

Christopher Ashley Cline  
INLS 723: Databases III  
Arcot Rajasekar  
2012-10-10

## Database Vacuuming Exercise

### Inserting Rows

```
test=# insert into test2 (id, name)
test=# values
test=# (1,'blah'),
test=# (1,'blah'),
test=# (1,'blah'),
test=# (1,'blah'),
test=# (1,'blah'),
```

**[etc...]**

```
test=# select count(*) from test2;
count
-----
   556
(1 row)
```

### Initial Vacuuming

```
test=# vacuum VERBOSE ANALYZE;
INFO: vacuuming "pg_catalog.pg_statistic"
INFO: index "pg_statistic_relid_att_inh_index" now contains 373 row versions in 5 pages
DETAIL: 0 index row versions were removed.
0 index pages have been deleted, 0 are currently reusable.
CPU 0.00s/0.00u sec elapsed 0.00 sec.
INFO: "pg_statistic": found 0 removable, 373 nonremovable row versions in 22 out of 22
pages
DETAIL: 0 dead row versions cannot be removed yet.
There were 254 unused item pointers.
0 pages are entirely empty.
CPU 0.00s/0.00u sec elapsed 0.00 sec.
```

**[...]**

**[Relevant Section]**

```
INFO: vacuuming "public.test2"
INFO: scanned index "test2_pkey" to remove 1 row versions
DETAIL: CPU 0.00s/0.00u sec elapsed 0.00 sec.
INFO: "test2": removed 1 row versions in 1 pages
DETAIL: CPU 0.00s/0.00u sec elapsed 0.00 sec.
INFO: index "test2_pkey" now contains 555 row versions in 4 pages
DETAIL: 1 index row versions were removed.
0 index pages have been deleted, 0 are currently reusable.
CPU 0.00s/0.00u sec elapsed 0.00 sec.
INFO: "test2": found 0 removable, 555 nonremovable row versions in 4 out of 4 pages
DETAIL: 0 dead row versions cannot be removed yet.
There were 0 unused item pointers.
0 pages are entirely empty.
CPU 0.00s/0.00u sec elapsed 0.00 sec.
INFO: analyzing "public.test2"
INFO: "test2": scanned 4 of 4 pages, containing 555 live rows and 0 dead rows; 555 rows in
sample, 555 estimated total rows
```

**[...]**

## Initial Discussion

This initial vacuuming process changed next to nothing and took next to no time to complete. It deleted a single index row reference.

## Deletion of Random Rows

```
test=# delete from test2 where test2.new_id in (select ceil(random()*count(*)) as rand_row
from test2);
DELETE 1
test=# delete from test2 where test2.new_id in (select ceil(random()*count(*)) as rand_row
from test2);
DELETE 1
test=# delete from test2 where test2.new_id in (select ceil(random()*count(*)) as rand_row
from test2);
DELETE 1
```

**[etc...]**

```
test=# select count(*) from test2;
 count
-----
   429
(1 row)
```

## Second Vacuuming

```
test=# vacuum verbose analyze test2;
INFO: vacuuming "public.test2"
INFO: scanned index "test2_pkey" to remove 126 row versions
DETAIL: CPU 0.00s/0.00u sec elapsed 0.00 sec.
INFO: "test2": removed 126 row versions in 3 pages
DETAIL: CPU 0.00s/0.00u sec elapsed 0.00 sec.
INFO: index "test2_pkey" now contains 429 row versions in 4 pages
DETAIL: 126 index row versions were removed.
0 index pages have been deleted, 0 are currently reusable.
CPU 0.00s/0.00u sec elapsed 0.00 sec.
INFO: "test2": found 67 removable, 429 nonremovable row versions in 4 out of 4 pages
DETAIL: 0 dead row versions cannot be removed yet.
There were 1 unused item pointers.
0 pages are entirely empty.
CPU 0.00s/0.00u sec elapsed 0.01 sec.
INFO: analyzing "public.test2"
INFO: "test2": scanned 4 of 4 pages, containing 429 live rows and 0 dead rows; 429 rows in
sample, 429 estimated total rows
```

## Discussion of Second Vacuuming

During the second vacuuming process, 126 rows were removed from the table index, which corresponds to the number of rows randomly deleted from the table (555 rows initially, 429 subsequently). Interestingly, only 67 rows were removed from the table itself. I am not sure about the origins of that number, nor the origins of the single unused item pointer. Message boards online seem to indicate that unused item pointers correspond to empty space which can be reclaimed by running a full vacuum on the table. When I ran a full, verbose vacuum on this table, however, it made no mention of cleaning up an unused item pointer.