063.04 rows=12000 width=12)
         Sort Key: o.customerid
         ->  Seq Scan on orders o  (cost=0.00..220.00 rows=12000
width=12)
(9 rows)
```

d)

```
                                QUERY PLAN
-------------------------------------------------------------------------------
 Nested Loop  (cost=0.00..5749.36 rows=597 width=8)
   ->  Seq Scan on orders o  (cost=0.00..220.00 rows=12000
width=12)
   ->  Index Scan using customers_pkey on customers c
(cost=0.00..0.45 rows=1 width=4)
         Index Cond: (c.customerid = o.customerid)
         Filter: ((c.country)::text = 'Japan'::text)
(5 rows)
```

Exercise 5

a)

```
                                QUERY PLAN
-------------------------------------------------------------------------------
 Sort  (cost=1501.33..1501.36 rows=11 width=13)
   Sort Key: (avg(o.totalamount))
   ->  HashAggregate  (cost=1501.00..1501.14 rows=11 width=13)
         ->  Hash Join  (cost=921.00..1441.00 rows=12000 width=13)
               Hash Cond: (o.customerid = c.customerid)
               ->  Seq Scan on orders o  (cost=0.00..220.00 rows=12000
width=12)
               ->  Hash  (cost=671.00..671.00 rows=20000 width=9)
                     ->  Seq Scan on customers c  (cost=0.00..671.00
rows=20000 width=9)
(8 rows)
```

```
                                QUERY PLAN
-------------------------------------------------------------------------------
------------
 Sort  (cost=2325.52..2325.55 rows=11 width=13)
   Sort Key: (avg(o.totalamount))
   ->  HashAggregate  (cost=2325.19..2325.33 rows=11 width=13)
         ->  Merge Join  (cost=1033.15..2265.19 rows=12000
width=13)
               Merge Cond: (c.customerid = o.customerid)
               ->  Index Scan using customers_pkey on customers c
(cost=0.00..1002.25 rows=20000 width=9)
```

```
          ->  Sort  (cost=1033.04..1063.04 rows=12000 width=12)
                Sort Key: o.customerid
                ->  Seq Scan on orders o  (cost=0.00..220.00
rows=12000 width=12)
(9 rows)
```

b)

```
                                QUERY PLAN
-------------------------------------------------------------------------------------------------
--
 Merge Join  (cost=1033.15..2265.19 rows=12000 width=192)
   Merge Cond: (c.customerid = o.customerid)
   ->  Index Scan using customers_pkey on customers c
(cost=0.00..1002.25 rows=20000 width=156)
   ->  Sort  (cost=1033.04..1063.04 rows=12000 width=36)
         Sort Key: o.customerid
         ->  Seq Scan on orders o  (cost=0.00..220.00 rows=12000
width=36)
(6 rows)

                              QUERY PLAN
-------------------------------------------------------------------------------
 Sort  (cost=3783.54..3813.54 rows=12000 width=192)
   Sort Key: c.customerid
   ->  Hash Join  (cost=370.00..1861.00 rows=12000 width=192)
         Hash Cond: (c.customerid = o.customerid)
         ->  Seq Scan on customers c  (cost=0.00..671.00 rows=20000
width=156)
         ->  Hash  (cost=220.00..220.00 rows=12000 width=36)
               ->  Seq Scan on orders o  (cost=0.00..220.00 rows=12000
width=36)
(7 rows)
```

Exercise 6

a)

```
                          QUERY PLAN
-----------------------------------------------------------------
 Seq Scan on customers c  (cost=0.00..5001021.00 rows=6667
width=15)
   Filter: (4 < (SubPlan 1))
   SubPlan 1
     ->  Aggregate  (cost=250.00..250.01 rows=1 width=0)
           ->  Seq Scan on orders o  (cost=0.00..250.00 rows=1
width=0)
                 Filter: (customerid = $0)
(6 rows)
```

d)

```
                                                QUERY PLAN
-----------------------------------------------------------------------------------------------------------
 Merge Join  (cost=1033.15..3887.44 rows=12000 width=15)
   Merge Cond: (c.customerid = c.customerid)
   ->  GroupAggregate  (cost=1033.15..2565.19 rows=12000
width=4)
         Filter: (4 < count(*))
         ->  Merge Join  (cost=1033.15..2265.19 rows=12000 width=4)
               Merge Cond: (c.customerid = o.customerid)
               ->  Index Scan using customers_pkey on customers c
(cost=0.00..1002.25 rows=20000 width=4)
               ->  Sort  (cost=1033.04..1063.04 rows=12000 width=4)
                     Sort Key: o.customerid
                     ->  Seq Scan on orders o  (cost=0.00..220.00
rows=12000 width=4)
   ->  Index Scan using customers_pkey on customers c
(cost=0.00..1002.25 rows=20000 width=15)
(11 rows)
```

Exercise 7

a)

