# IBM Optim Query Workload Tuner for DB2 for z/OS®

## Hands-on Labs

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>INTRODUCTION</td>
<td>2</td>
</tr>
<tr>
<td>SINGLE QUERY TUNING</td>
<td>5</td>
</tr>
<tr>
<td>LAB 1 CUT COST AND OPTIMIZE PERFORMANCE</td>
<td>7</td>
</tr>
<tr>
<td>1.1 GETTING STARTED</td>
<td>8</td>
</tr>
<tr>
<td>1.2 CREATING A SAMPLE PROJECT</td>
<td>9</td>
</tr>
<tr>
<td>1.3 VIEWING QUERY ANNOTATIONS</td>
<td>12</td>
</tr>
<tr>
<td>1.4 VIEWING AN ACCESS PLAN GRAPH</td>
<td>19</td>
</tr>
<tr>
<td>LAB 2 TUNING A SINGLE QUERY</td>
<td>26</td>
</tr>
<tr>
<td>2.1 VIEWING STATISTICS ADVISOR RECOMMENDATIONS</td>
<td>26</td>
</tr>
<tr>
<td>2.2 VIEWING QUERY ADVISOR RECOMMENDATIONS</td>
<td>35</td>
</tr>
<tr>
<td>2.3 VIEWING INDEX ADVISOR RECOMMENDATIONS</td>
<td>44</td>
</tr>
<tr>
<td>LAB 3 VIEWING QUERY REPORTS</td>
<td>49</td>
</tr>
<tr>
<td>3.1.1 GENERATING QUERY REPORTS</td>
<td>49</td>
</tr>
<tr>
<td>WORKLOAD TUNING</td>
<td>52</td>
</tr>
<tr>
<td>LAB 4 SAMPLE WORKLOADS</td>
<td>53</td>
</tr>
<tr>
<td>4.1 VIEWING SAMPLE WORKLOADS</td>
<td>53</td>
</tr>
<tr>
<td>LAB 5 TUNING A WORKLOAD</td>
<td>55</td>
</tr>
<tr>
<td>5.1 VIEWING WORKLOADS</td>
<td>56</td>
</tr>
<tr>
<td>5.2 VIEWING WORKLOAD WITH STATISTICS ADVISORS RECOMMENDATIONS</td>
<td>59</td>
</tr>
<tr>
<td>5.3 VIEWING WORKLOAD WITH QUERY ADVISORS RECOMMENDATIONS</td>
<td>62</td>
</tr>
<tr>
<td>5.4 VIEWING WORKLOAD WITH INDEX ADVISORS RECOMMENDATIONS</td>
<td>65</td>
</tr>
<tr>
<td>LAB 6 VIEWING QUERY ENVIRONMENT CAPTURE</td>
<td>72</td>
</tr>
<tr>
<td>6.1 VIEWING QUERY ENVIRONMENT CAPTURE</td>
<td>72</td>
</tr>
<tr>
<td>6.2 OPTIONAL – REFERENCE ONLY : DEFINE A WORKLOAD USING THE WORKLOAD WIZARD</td>
<td>76</td>
</tr>
<tr>
<td>6.2.1 WORKLOAD SOURCE: DYNAMIC STATEMENT CACHE</td>
<td>76</td>
</tr>
<tr>
<td>CONCLUSION</td>
<td>84</td>
</tr>
</tbody>
</table>
Introduction

IBM Optim query tuning solutions, represented by Query Tuner (for single-query tuning) and Optim Query Workload Tuner for DB2 for z/OS (for query workload tuning), are both built on open-source Eclipse technology.

The first part of this lab covers the single query tuning provided by Optim Query Tuner for DB2 for z/OS, which provides the following benefits:

- Reduces costs and risks by enabling developers to tune SQL during development, while problems are still relatively inexpensive to fix and before they cause a costly outages or performance issues.
- Features seamless integration and natural launch points within Optim Development Studio.
- Accelerates query tuning analysis by providing expert advice and recommendations for indexes, statistics and queries.
- Fosters collaboration among developers and DBAs; DBAs can extend query tuning advice to SQL Workloads (multiple queries) by using Optim Query Workload Tuner for DB2 for z/OS.

