**Useful Queries to check Indexes and its Utilization**

--List all Indexes in public schema:

SELECT

tablename as "TableName",

indexname as "Index Name",

indexdef as "Index script"

FROM

pg\_indexes

WHERE

schemaname = 'public'

ORDER BY

tablename,

indexname;

--List all the indexes in a table and whether it is Primary or Unique key:

select

c.relnamespace::regnamespace as schema\_name,

c.relname as table\_name,

i.indexrelid::regclass as index\_name,

i.indisprimary as is\_pk,

i.indisunique as is\_unique

from pg\_index i

join pg\_class c on c.oid = i.indrelid

where c.relname = 'pgbench\_tellers';

-- Unused Indexes:

select \* from pg\_stat\_all\_indexes where idx\_scan = 0 and schemaname='public';

or

SELECT

relname AS table\_name,

indexrelname AS index\_name,

pg\_size\_pretty(pg\_relation\_size(indexrelid)) AS index\_size,

idx\_scan AS index\_scan\_count

FROM

pg\_stat\_user\_indexes

WHERE

idx\_scan < 100

ORDER BY

index\_scan\_count ASC,

pg\_relation\_size(indexrelid) DESC;

-- Does table needs an Index:

SELECT relname, seq\_scan-idx\_scan AS too\_much\_seq, CASE WHEN seq\_scan-idx\_scan>0

THEN 'Missing/Ineff Index'

ELSE 'OK' END,

pg\_relation\_size(relname::regclass) AS rel\_size, seq\_scan, idx\_scan FROM pg\_stat\_all\_tables WHERE schemaname='public' AND pg\_relation\_size(relname::regclass)>80000 ORDER BY too\_much\_seq DESC;

-- How many indexes are in cache:

SELECT sum(idx\_blks\_read) as idx\_read, sum(idx\_blks\_hit) as idx\_hit FROM pg\_statio\_user\_indexes;

-- Index % usage:

SELECT relname, 100 \* idx\_scan / (seq\_scan + idx\_scan) percent\_of\_times\_index\_used, n\_live\_tup rows\_in\_table FROM pg\_stat\_user\_tables WHERE (seq\_scan + idx\_scan) > 0 ORDER BY n\_live\_tup DESC;

-- Duplicate Indexes:

SELECT ni.nspname || '.' || ct.relname AS "table",

ci.relname AS "dup index",

pg\_get\_indexdef(i.indexrelid) AS "dup index definition",

i.indkey AS "dup index attributes",

cii.relname AS "encompassing index",

pg\_get\_indexdef(ii.indexrelid) AS "encompassing index definition",

ii.indkey AS "enc index attributes"

FROM pg\_index i

JOIN pg\_class ct ON i.indrelid=ct.oid

JOIN pg\_class ci ON i.indexrelid=ci.oid

JOIN pg\_namespace ni ON ci.relnamespace=ni.oid

JOIN pg\_index ii ON ii.indrelid=i.indrelid AND

ii.indexrelid != i.indexrelid AND

(array\_to\_string(ii.indkey, ' ') || ' ') like (array\_to\_string(i.indkey, ' ') || ' %') AND

(array\_to\_string(ii.indcollation, ' ') || ' ') like (array\_to\_string(i.indcollation, ' ') || ' %') AND

(array\_to\_string(ii.indclass, ' ') || ' ') like (array\_to\_string(i.indclass, ' ') || ' %') AND

(array\_to\_string(ii.indoption, ' ') || ' ') like (array\_to\_string(i.indoption, ' ') || ' %') AND

NOT (ii.indkey::integer[] @> ARRAY[0]) AND -- Remove if you want expression indexes (you probably don't)

NOT (i.indkey::integer[] @> ARRAY[0]) AND -- Remove if you want expression indexes (you probably don't)

i.indpred IS NULL AND -- Remove if you want indexes with predicates

ii.indpred IS NULL AND -- Remove if you want indexes with predicates

CASE WHEN i.indisunique THEN ii.indisunique AND

array\_to\_string(ii.indkey, ' ') = array\_to\_string(i.indkey, ' ') ELSE true END

JOIN pg\_class ctii ON ii.indrelid=ctii.oid

JOIN pg\_class cii ON ii.indexrelid=cii.oid

WHERE ct.relname NOT LIKE 'pg\_%' AND

NOT i.indisprimary

ORDER BY 1, 2, 3;

Useful Functions to Find size of Objects:

pg\_size\_pretty() function to format the size.

pg\_relation\_size() function to get the size of a table.

pg\_total\_relation\_size() function to get the total size of a table.

pg\_database\_size() function to get the size of a database.

pg\_indexes\_size() function to get the size of an index.

pg\_total\_index\_size() function to get the size of all indexes on a table.

pg\_tablespace\_size() function to get the size of a tablespace.

pg\_column\_size() function to obtain the size of a column of a specific type.