Lab Scenario

The marketing system includes a query that is constantly executed and is performing too slowly. It retrieves 5000 web log entries beyond a given starting time. Previously, a non-clustered index was created on the SessionStart column. When 100 web log entries were being retrieved at a time, the index was being used. The developer is puzzled that changing the request to 5000 entries at a time has caused SQL Server to ignore the index he built. You need to investigate the query and suggest the best non-clustered index to support the query. You will then test your suggestion.

After you have created the new index, the developer noted the cost of the sort operation and tried to create another index that would eliminate the sort. You need to explain to him why SQL Server has decided not to use this index. Later you will learn to set up a basic query tuning trace in SQL Server Profiler and use the trace captured in Database Engine Tuning Advisor.

If time permits, you will design a required nonclustered index.

Supporting Documentation

Query 1: Query to test

```
DECLARE @StartTime datetime2 = '2010-08-30 16:27';

SELECT TOP(5000) wl.SessionID, wl.ServerID, wl.UserName
FROM Marketing.WebLog AS wl
WHERE wl.SessionStart >= @StartTime
ORDER BY wl.SessionStart, wl.ServerID;
```

Query 2: Current index definition

```
CREATE INDEX IX_WebLog_Perf_20100830_B
      ON Marketing.WebLog (ServerID, SessionStart)
      INCLUDE (SessionID, UserName);
```

Query 3: Query to review

```
SELECT PostalCode, Country
FROM Marketing.PostalCode
WHERE StateCode = 'KY'
ORDER BY StateCode, PostalCode;
```

Exercise 1: Nonclustered Index Usage Review

Scenario

The marketing system includes a query that is constantly executed and is performing too slowly. It retrieves 5000 web log entries beyond a given starting time. Previously, a non-clustered index was created on the SessionStart column. When 100 web log entries were being retrieved at a time, the index was being used. The developer is puzzled that changing the request to 5000 entries at a time has caused SQL Server to ignore the index he built. You need to investigate the query and suggest the best non-clustered index to support the query. You will then test your suggestion.

The main tasks for this exercise are as follows:

1. Review the query.
2. Review the existing Index and Table structures.
3. Design a more appropriate index.
4. Test your design.

▶ Task 1: Review the query

- Review Query 1 in the supporting documentation.

▶ Task 2: Review the existing Index and Table structures

- Review the existing Index and Table structures.

▶ Task 3: Design and implement a more appropriate index

- Design a more appropriate index.
- Implement the index that you have designed.

▶ Task 4: Test your design

- In the supporting documentation, use Query 1 to test your new index.

Results: After this exercise, you have created a non-clustered index.

Exercise 2: Improving Nonclustered Index Designs

Scenario

After you have created the new index, the developer noted the cost of the sort operation and tried to create another index that would eliminate the sort. Explain why SQL Server has decided not to use this index:

The main tasks for this exercise are as follows:

1. Review the index design.
2. Implement the index.
3. Test the design and explain why the index was not used.

▶ Task 1: Review the index design

- In Query 2 in the supporting documentation, review the index design.

▶ Task 2: Implement the index

- Create the index as per the index design.

▶ Task 3: Test the design and explain why the index was not used

- Enable Include Actual Execution Plan.
- Execute the query.
- Review the Execution Plan and explain why the index was not used.

Results: After this exercise, you have understood why some indexes are not appropriate in some scenarios.

Exercise 3: SQL Server Profiler and Database Engine Tuning Advisor

Scenario

Query 3 is another important query. You need to investigate the query and suggest the best non-clustered index to support the query. You will then test your suggestion.

The main tasks for this exercise are as follows:

1. Review the query.
2. Review the existing Index and Table structures.
3. Design a more appropriate index by following the Missing Index suggestion.
4. Create a better index that removes the sort operation. If you create another index, confirm that SQL Server selects it.

▶ Task 1: Review the query

- Review Query 3 in the supporting documentation.

▶ Task 2: Review the existing Index and Table structures

- Review the existing Index and Table structures.

▶ Task 3: Design a more appropriate index by following the Missing Index suggestion

- Review and implement the Missing Index that SQL Server has suggested.
- Test to ensure that the new index is being used.

▶ Task 4: Create a better index that removes the sort operation. If you create another index, confirm that SQL Server selects it

- Create a new index that will remove the Sort operation.
- Test to ensure that the new index is being used.

Results: After this exercise, you should have created a better index that will remove the sort operation.

Challenge Exercise 4: Nonclustered Index Design (Only if time permits)

Scenario

You will learn to set up a basic query tuning trace in SQL Server Profiler and to analyse the trace captured in DETA.

The main tasks for this exercise are as follows:

1. Open SQL Server Profiler and configure and start a trace.
2. Load and execute the workload file.
3. Stop and analyze the trace using DETA.

▶ Task 1: Open SQL Server Profiler and configure and start a trace

- Open SQL Server Profiler.
- Configure it use the following:
  a. Template: Tuning
  b. Save To File: should be selected and any file name provided for a file on the desktop
  c. Enable file rollover: Not selected
  d. Maximum File Size: 500MB
  e. Filter: DatabaseName LIKE MarketDev
- Start the SQL Server Profiler Trace.
- Disable AutoScroll from the Window Menu.

▶ Task 2: Load and execute the workload file

- Load and execute the workload file 81 – Lab Exercise 4.sql.

▶ Task 3: Stop and analyze the trace using DETA

- Stop the SQL Server Profiler trace.
- Analyze the trace results using DETA.
- Review the recommendations provided by the Database Tuning Advisor.

Results: After this exercise, you should have created a SQL Server Profiler trace and analyzed the recommendations from the Database Tuning Advisor.