**Lab Exercise 9- Understanding PostgreSQL Index Types – B-Tree, Hash, GIN, GiST**

**Objective**

Understand the purpose, use cases, and creation of different PostgreSQL index types:

* B-Tree
* Hash
* GIN (Generalized Inverted Index)
* GiST (Generalized Search Tree)

**Prerequisites**

* PostgreSQL installed on Windows
* Access to SQL Shell (psql) or pgAdmin
* A working database and basic knowledge of SQL

**Step 1. Create a Test Table**

Connect to your database using SQL Shell (psql) or pgAdmin and run:

CREATE TABLE sample\_data (

id SERIAL PRIMARY KEY,

name TEXT,

tags TEXT[],

description TEXT,

location POINT

);

This table includes a normal text column, an array, and a geometric type for testing various index types.

**Part 1: B-Tree Index**

**Use Case**

Default index type in PostgreSQL. Best for sorting, range queries (<, >, =, BETWEEN), and equality checks.

**Steps**

1. Create a B-Tree index:

CREATE INDEX idx\_name\_btree ON sample\_data (name);

1. Query using indexed column:

SELECT \* FROM sample\_data WHERE name = 'Alice';

1. Check index usage:

EXPLAIN SELECT \* FROM sample\_data WHERE name = 'Alice';

**Part 2: Hash Index**

**Use Case**

Useful for **only equality comparisons** (=). Not suitable for range or sorting operations.

**Steps**

1. Create a hash index:

CREATE INDEX idx\_name\_hash ON sample\_data USING hash (name);

1. Query and test:

SELECT \* FROM sample\_data WHERE name = 'Alice';

EXPLAIN SELECT \* FROM sample\_data WHERE name = 'Alice';

Note: Hash indexes are less commonly used, as B-Tree often performs equally well.

**Part 3: GIN Index (Generalized Inverted Index)**

**Use Case**

Best for indexing **arrays**, **JSONB**, and **full-text search**.

**Steps**

1. Create a GIN index on the tags array:

CREATE INDEX idx\_tags\_gin ON sample\_data USING gin (tags);

1. Query using the array:

SELECT \* FROM sample\_data WHERE tags @> ARRAY['science'];

1. Check index usage:

EXPLAIN SELECT \* FROM sample\_data WHERE tags @> ARRAY['science'];

**Part 4: GiST Index (Generalized Search Tree)**

**Use Case**

Used for **complex data types** like geometric types, full-text search, and range types.

**Steps**

1. Create a GiST index on a point column:

CREATE INDEX idx\_location\_gist ON sample\_data USING gist (location);

1. Query based on proximity:

SELECT \* FROM sample\_data

WHERE location <@ box '((0,0),(10,10))';

1. Check index usage:

EXPLAIN SELECT \* FROM sample\_data

WHERE location <@ box '((0,0),(10,10))';

**Summary Table**

| **Index Type** | **Best For** | **Example Column** |
| --- | --- | --- |
| B-Tree | Equality, range, sorting | name, id |
| Hash | Fast equality lookup only | name |
| GIN | Arrays, JSONB, full-text search | tags |
| GiST | Geometric types, full-text, ranges | location (point), box |