Experiment 7

**Name** : Mohammad Wasi

**SAP ID** : 500110709

**Batch** : AIML B8

Aim: **. To understand the concepts of Index.**

**Objective :**

1. Create indexes on database tables to improve query performance.
2. Understand the importance of indexing in database optimization.
3. Implement different types of indexes, such as unique indexes, composite indexes, and function-based indexes.
4. Perform operations like creating, dropping, and analyzing indexes in an RDBMS environment.

Theory :

In SQL, indexes are essential data structures used to optimize database performance by facilitating efficient data retrieval operations. They function as lookup tables, allowing the database engine to quickly locate rows within a table based on the values stored in specific columns. By organizing and sorting data in a structured format, indexes enable faster query execution and enhance overall database efficiency.

The primary use of indexes is to improve query performance. When a query is executed, the database engine can utilize indexes to locate relevant rows without having to perform a full table scan. Instead, it can quickly narrow down the search space by utilizing the index's data structure, resulting in significantly faster response times, especially for large datasets.

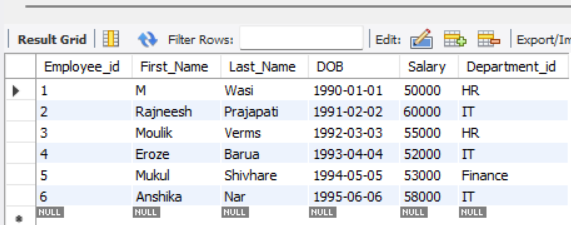
Additionally, indexes can enforce uniqueness constraints on columns, ensuring that no duplicate values are allowed. This is achieved through the creation of unique indexes, which prevent the insertion of duplicate values into indexed columns. Unique indexes are particularly useful for maintaining data integrity and enforcing data consistency within a database.

**Code**

**- 1) Create an index of name employee\_idx on EMPLOYEES with column Last\_Name, Department\_id**

CREATE INDEX employee\_idx ON EMPLOYES(Last\_Name, Department\_id);

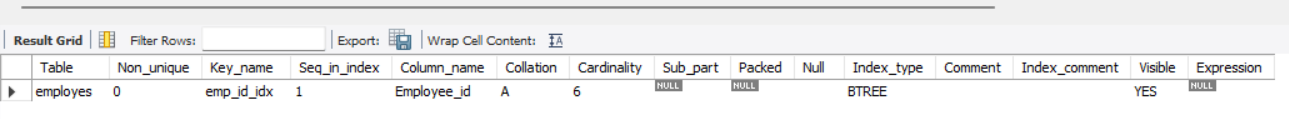
SELECT\* FROM EMPLOYES;



**-- 2) Find the ROWID for the above table and create a unique index on employee\_id column of the EMPLOYEES.**

CREATE UNIQUE INDEX emp\_id\_idx ON EMPLOYES(Employee\_id);

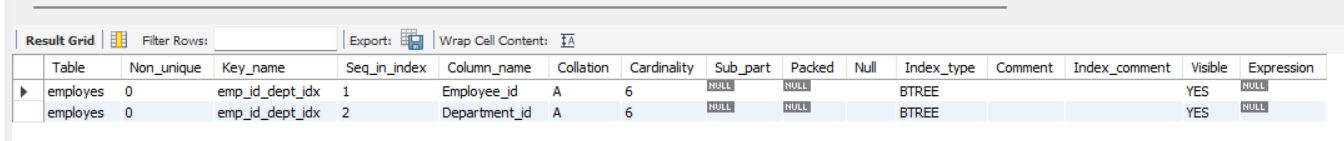
SHOW INDEX FROM EMPLOYES WHERE Key\_name = 'emp\_id\_idx';



**-- 3) Create a unique and composite index on employee\_id and check whether there is duplicity of tuples or not.**

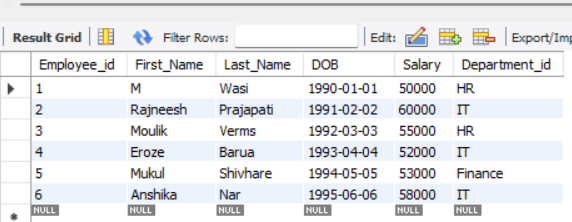
CREATE UNIQUE INDEX emp\_id\_dept\_idx ON EMPLOYES(Employee\_id, Department\_id);

SHOW INDEX FROM EMPLOYES WHERE key\_name='emp\_id\_dept\_idx';

**-- 4) Create a reverse index on employee\_id column of the EMPLOYEES.**

CREATE INDEX employee\_idxX ON EMPLOYES( Department\_id DESC);

SELECT\* FROM EMPLOYES;



**-- 6) Drop the function based index on column Last\_Name.**

CREATE INDEX upper\_last\_name\_idx ON EMPLOYES(Last\_Name);

DROP INDEX upper\_last\_name\_idx on EMPLOYES;

