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.

SOLUTION:

1. **Faster Data Retrieval**: With the index in place, the database engine can quickly locate rows in the Books table that match a specific EMPLOYEEID without having to scan the entire table. This reduces the time required to retrieve relevant data, especially for large tables.
2. **Efficient Joins**: If you join the Books table with another table based on the EMPLOYEEID column, the index can significantly speed up the join operation by allowing the database engine to quickly match rows based on the indexed values.
3. **Improved Sorting**: If you need to sort the results of a query by EMPLOYEEID, the index can speed up the sorting process by providing an ordered list of values, reducing the need for the database engine to perform expensive sorting operations.

```
create table employee(
EmployeeID INT PRIMARY KEY,
FirstName VARCHAR(30),
LastName VARCHAR(30),
Department VARCHAR(30)
);
INSERT INTO
employee(EmployeeID,FirstName,LastName,Department)
VALUES(1,'S','PRANUSH','IT'),
```

```
(2,'K','HAYAAN','CYBER'),
(3,'P','VIHAAN','CSE');

CREATE INDEX idx_LastName ON employee(LastName);
select * from employee where LastName='PRANUSH';

DROP INDEX idx_LastName on employee;
select * from employee where LastName='PRANUSH';
```

OUTPUT:

```
1 •  SELECT * FROM assignment5.employee;
```

Result Grid | Filter Rows: | Edit:

EmployeeID	FirstName	LastName	Department
1	S	PRANUSH	IT
2	K	HAYAAN	CYBER
3	P	VIHAAN	CSE
NULL	NULL	NULL	NULL

```
13 •  CREATE INDEX idx_LastName ON employee(LastName);
14 •  select * from employee where LastName='PRANUSH';
15
```

esult Grid | Filter Rows: | Edit:

EmployeeID	FirstName	LastName	Department
1	S	PRANUSH	IT
NULL	NULL	NULL	NULL