The second part of the lab covers query workload tuning. Applications and workloads need to be tuned to optimize performance and maximize system capacity. Poorly performing SQL can have a variety of problems: inaccurate statistics, inefficiently designed SQL statements, missing indexes, and poorly selected access plans. IBM InfoSphere Optim Query Workload Tuner for DB2 for z/OS simplifies and speeds up analysis for DBAs, developers, designers and other roles in the organization. It not only removes much of the tedium from query analysis, but provides expert recommendations to improve query workload performance. The improved workload performance and return on investment generated by Query Workload Tuner directly benefits the business.

IBM InfoSphere Optim Query Workload Tuner for DB2 for z/OS delivers a unique solution to the marketplace. It fosters teamwork and collaboration by helping DBAs and developers gain insights into SQL performance, identify and tune poorly performing query workloads. DBAs can identify and tune poorly performing queries and workloads with tighter integration with other SQL tuning and monitoring solutions including DB2 Query Monitor, DB2 Path Check for z/OS, DB2 SQL Performance Analyzer for z/OS, and Tivoli OMEGAMON XE for DB2 Performance Expert on z/OS. Developers can launch InfoSphere Optim Query Workload Tuner from IBM Data Studio 4.14.1 and gain immediate insight into performance enhancements during development.

IBM InfoSphere Optim Query Workload Tuner for DB2 for z/OS provides an environment for performing query workload monitoring and tuning tasks. DBAs can work with Workload Access Plan Comparison, which helps compare cost and plans structure and Workload Access Plan Explorer that helps narrow down performance problems quicker. The following tools and advisors are provided along with the solution:

- **Workload Statistics Advisor** provides advice on missing, outdated, and conflicting statistics and suggests additional statistics that should be gathered to improve query performance for those queries within a workload.
- **Workload Query Advisor** makes recommendations based on best practice rules to modify SQL to improve overall performance.
- **Workload Index Advisor** provides advice on new indexes that may improve performance of a workload. With Workload Index Advisor, use a set of predefined scenarios to perform an analysis based on the scenario selected.
• **Workload Table Report** identifies key statistics, access path information, and physical database design information related to queries within a workload.
• **Workload Access Plan Explorer** provides tabular view of the query plan for quicker diagnosis of performance issues
• **Workload Access Plan Comparison** helps compare cost and plan structure, categorizes changes, and provides the ability to drilldown.

IBM InfoSphere Optim Query Workload Tuner for DB2 for z/OS delivers an easy to use interface which allows DBAs to access performance information at any time from any location. Also, its easy-to-use advisors enable DBAs of any skill level to monitor and tune queries to minimize performance problems helping you reduce downtime and associated costs that occur while responding to emergent problems. The net result makes your organization more resilient.
Before we get you started let’s review a few annotations that you will find in the lab instructions:

**Informational annotations provide additional information.**
Ignore them if you are in a hurry. You’ll be able to complete the lab even if you decide to skip them.

**Caution**
Pay attention to these details. They highlight common problems.

**Productivity**
Productivity annotations provide insights that can help you become more productive with Query Workload Tuner.

**If you have never used Query Tuner or Query Workload Tuner before, ignore this annotation.**
However, if you are already familiar with older releases of Optimization Expert pay attention to this annotation because there have been some changes.
Single Query Tuning

Optim Query Tuner for DB2 for z/OS cuts cost and improves performance by providing expert advice on writing high quality queries and improving database design. Its easy-to-use advisors can help developers to write more efficient SQL queries.

