RIGA TECHNICAL UNIVERSITY
FACULTY OF COMPUTER SCIENCE AND INFORMATION TECHNOLOGY
INSTITUTE OF APPLIED COMPUTER SYSTEMS

Practical Assignment #3
“Database Management Systems”
Data Structures Comparison

Author: Emir Oğuz
Course, Group: DSP201, Group 1
Student Card No: 230ADB011

Checked: Andrejs Gaidukovs

2022 / 2023 Study Year

Content

1 Task – Data Structures Comparison

1. For choosen topic select two tables; the same table must be used for all structures.
2. Enter data in those two tables (minimum 20). The same data must be used for all structures.
3. Analyze following data structures:
    a. Heap-Organized Table. Two tables that are linked by one to many (1:N)
    b. B-Tree Index. One index for every table
    c. Bitmap Join Index.
    d. Indexed Cluster. The cluster combines two tables.
    e. Hash Cluster. The cluster combines two tables.
    f. Index Organized Tables.
4. Tables (CREATE TABLE), clusters (CREATE CLUSTER) and indexes (CREATE INDEX) must be used.
5. You need to create 3 queries (SELECT) that retrieves:
    a. all records;
    b. the total number of records;
    c. one record;
6. Take small screenshot where execution time of the query and the cost of the execution plan (COST) can be seen for every query (tip: 5 structures * 3 queries = 15 screenshots). Fill tables in conclusions the execution time of the query and the cost of the execution plan (COST) for every query for each index;
7. Write conclusion: at least 250 words;
8. Upload 2 files, types and naming:
    a. Report in Word: *DBMS_3_YourSurname.docx*;
    b. Sql script: *DBMS_3_YourSurname.sql*; - sql script should include create and drop operations for tables and indexes, insert operations;

2 Main Sections of Practical Work

2.1 Heap-Organized Table

2.1.1 Query all record

SQL | 0.017 seconds

OPERATION	OBJECT_NAME	OPTIONS	CARDINALITY	COST
SELECT STATEMENT			10	3
TABLE ACCESS	BOOKS	FULL	10	3

2.1.2 Query the total number of records

SQL | 0.013 seconds

OPERATION	OBJECT_NAME	OPTIONS	CARDINALITY	COST
SELECT STATEMENT			1	1
SORT		AGGREGATE	1	
INDEX	BOOKS_AUTHORS_FK	FULL SCAN	10	1

2.1.3 Query one record

SQL | 0.012 seconds

OPERATION	OBJECT_NAME	OPTIONS	CARDINALITY	COST
SELECT STATEMENT			1	1
TABLE ACCESS	BOOKS	BY INDEX ROWID	1	1
INDEX	PK_BOOKS	UNIQUE SCAN	1	1
Access Predicates				
ISBN='9780545010221'				

2.2 B-Tree Index

2.2.1 Query all record

SQL | 0.012 seconds

OPERATION	OBJECT_NAME	OPTIONS	CARDINALITY	COST
SELECT STATEMENT			10	3
TABLE ACCESS	BOOKS	FULL	10	3

2.2.2 Query the total number of records

SQL | 0.012 seconds

OPERATION	OBJECT_NAME	OPTIONS	CARDINALITY	COST
SELECT STATEMENT			1	1
SORT		AGGREGATE	1	
INDEX	IDX_BOOKS_AUTHO...	FULL SCAN	10	1

2.2.3 Query one record

SQL | 0.011 seconds

OPERATION	OBJECT_NAME	OPTIONS	CARDINALITY	COST
SELECT STATEMENT			1	1
TABLE ACCESS	BOOKS	BY INDEX ROWID	1	1
INDEX	PK_BOOKS	UNIQUE SCAN	1	1
Access Predicates				
ISBN='9780545010221'				

2.3 Bitmap Join Index

2.3.1 Query all record

SQL | 0.012 seconds

OPERATION	OBJECT_NAME	OPTIONS	CARDINALITY	COST
SELECT STATEMENT			10	3
TABLE ACCESS	BOOKS	FULL	10	3

2.3.2 Query the total number of records

SQL | 0.011 seconds

OPERATION	OBJECT_NAME	OPTIONS	CARDINALITY	COST
SELECT STATEMENT			1	2
SORT		AGGREGATE	1	
INDEX	PK_BOOKS	FAST FULL SCAN	10	2

2.3.3 Query one record

SQL | 0.01 seconds

OPERATION	OBJECT_NAME	OPTIONS	CARDINALITY	COST
SELECT STATEMENT			1	1
TABLE ACCESS	AUTHORS	BY INDEX ROWID	1	1
INDEX	PK_AUTHORS	UNIQUE SCAN	1	1

