

RDBMS FUNDAMENTALS

ASSIGNMENT 5:

Demonstrate the creation of an index on a table and discuss how it improves query performance. Use a DROP INDEX statement to remove the index and analyze the impact on query execution.

The screenshot shows the Oracle Database Express Edition interface. The SQL Commands window contains the following code:

```
// CREATE A TABLE
CREATE TABLE Emp ID number PRIMARY KEY, Name VARCHAR(20) NOT NULL, Department VARCHAR(20) NOT NULL, Salary DECIMAL(10,2);
// SHOW THE STRUCTURE OF THE TABLE
DESC Emp;
// INSERT THE VALUES TO THE TABLE
INSERT INTO Emp VALUES(1,'DEEPIKA NAIK','IT',35000);
INSERT INTO Emp VALUES(2,'PRIYANKA NAIK','HR',36000);
INSERT INTO Emp VALUES(3,'AJAY KUMAR','FINANCE',50000);
INSERT INTO Emp VALUES(4,'RUKMINI','IT',45000);
INSERT INTO Emp VALUES(5,'PUNYA','HR',46000);
//SHOW THE INSERTED VALUES FROM THE TABLE
SELECT * FROM Emp;
//CREATE AN INDEX
```

The Results window shows the following table structure:

ID	NAME	DEPARTMENT	SALARY
1	DEEPIKA NAIK	IT	35000
2	PRIYANKA NAIK	HR	36000
3	AJAY KUMAR	FINANCE	50000
4	RUKMINI	IT	45000
5	PUNYA	HR	46000

The screenshot shows the Oracle Database Express Edition interface. The SQL Commands window contains the following code:

```
//SHOW THE INSERTED VALUES FROM THE TABLE
SELECT * FROM Emp;
//CREATE AN INDEX
CREATE INDEX idx1 on Emp (Department);
//ANALYSE THE PERFORMANCE WITH THE INDEX
EXPLAIN PLAN FOR SELECT * FROM Emp WHERE Department='IT';
SELECT * FROM TABLE (DBMS_XPLAN.DISPLAY);
SELECT * FROM Emp WHERE Department='IT';
//DROP THE INDEX
DROP INDEX idx1;
//ANALYSE THE PERFORMANCE WITHOUT THE INDEX
EXPLAIN PLAN FOR SELECT * FROM Emp WHERE Department='IT';
SELECT * FROM TABLE (DBMS_XPLAN.DISPLAY);
SELECT * FROM Emp WHERE Department='IT';
```

The Results window shows the following table structure:

ID	NAME	DEPARTMENT	SALARY
1	DEEPIKA NAIK	IT	35000
4	RUKMINI	IT	45000

2 rows returned in 0.00 seconds

With Index :

- With the index , the query will likely use an index scan instead of a full table scan, improving performance significantly.
- Execution Plan : The query optimizer will use the index to quickly locate rows where Department='IT'
- Performance : The index scan is faster, especially with a large number of rows, because it doesn't have to scan the entire table. Only the relevant index entries are scanned.

Without Index :

- Without the index, the query execution plan will likely show a full table scan, which is generally slower than an index scan for large datasets.
- Execution Plan : The optimizer will perform a full table scan, checking each row to see if the Department matches 'IT'.
- Performance : Full table scans are typically slower because each row in the table must be examined, which is inefficient for large datasets.

Creating an index on the **Department** column significantly improves query performance by reducing the time needed to locate the relevant rows. When the index is dropped, the database resorts to a full table scan, which is much slower, especially as the table size grows. Proper indexing is crucial for optimizing database query performance, particularly for large datasets and frequently queried columns.