- Reduces costs and risks by enabling developers to tune SQL during development, while problems are still relatively inexpensive to fix and before they cause a costly outages or performance issues.
- Operates within a familiar Eclipse development environment and features seamless integration and natural launch points within Optim Development Studio.
- Accelerates query tuning analysis by providing expert advice and recommendations.
- Fosters collaboration among developers and DBAs; DBAs can extend query tuning advice to SQL Workloads (multiple queries) by using Optim Query Workload Tuner for DB2 for z/OS
- Operating systems supported: HP-UX, Linux, Windows family, z

Optim Query Tuner captures and analyzes queries from a variety of sources:

- Packages, SQL stored procedures, triggers, and user-defined functions (UDFs) - all from within the Data Source Explorer in the Eclipse environment
- The integrated query editor and routine editor within IBM Data Studio
- The SQL Outline and the Java editor within IBM Data Studio
- The DB2 catalog, for database packages and stored procedures

Optim Query Tuner helps reduce downstream risks and costs provides providing expert query tuning advice

Optim Query Tuner includes the following tools and advisors:

- **Access Plan Graph**: Provides a graphical view of the access plan chosen by the optimizer.
- **Query Annotation**: Formats the query with collapsible sections and highlighting to improve readability and navigation through complex queries. Provides catalog statistics and cost estimates for the tables and columns used in the query.
- **Query Advisor**: Provides query rewrite advice based on a set of best practice rules.
- **Access Path Advisor**: Provides advice on how to resolve access path issues that may result in poor performance.
- **Statistics Advisor**: Provides advice on missing, outdated and conflicting statistics and suggests additional statistics that should be gathered to improve individual query performance. Use the statistics advisor to tune the DB2 auto-stats profile.
- **Index Advisor**: Provides advice on new indexes that may improve performance for individual queries as well as the ability to define user-defined virtual indexes.
- **Query Report**: Simplifies collaboration identifying key statistics, access path information, and physical database design information related to a query.

This hands-on lab is using a sample project / connections that are provided with the Optim Query Tuner client. You can use the sample project to learn about how to tune queries with the query tuner client without a connection to an actual data server. Because the samples do not use a real data server connection, many functions are disabled, but the sample projects are pre-populated with example analysis and tuning results.
Because no data server connection is available for the sample projects, any actions that require an actual data server connection, including running the advisors and capturing a query environment, are disabled. Similarly, no items can be deleted from or added to the sample projects, and functional query groups cannot be added to a sample project. You can, however, create and delete sample projects.

Query Tuning Perspective
Lab 1  Cut Cost and Optimize Performance

Before you can use Data Studio and IBM Optim Query Workload Tuner to tune SQL queries and workloads, you have to connect to the DB2 for z/OS subsystem or data-sharing group member where those queries and workloads are run. Upon completion of this lab, you will have a basic understanding of how to use the Optim Query Workload Tuner 4.1 to tune a single query, and workload.

For this tutorial, you will examine the tuning process for the following sample query:

```
SELECT EMPNO, FIRSTNAME, LASTNAME, JOB, SALARY, BONUS, COMM, LOCATION, PROJNAME
FROM DSN8910.EMP A, DSN8910.DEPT B, DSN8910.EMPROJ C
WHERE A.WORKDEPT = B.DEPTNO
AND B.DEPTNO = C.DEPTNO
AND A.EMPNO IN (SELECT MGRNO FROM DSN8910.DEPT WHERE MGRNO IS NOT NULL ) ORDER BY 1,2,3
```

The purpose of the example query is to list employee information, including department and project information for employees who are managers. The query contains a subquery and references to three tables.
1.1 Getting Started

1. To launch IBM Query Tuner Client, click Start → All Programs → IBM Data Studio → Data Studio 4.1.0.0 Client

2. When prompted for a workspace, create a new workspace by typing: C:\Documents and Settings\Administrator\IBM\rational\workspace. Then click OK to continue.
1.2 Creating a sample project

You can use the Project Explorer to create a sample project and choose your sample query from a list in the customized New Project wizard. This method of creating a sample project is similar to how you might approach the tuning of a query that you have already identified in another context, especially if you can copy and paste the query text.

**What is a project?**

Projects are used to hold the queries and other metadata. Similar to directories in Windows, it is use as a directory to hold associated files to a specific project.

1. To create a new project, go to File -> New -> Query Tuner Sample Project
2. Next the Project wizard is going to guide you through the process of creating a new project. When prompted to enter a project name, type in QT_Sample_Project. Make sure that radio button is select for DB2 for z/OS sample project. In the Query tab, select the query that has the Statement ID 100 and click Finish.
3. You have now created a project called **QT_Sample_Project**. You will be brought to the Query Tuner Client Workflow Assistant page.