```
                                                QUERY PLAN
-----------------------------------------------------------------------------------------------------------
-------
 Sort  (cost=614926.51..614927.01 rows=199 width=130)
   Sort Key: ordercounts1.customerid
   ->  Subquery Scan ordercounts1  (cost=1008.88..614918.91
rows=199 width=130)
         Filter: (5 >= (SubPlan 1))
         ->  HashAggregate  (cost=1008.88..1016.35 rows=597
width=15)
               ->  Hash Join  (cost=733.44..1004.41 rows=597 width=15)
                     Hash Cond: (o.customerid = c.customerid)
                     ->  Seq Scan on orders o  (cost=0.00..220.00
rows=12000 width=4)
                     ->  Hash  (cost=721.00..721.00 rows=995 width=15)
                           ->  Seq Scan on customers c  (cost=0.00..721.00
rows=995 width=15)
                                 Filter: ((country)::text = 'Japan'::text)
         SubPlan 1
           ->  Aggregate  (cost=1028.29..1028.30 rows=1 width=0)
                 ->  HashAggregate  (cost=1010.38..1020.83 rows=597
width=15)
                       Filter: ($0 < count(*))
                       ->  Hash Join  (cost=733.44..1004.41 rows=597
width=15)
```

```
                      Hash Cond: (o.customerid = c.customerid)
                      ->  Seq Scan on orders o  (cost=0.00..220.00
rows=12000 width=4)
                      ->  Hash  (cost=721.00..721.00 rows=995
width=15)
                            ->  Seq Scan on customers c
(cost=0.00..721.00 rows=995 width=15)
                                  Filter: ((country)::text = 'Japan'::text)
(21 rows)
```

d)

```
                                                QUERY PLAN
-----------------------------------------------------------------------------------------------
----------------------
 Sort  (cost=12950.52..12952.01 rows=597 width=31)
   Sort Key: c.customerid
   ->  Hash Join  (cost=12916.68..12922.99 rows=597 width=31)
         Hash Cond: (c.customerid = c.customerid)
         ->  HashAggregate  (cost=10952.47..10955.97 rows=200
width=4)
               Filter: (count(*) <= 5)
               ->  Nested Loop Left Join  (cost=2028.81..10061.45
rows=118803 width=4)
                     Join Filter: ((count(*)) < ocj2.numorders)
                     ->  HashAggregate  (cost=1007.39..1014.85 rows=597
width=4)
                           ->  Hash Join  (cost=733.44..1004.41 rows=597
width=4)
                                 Hash Cond: (o.customerid = c.customerid)
                                 ->  Seq Scan on orders o  (cost=0.00..220.00
rows=12000 width=4)
                                 ->  Hash  (cost=721.00..721.00 rows=995
width=4)
                                       ->  Seq Scan on customers c
(cost=0.00..721.00 rows=995 width=4)
                                             Filter: ((country)::text = 'Japan'::text)
                     ->  Materialize  (cost=1021.42..1027.39 rows=597
width=8)
                           ->  Subquery Scan ocj2  (cost=1007.39..1020.82
rows=597 width=8)
                                 ->  HashAggregate  (cost=1007.39..1014.85
rows=597 width=4)
                                       ->  Hash Join  (cost=733.44..1004.41
rows=597 width=4)
                                             Hash Cond: (o.customerid =
c.customerid)
                                             ->  Seq Scan on orders o
(cost=0.00..220.00 rows=12000 width=4)
```

```
                                  ->  Hash  (cost=721.00..721.00
rows=995 width=4)
                                        ->  Seq Scan on customers c
(cost=0.00..721.00 rows=995 width=4)
                                              Filter: ((country)::text =
'Japan'::text)
      ->  Hash  (cost=1956.75..1956.75 rows=597 width=27)
            ->  Hash Join  (cost=1928.39..1956.75 rows=597
width=27)
                  Hash Cond: (c.customerid = c.customerid)
                  ->  HashAggregate  (cost=1007.39..1014.85 rows=597
width=4)
                        ->  Hash Join  (cost=733.44..1004.41 rows=597
width=4)
                              Hash Cond: (o.customerid = c.customerid)
                              ->  Seq Scan on orders o  (cost=0.00..220.00
rows=12000 width=4)
                              ->  Hash  (cost=721.00..721.00 rows=995
width=4)
                                    ->  Seq Scan on customers c
(cost=0.00..721.00 rows=995 width=4)
                                          Filter: ((country)::text = 'Japan'::text)
                  ->  Hash  (cost=671.00..671.00 rows=20000
width=15)
                        ->  Seq Scan on customers c  (cost=0.00..671.00
rows=20000 width=15)
(36 rows)
```