2.4 Clustered Index

2.4.1 Query all record

SQL | 0.011 seconds

OPERATION	OBJECT_NAME	OPTIONS	CARDINALITY	COST
SELECT STATEMENT			10	3
TABLE ACCESS	BOOKS	FULL	10	3

2.4.2 Query the total number of records

SQL | 0.011 seconds

OPERATION	OBJECT_NAME	OPTIONS	CARDINALITY	COST
SELECT STATEMENT			1	4
SORT		AGGREGATE	1	
VIEW	SYS.VM_NWVW_1		10	4
HASH		GROUP BY	10	4
TABLE ACCESS	BOOKS	FULL	10	3

2.4.3 Query one record

SQL | 0.009 seconds

OPERATION	OBJECT_NAME	OPTIONS	CARDINALITY	COST
SELECT STATEMENT			1	2
TABLE ACCESS	AUTHORS	CLUSTER	1	2
INDEX	IDX_AU_BK_CLUSTER	UNIQUE SCAN	1	1
Access Predicates				
AUTHORID='A01'				

2.5 Hash Cluster

2.5.1 Query all record

SQL | 0.012 seconds

OPERATION	OBJECT_NAME	OPTIONS	CARDINALITY	COST
SELECT STATEMENT			10	4
TABLE ACCESS	BOOKS	FULL	10	4

2.5.2 Query the total number of records

SQL | 0.01 seconds

OPERATION	OBJECT_NAME	OPTIONS	CARDINALITY	COST
SELECT STATEMENT			1	5
SORT		AGGREGATE	1	
VIEW	SYS.VM_NWVW_1		10	5
HASH		GROUP BY	10	5
TABLE ACCESS	BOOKS	FULL	10	4

2.5.3 Query one record

SQL | 0.009 seconds

OPERATION	OBJECT_NAME	OPTIONS	CARDINALITY	COST
SELECT STATEMENT			1	4
TABLE ACCESS	BOOKS	FULL	1	4
Filter Predicates				
ISBN='9780545010221'				

2.6 Index-Organized Tables

2.6.1 Query all record

SQL | 0.012 seconds

OPERATION	OBJECT_NAME	OPTIONS	CARDINALITY	COST
SELECT STATEMENT			10	10
INDEX	PK_BOOKS	FAST FULL SCAN	10	10

2.6.2 Query the total number of records

SQL | 0.012 seconds

OPERATION	OBJECT_NAME	OPTIONS	CARDINALITY	COST
SELECT STATEMENT			1	1
SORT		AGGREGATE	1	
INDEX	BOOKS_AUTHORS_FK	FULL SCAN	10	1

2.6.3 Query one record

SQL | 0.009 seconds

OPERATION	OBJECT_NAME	OPTIONS	CARDINALITY	COST
SELECT STATEMENT			1	2
INDEX	PK_AUTHORS	UNIQUE SCAN	1	1
Access Predicates				
AUTHORID='A01'				

3 Conclusions

Cost

	Heap-Organized Table	B-Tree Index	Bitmap Join Index	Clustered Index	Hash Cluster	Index-Organized Tables
All records	3	1	3	3	4	10
Count of records	1	1	2	4	5	1
Single record	1	1	1	2	4	2

Time

	Heap-Organized Table	B-Tree Index	Bitmap Join Index	Clustered Index	Hash Cluster	Index-Organized Tables
All records	0.017	0.012	0.012	0.011	0.013	0.012
Count of records	0.013	0.012	0.011	0.011	0.010	0.012
Single record	0.011	0.011	0.010	0.009	0.009	0.009

As part of this assignment, I learned how to use different indexes in a database management system. Throughout the process, I saw the differences in the indexes in terms of cost and time. I created different code blocks for the indexes that were wanted to be analyzed in the assignment and tried to analyze them one by one. It was quite an instructive work.

In conclusion, indexes are important components of database management systems, and knowledge of the many types and uses for them may enhance a database's functionality and effectiveness significantly. The most basic type of table without an indexing structure, useful for temporary data storage, is a Heap-Organized Table. Database management systems frequently employ B-Tree Indexes because of their effectiveness in searching and retrieving data from tables. Bitmap Join Indexes are a useful tool for speeding up queries that use join operations on many tables. To integrate two or more linked tables into a single entity and speed up related data retrieval, Clustered Index and Hash Cluster are utilized. An advanced type of table is an Index-Organized Table.

In summary, understanding the various index types, their use, and implementation may significantly boost a database's performance and effectiveness. The nature of the data, how frequently it is accessed, and the sorts of queries being run all influence the choice of index type. These elements will be taken into account by an efficient database management system when choosing the index to employ for a certain table or query. Database managers may make sure that their databases are optimized for efficiency and that their applications run as efficiently as possible by making effective use of indexes.

4 References

- Riga Technical University, Faculty of Computer Science and Information Technology, Institute of Applied Computer Systems, DSP201 – Database Management Systems, Presentations