The new workflow assistant for query tuning guides you through capturing single SQL statements, creating query workloads, managing tasks for query workloads, performing query analyses, and tuning. The workflow assistant consists of sections that represent general steps in the process of tuning.

1. Status, for ensuring that your database or subsystem is configured correctly for query tuning
2. Capture, for capturing SQL statements from the supported sources
3. Manage, for creating and managing query workloads
4. Invoke, for running analysis tools, reports, and advisors on an SQL statement or a query workload
5. Review, for reviewing the results of the analyses, reading the reports, and reviewing and implementing the recommendations of advisors
6. Compare, for comparing the access plan graphs of SQL statements as you progressively tune them
1.3 Viewing Query Annotations

Optim Query Workload Tuner solutions make it easy to visualize queries. Depending on the source of the query, you might find one large block of text that is difficult to read. Query Tuner automatically formats the queries providing a good starting point for analysis. In the formatted query, each table reference, each column reference under the SELECT clause, and each predicate, is shown on its own line. Formatting alone can save hours of DBA time. Now it is easy to see how many and which tables are accessed in the query, what joins are performed, etc.

You can expand and collapse sections of complex queries, such as query blocks and subqueries, to see an overview of the query and drill into parts of the query in more detail. When you click any line in the formatted query, other lines of the query that contain column or table references from the same table are also highlighted. You can also customize the formatting ordering the predicates according to various criteria such as local predicates or join predicates, table references, and highest filter factor.

Query Tuner also adds annotations of the relevant statistics for predicates and table references such as cardinality and qualified rows. It automatically traces back through views handling the translations behind nested views. Plus it adds cost estimates and additional information such as data skew or default values.

This way, DBAs can accelerate analysis and reduce downtime for urgent situations. They can:

- Easily spot human errors like a missing predicate or missing comma
- Identify where filtering should occur based on filter factors.
- Determine where optimizer is using filtering
- Spot transformations which will occur

Query annotation allows you to see the formatted query for readability and annotations with key statistics for predicates and query blocks.

Annotation is only available in Optim Query Workload Tuner
1. In the Query Tuner editor, click the button **Select What To Run**. This will begin the process of query tuning and will generate an analysis result. Click Ok.
2. Click Ok

3. Once completed, you will see the tab **Summary - Analysis Result 1** results as shown below:
4. Now we are going to take a look at the query. In the Query Tuner Workflow Assistant, click **Open Formatted Query** view.

5. Next, click on the **Original** tab.
6. Scroll down to examine the formatted original query.

Formatted queries give you the ability to the following:

- Drill into parts of the query in more detail, such as referenced views and sub-queries, by expanding and collapsing sections of a complex SQL.
- Easily see how a specific table is accessed in the SQL. When you click any line in the formatted query, other lines of the query that contain column or table references from the same table are also highlighted.
- Customize the formatting ordering of the predicates according to various criteria such as local predicates or join predicates, and table references.
__7. Next click on the **Transformed** tab.

![Image](image1.png)

__8. When you compare the transformed query to the original query, you can see that the optimizer creates a virtual table to process the transformed correlated subquery.

__a. Collapse the **SELECT** clause by clicking on the node next to the **SELECT** statement.

![Image](image2.png)

The optimizer creates a virtual table to process the IN-list sub-query. In addition, the non-correlated sub-query has been transformed to a correlated sub-query.

This is an optimization that was introduced in DB2 for z/OS V9.1, which enables DB2 to optimize a query as a whole rather than as several independent query blocks. When a query is optimized as a whole, DB2 can consider the effect of one query block on another, and can consider reordering query blocks to determine an optimal query path.
b. Upon further examination, you can see that the EMPNO and DEPTNO tables are joined inside the subquery before the outer query is processed.

c. To quickly identify related parts of the query, click on the row that contains `SELECT DSN8910.DEPT.MRGNO`, and all the associated parts of the query that are related to the DEPT table are also highlighted.
d. Additional annotations with table and column statistics for the query such as skewed data warning are displayed in the third column for detailed analysis.

