Lab Session 08

8. Indexing Experimentation

Aim:

To evaluate the impact of various index types on query performance in a dataset, and to analyze the effectiveness of indexes in optimizing data retrieval operations.

Description:

Indexes are essential in database management systems to expedite data retrieval operations. Different index types (e.g., B-tree, Hash, Bitmap) have distinct characteristics that can significantly impact query performance. In this experiment, we aim to create multiple index types on a dataset, measure query performance, and analyze the influence of indexes on query execution time.

Prerequisites:

1. Understanding of database management systems and SQL.
2. Familiarity with index structures and their functionalities.
3. Basic knowledge of query optimization techniques.

Pre-Lab:

1. **Define what an index is in the context of a database.**

Speeds up data retrieval using a special data structure.

2. **Why are indexes crucial for efficient data retrieval?**

Reduces scanned rows, improving query speed.

3. **Explain the differences between B-tree, Hash, and Bitmap indexes. What are the strengths and weaknesses of each type?**

B-tree: good for range;

Hash: fast for equals;

Bitmap: best for few unique values.

4. **How does the choice of index type affect query performance in a database?**

Right type boosts speed; wrong type slows performance.

5. **Discuss the factors that influence the selection of an appropriate index type for a given dataset.**

Depends on query type, data size, and column uniqueness.

In Lab Task:

1. **Select a suitable dataset for experimentation. Here we taken Orders dataset as shown below.**

order_id	customer_id	product_id	order_date	quantity	total_price
1	101	201	2024-05-01	2	50.00
2	102	202	2024-05-02	1	25.00
3	103	201	2024-05-02	3	75.00
4	101	203	2024-05-03	2	60.00
5	104	204	2024-05-04	1	30.00
6	102	205	2024-05-05	4	100.00
7	101	201	2024-05-06	2	50.00
8	105	206	2024-05-06	3	90.00
9	103	207	2024-05-07	1	40.00
10	106	208	2024-05-08	2	55.00

```
CREATE TABLE orders (
    order_id INT PRIMARY KEY,
    customer_id INT,
    product_id INT,
    order_date DATE,
    quantity INT,
    total_price DECIMAL(10, 2)
);

INSERT INTO orders VALUES (1, 101, 201, '2024-05-01', 2, 50.00);
INSERT INTO orders VALUES (2, 102, 202, '2024-05-02', 1, 25.00);
INSERT INTO orders VALUES (3, 103, 201, '2024-05-02', 3, 75.00);
INSERT INTO orders VALUES (4, 101, 203, '2024-05-03', 2, 60.00);
INSERT INTO orders VALUES (5, 104, 204, '2024-05-04', 1, 30.00);
INSERT INTO orders VALUES (6, 102, 205, '2024-05-05', 4, 100.00);
INSERT INTO orders VALUES (7, 101, 201, '2024-05-06', 2, 50.00);
INSERT INTO orders VALUES (8, 105, 206, '2024-05-06', 3, 90.00);
INSERT INTO orders VALUES (9, 103, 207, '2024-05-07', 1, 40.00);
INSERT INTO orders VALUES (10, 106, 208, '2024-05-08', 2, 55.00);
```

2) Design a set of SQL queries representing various data retrieval operations, including simple lookups, range queries.

Simple Lookups:

```
SELECT * FROM orders WHERE customer_id = 101;

SELECT * FROM orders WHERE product_id = 201;
```

Range Queries:

```
SELECT * FROM orders WHERE order_date BETWEEN '2024-05-02' AND
'2024-05-05';

SELECT * FROM orders WHERE total_price > 50;
```

3) Create multiple index types (e.g., B-tree, Hash, Bitmap) on selected columns of the dataset.

```
CREATE INDEX idx_customer_btree ON orders(customer_id);

CREATE INDEX idx_product_hash ON orders USING HASH (product_id);

CREATE BITMAP INDEX idx_date_bitmap ON orders(order_date);
```

5) Normal Query with out INDEX to measure the execution time of queries

```
SELECT * FROM orders WHERE customer_id = 101;
```

6) Query with BTREE INDEXING to measure the execution time of queries

```
SELECT * FROM orders WHERE customer_id = 101;
```

Post-Lab:

1) What was the purpose of creating different index types on the dataset?

To speed up query performance.

2) Describe the process you followed to create the different index types. Which indexing techniques did you use?

Used SQL to create B-tree, hash, and composite indexes.

3) How did you measure the query performance before and after creating indexes?

Measured using EXPLAIN ANALYZE and execution time.

4) What were the key metrics used to analyze the impact of indexes on query performance?

Time, rows scanned, CPU, and disk I/O.

5) Did you observe any significant differences in query performance after implementing indexes? If so, elaborate on the changes.

Yes, faster SELECT queries and fewer scanned rows.

6) Discuss any challenges or limitations encountered during the experiment.

Slower inserts, more storage, and index creation time.

7) How do you think the choice of index type influenced query performance?

Right index type matched query pattern, improved speed.

8) Can you explain any trade-offs involved in selecting different index types?

Faster reads vs. slower writes and higher storage.

Students Signature

(For Evaluator's use only)

Comment of the Evaluator (if Any)	Evaluator's Observation Marks Secured:_____ out of _____ Full Name of the Evaluator: Signature of the Evaluator Date of Evaluation: