

Practice 4

Index and Query Processing



hyunsikyoona@korea.ac.kr

Index Creation and Scan

- DB Setup & Table Information
- Index Creation and Scan

DB Setup & Table Information

- Type on cmd (**Note:** Change directory to PostgreSQL's bin directory)
 - `psql -U postgres [YOUR DB NAME (default: postgres)] < [YOUR FILE PATH]/table1.dump`
- Type on psql command line
 - `SET enable_bitmapscan=false;`
- A schema of the “table1” is as follows:

Attribute Name	Data type	Cardinality	Features
Sorted	Integer	<i>2000000</i>	Sorted
Unsorted	Integer	<i>2000000</i>	Unsorted
rndm	Integer	<i>100000</i>	Dummy field
Dummy	Char(40)	<i>1</i>	Dummy field

Index Creation and Scan

- Create two indexes on “table1”
 - Target attributes are “sorted” and “unsorted”
 - PostgreSQL automatically makes clustered and non-clustered index based on attribute’s data distribution
 - **Which is clustered index or non-clustered index?**
 - Type “\h create index” for detailed index creation syntax

Index Creation and Scan

- PostgreSQL supports following index-based query execution plans
 - Seq scan : All rows in a table are read sequentially
 - Index scan : some (or all) rows in a table are read after traversing an index
 - Index only scan : query is processed in an index, not accessing table data.
 - ‘EXPLAIN ANALYZE’ statement shows the query plan and execution time of the query
Example) EXPLAIN ANALYZE SELECT * FROM table1;
1. Make (and execute) three queries each of which uses *seq scan*, *index scan*, and *index only scan* respectively. (use the ‘explain analyze’)
 2. Make two queries using *clustered index* and *non-clustered index*. Compare their execution times.
 3. Execute and compare the following two queries. Explain why their query plans are different.
 - *SELECT sorted , rndm FROM table1 where sorted > 1999231 and rndm = 1005;*
 - *SELECT sorted , rndm FROM table1 where sorted < 1999231 and rndm = 1005;*

B-tree and Hash

- DB Setup & Table Information
- B-tree and Hash

DB Setup & Table Information

- Type on cmd (**Note:** Move to PostgreSQL's bin directory)
 - `psql -U postgres [YOUR DB NAME (default: postgres)] < [YOUR FILE PATH]/table_btree_hash_no.sql`
- Type on psql command line
 - `SET enable_bitmapscan=false;`
- Number of records : 10,000,000
- “table_btree” , “table_hash” and “table_noindex” have the exactly same records
- A schema of the tables are as follows:

Attribute Name	Data type	Data range
recordid	Integer	<i>0 ~ 10,000,000</i>
rndm	Integer	<i>0 ~ 100,000</i>
dummy	Char(40)	
recordid	Integer	<i>0 ~ 10,000,000</i>

B-tree and Hash

1. Create two indexes
 - Create indexes on attribute “recordid” in “table_btree” and “table_hash”
 - Create “b-tree” in “table_btree.recordid” column
 - Create “hash index” in “table_hash.recordid” column
 - Type “\h create index” for detailed index creation syntax
 - Use a method name “btree” for creating b-tree and “hash” for creating hash index
2. Run two queries. And compare the query execution plan and total execution time
 - *Select * from table_btree where recordid = 10001;*
 - *Select * from table_hash where recordid = 10001;*
3. Run two queries. And compare the query execution plan and total execution time
 - *Select * from table_btree where recordid > 250 and recordid < 550;*
 - *Select * from table_hash where recordid > 250 and recordid < 550;*

Use the ‘EXPLAIN ANALYZE’ statement

B-tree and Hash

4. Update a single “recordid” field in “table_btree”. And update a single “recordid” field in “table_noindex”. Then find a difference
Hint) Update recordid 9,999,997 to 9,999,998
5. Update 2,000,000 “recordid” fields in “table_btree”. And update 2,000,000 “recordid” fields in “table_noindex”. Then find a difference
Hint) Raise “recordid” fields 100% whose value is greater than 8,000,000 (This query will update 2,000,000 records)
6. Update all “recordid” fields in “table_btree”. And update all “recordid” fields in “table_noindex”. Then find a difference
Hint) Raise all “recordid” fields 10%

Use the ‘EXPLAIN ANALYZE’ statement

Homework

- Homework 6

Homework 5

- Follow the directives in this slides
- Take screenshots of your queries and execution results
- Submit your homework online (blackboard)
 - Deadline
 - 10:30 am, November 26th, 2019 (before class)
 - **Only PDF files** are accepted
 - **No late submissions**

End



KOREA UNIVERSITY
DATABASE LAB

hyunsikyon@korea.ac.kr