1.4 Viewing an Access Plan Graph

You can view the access plan that DB2 for z/OS uses to execute a query in graphical format. The diagram consists of nodes that represent tables, indexes, operations, and returned data. The nodes are arranged and connected by links that indicate the flow of the process. The graph is read left to right, bottom to top. And each node is annotated by statistics, estimated costs, selectivity information and so forth that are used to determine the access plan flow.

Intuitively, an access path is a procedural description of query execution that consists of three components:

• The join sequence of the tables
• The algorithm by which a table is scanned (access method)
• The algorithm by which a join operation is performed (join method)

Understanding the access plan is important for understanding and influencing performance, as well as for stabilizing performance. It is important to evaluate whether the optimizer has all the information it needs, whether it can apply filtering early to minimize rows retrieved, and whether the costs between alternative access plans are sufficiently distinct that they won’t be impacted by routine maintenance.
1. We are going to return back to the Query Tuner Workflow Assistant, click on **Open Access Plan Graph**, and take a look at the Access Plan Graph, as shown below.

**Caution**
If you accidentally closed the view, you can get back to it by clicking QT_SAMPLE_Project > Query Group 1 > Query Before Tuning > Initial Analysis then click “Select What to Run”
2. Scroll up and down, or left to right to view the operations in the Access Plan Graph.

3. The access plan graph shows that index scans are used to access the EMPLOYEE and DEPARTMENT tables in the virtual subquery and filter the rows before joining the tables with a nested-loop join. The results of the subquery are joined to the results of the EMPROJ table, which was accessed by a table scan, through another nested-loop join.
4. To see the description of the node, right click on **NLJOIN** node and select **Show Description**.
5. You can see the estimated performance cost of the query, by clicking on the join and looking below in the attributes selection. Expand the join node and the Join predicates, click on A.WORKDEPT=B.DEPTNO to see the cost estimation for that section of the query.

6. To return to the access plan diagram click the close button, located on the top right-hand corner.
7. Optional: Query Tuner provides the capability for you to search for nodes within the diagram. To search for nodes within the access diagram, on the left hand menu click Search for Node.

8. Now we are going to restore the view back. Click on Open Single Query Recommendations.
9. In the Query Tuner editor, look at the **Summary -Analysis Result 1** tab. You can use the advisor recommendations to tune the query. If you scroll through, you can see that there is one high recommendation from the statistics advisor.

**Access Plan Explorer**

This new tool displays access plans as tables or hierarchical trees.

**For access plans on DB2 for Linux, UNIX, and Windows:** Sort and arrange the columns in the table view to find the most costly operations in the access plan. Use the **Highlight Inflow** and **Highlight Outflow** buttons to see which operators feed information to a selected operator and which operator is sent information by the selected operator. You can also view the flows of information by using the tree view. Select an operator and view the information about it in the expandable tables in the lower portion of the Access Plan Explorer.

**For access plans on DB2 for z/OS:** Select a query block or mini-plan and view the information about it in the expandable tables in the lower portion of the Access Plan Explorer. For example, in looking at a mini-plan, you might find that the corresponding DB2 for z/OS table is accessed by a table-space scan. In the details for the mini-plan, you might find that the size of the table is large and the value of the filter factor for the predicate is also large. In this case, a large number of rows match the predicate and increase the cost of the mini-plan.

You can select a correlated subquery and click the **Highlight Relationship of Query Blocks** button to discover which mini-plan in the parent subquery the correlated subquery is bound to.
Lab 2  Tuning a single query

2.1  Viewing statistics advisor recommendations

DB2 uses a cost-based optimizer. Database statistics are the facts on which the optimizer makes decisions about access plans. The statistics typically include information about the number of rows in a table and the number of distinct values, the most frequent values, and the distribution of values in a column. The optimizer uses these statistics to compute the cost for each step in an access plan.

Consequently, if the statistics are inaccurate, outdated, or conflicting, then the optimizer will create inaccurate estimates for the costs of the steps in a query plan leading to poor performance.

RUNSTATS TABLE ALL INDEX ALL collects unnecessary statistics and misses key statistics: multicolumn and histogram stats are not collected. There are often statistical correlations between columns. For example, there is a strong correlation between "MODEL" and "MAKE" columns. Likewise, there is a strong correlation between "CITY" and "STATE" columns. Collecting statistics of individual columns [the defaults] may not be enough to provide the information required, so we need to collect column group statistics. Optim Query Tuner can recommend which statistics to collect based on the query and the workload. Interestingly, about half the access path issues reported to IBM DB2 Support is the result of incorrect statistics.

So the statistics advisor analyzes which objects are most interesting for the query or workload and provides advice on:

- **Missing statistics** – when statistics are missing, the optimizer assumes the default values to determine costs which could be completely inaccurate.

- **Conflicting statistics** – the optimizer may find that there has been an inconsistent statistics collection strategy which can result in conflicting statistics; for example, table and index statistics may be collected at different intervals.

- **Obsolete statistics** – the statistics could be very old and no longer represent the current state of the table.

The statistics advisor also assesses the relative importance of the advice it gives:

- **High:** Recommendations of this type indicate that important statistics are missing, obsolete, or that conflict exist among the statistics. The recommendation includes a RUNSTATS command that you can run to capture and repair the relevant statistics.

- **Maintenance:** Recommendations of this type have the lowest priority, and indicate that the catalog statistics are complete and accurate. The recommendation provides a RUNSTATS job that you might want to include periodically in your maintenance cycle to maintain the health of the relevant statistics.
1. We are going to return back to the QT_SAMPLE_Project Group Query Before Tuning, and take a look at the Advisor Recommendation Overview.

Caution
If you accidentally closed the view, you can get back to it by clicking QT_SAMPLE_Project > Query Group 1 > Query Before Tuning

2. In the Advisor Recommendation Overview, double click on the Statistics priority 1 row.

Summary - 1 Initial Analysis

<table>
<thead>
<tr>
<th>Recommendation</th>
<th>Number</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Query revision</td>
<td>2</td>
<td>Provide a join predicate based on the referential constraint between tables DSN20A and DSN20B.</td>
</tr>
<tr>
<td>Query revision</td>
<td>3</td>
<td>Provide a predicate on column WORKDEPT.</td>
</tr>
<tr>
<td>Access path</td>
<td>4</td>
<td>Avoid reading all index keys on an index scan (QBLOCKNO = 1, PLANNO = 1).</td>
</tr>
<tr>
<td>Index</td>
<td>5</td>
<td>Index recommendations found.</td>
</tr>
</tbody>
</table>
__3. In **Statistics** tab, located below the Advisor Recommendation Overview, you can see the RUNSTATS job recommended by the statistics advisor for collecting and repairing missing, obsolete, and conflicting statistics.

Run either of two recommended RUNSTATS commands. The Complete version of the RUNSTATS command collects a full set of statistics for the objects used in access plans for the statements that you are tuning. The Repair version fixes just the problems that the advisors identify.

**Recommendation 1:** Repair statistics problems for this query. Gather missing statistics. Recollect conflicting statistics and data correlation problems.
4. If you were tuning a real query, you can use the controls in the **Statistics** view to run the RUNSTATS job and collect or repair the statistics. In the **Statistics** view, you can get more up to date information on your statistics in the **Statistics Advisor report** view located in the Statistics Advisor Details view. Scroll down into the **Table, Index, Column, and column group details section**, notice on the right you see the Statistics Advisor Summary. This report is showing the last collection time, but notice that the **Statistics are missing**.

**After Statistics Advisor**

Represents the analysis results after running the RUNSTATS command that was recommended by statistics advisor in the previous analysis result and reanalyzing the query. The annotated query, access plan graph, and advisor recommendations reflect the changes to the performance of the query from the improved statistics. Again, you can click recommendations in the summary in the Query Tuner editor, to see hover help that highlights the next tuning step, and to see the details of the recommendation in the Advisor Details view.
5. For this tutorial, the results of running the recommended job and reanalyzing the query are shown in the 2 After Statistics Advisor result. In the Project Explorer, expand the QTSAMPLE_Project folder, expand the Query Group 1, expand Query Before Tuning and double click on 2 After Statistics Advisor.

6. You have now opened the Query Tuner editor for 2 After Statistics Advisor. Click on the Compare Access Plan Graphs from the Query Tuner Workflow assistance page.
7. Next we are going to compare the new access plan graph to the original access plan graph.

a. In the Workflow assistant, select the 1 Initial Analysis file.

b. Click on the 1 Initial Analysis on the left hand side.

c. Right click on the 2 After Statistics Advisor and select Compare button.
8. This will open the Access Plan Graphs (APG) comparison.
You can see that the access plan for the query has changed based on the updated statistics. The table scan operation on the EMPROJ table is now done earlier, and is moved to the left side of the outer nested loop join. This change improves the performance of the query by filtering more rows earlier in the processing of the query. Click on the **Review** tab.
10. In the **Open Single Query Recommendations**, examine the overview of advisor recommendations. You can see that you have resolved the statistics recommendation and that two recommendations from the query advisor are shown:

<table>
<thead>
<tr>
<th>Recommendation</th>
<th>Number</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>2 After Statistics Advisor</td>
<td>1</td>
<td>Provide a join predicate based on the referential constraint between tables DSN8910.EMP and DSN8910.DEPT.</td>
</tr>
<tr>
<td>2 After Statistics Advisor</td>
<td>2</td>
<td>Provide a predicate on column WORKDEPT.</td>
</tr>
<tr>
<td>2 After Statistics Advisor</td>
<td>3</td>
<td>Avoid reading all index keys on an index scan (CBOINDEX = 1, PLANNING = 1).</td>
</tr>
<tr>
<td>2 After Statistics Advisor</td>
<td>4</td>
<td>Index recommendations found.</td>
</tr>
</tbody>
</table>

11. The two recommendations are:

   a. Consider adding join predicates between the EMPNO column in table DSN8910.EMP and the MGRNO column in table DSN8910.DEPT. This will enable use of the referential constraints between table DSN8910.EMP and table DSN8910.DEPT to avoid a potentially costly Cartesian join.

   b. Consider adding the following predicate to the WORKDEPT column in table DSN8910.EMP. This is because WORKDEPT is defined as not allowing nulls and can help filter the table earlier and unlock more possible join sequences.
2.2 Viewing query advisor recommendations

Before access path selection, the DB2 optimizer transforms SQL statements into forms that are semantically identical, such as by using predicate pushdown or predicate transitive closure. As a result, it can improve the possible access paths. In contrast, Query Tuner helps the person who writes or tunes queries to identify errors and oversights by making suggestions that further constrain the query, increase index utilization, and reduce data reads. The query advisor looks for opportunities in a query to do the following.

- Minimize the number of index pages and data rows that have to be read. For example, you can minimize the rows read by having predicates in the query that can determine the needed rows from just the index alone.
- Minimize sort operations. For example, whether ORDER BY or GROUP BY clauses are needed in the query, or if they can be resolved via index access.

Specifically, the query advisor checks for the following.
- Missing join predicates, but only if a foreign key is defined.
- Stage 2 predicates that can improve performance if rewritten as Stage 1 or indexable.
- Stage 1 predicates that can improve performance if rewritten as indexable.
- Additional local predicates that are not automatically provided by DB2 that can provide predicate transitive closure.
- Predicates that are pushed down to a nested table expression or materialized view without changing the result, and not already done automatically by DB2.
- Additional predicates added to a complex WHERE clause, containing OR, AND, and () parentheses. This might improve performance without changing the result.
- The use of SELECT * which can be replaced with specific column list.

The query advisor highlights potentially problematic parts of an SQL statement and explains why you should consider rewriting that part of the query.

1. To view the query advisor recommendations, in the Advisor Recommendation Overview, select double click the first Query revision. Go to the Query Revision Detail, located at the bottom of the menu.

2. Scroll down to examine the Advisor Detail view and look at the Query Revision Details.
3. Notice that the relevant part of the SQL text is highlighted in the SQL Text panel below for the selected recommendation.

4. Review the highlighted SQL text, description, and explanation for each recommendation.
Similar to the Access Path Advisor, you can choose to hide some recommendations which are not suitable for your specific query or environment.

5. The next step in tuning the query is to rewrite the query based on the recommendations of the query advisor. The query advisor does not provide the rewritten query text automatically. Normally, you would create a new query in the same project and enter your rewritten query text there. Doing this enables you to backtrack to the original query, if necessary. For this tutorial, the rewritten query text is provided for you in the Project Explorer (under Rewritten Query).

Rewritten Query
Represents the text of the query after it was modified it according to recommendations from the query advisor. Notice that it is a separate query in the query group. Creating a new query object in this manner is a good practice to save your original query text in case you want to revisit the original performance or backtrack to the original query. Not all of the sample queries include a rewritten query.

6. In the Project Explorer, double click Rewritten Query.

7. You can see that the join, `a.empno = b.mgnrno`, and the predicate, `a.workdept is not null`, have been added to the query. The results of analyzing the revised query are shown in the After Query Advisor result.

After Query Advisor
Represents the analysis result after the query text was revised and reanalyzed.
8. In the Query Tuner editor, click the button **Run Default Advisors and Tools**

9. Now we are going to compare the new access plan graph to the original access plan graph. Notice that the table scan of the EMPROJ table has moved to the left side of the nested loop join. You can also see the change in estimated cost for the rewritten query:

   a. Click on the **Compare Access Plan Graphs** from the Query Tuner Workflow assistance page.
__b. Click on the **Initial Analysis** on the left hand side. Click on Rewritten Query **1 After Query Advisor** and select **Compare**.

__c. On the Access Plan Graph, right click on **QUERY** from **QT_SAMPLE_Project / Query Group 1 / Query Before Tuning / 1 Initial Analysis** and select **Show Description**.

__d. This will pop up a **Selected Node Descriptor: Query** window. Ignore this window for now.
__e. Go into other diagram, QT_SAMPLE_Project / Query Group 1/ Rewritten Query/ 1 After Query Advisor right click on they QUERY node and select Show Description.
_f. This will bring another Selected Node Descriptor: Query window. You can compare the estimated CPU costs in each access plan graph to see the performance improvement.

__10. Click on button to close the two Selected Node Descriptor: QUERY windows.
__11. In the Project Explorer, double click on 1 After Query Advisor.

__12. Examine the overview of advisor recommendations. The query advisor recommendations have been resolved and are no longer shown. Now that the statistics and query advisor recommendations have been resolved, you can look at the access path advisor and index advisor recommendations.
13. In the overview of advisor recommendations, select the access path advisor recommendation. The recommendation states: **The DSN8910.EMP table is accessed by a non-matching index scan.** Because the problem is related to the indexes, the index advisor recommendations might help to resolve the problem.

<table>
<thead>
<tr>
<th>Recommendation</th>
<th>Number</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Access path</td>
<td>1</td>
<td>Avoid reading all index keys on an index scan (QBLOCKNO = 1, PLANNO = 1).</td>
</tr>
<tr>
<td>Index</td>
<td>2</td>
<td>Index recommendations found.</td>
</tr>
</tbody>
</table>
2.3 Viewing index advisor recommendations

Query Tuner also provides index advice. It analyzes the query and recommends additional indexes that would benefit the query access. Index advisor might recommend indexes for the following reasons.

- Foreign keys that do not have indexes defined.
- Indexes that will provide index filtering and/or screening for the SQL statement.
- Indexes that will provide index-only access for the SQL statement.
- Indexes that can help to avoid sorts.

The index advisor recommends indexes that you can create, modify, or delete to improve the performance of an SQL statement, and allows you to run the appropriate CREATE, ALTER, or DROP INDEX statements.

1. We will now take a look at the 1 After Query Advisor document in the Query Tuner editor. The file can be found in the Project Explorer in the QT_SAMPLE_Project folder → Query Group 1 → Rewritten Query → 1 After Query Advisor.

   ![Image of Task Launcher with 1 After Query Advisor selected]

2. To view the index advisor recommendations, in the Query Tuner editor, look at the Advisor Recommendation Overview and double click on the Index Advisor.

   ![Image of Advisor Recommendation Overview with Indexes highlighted]
3. After selecting **Index Advisor** look at the **Advisor Detail** located below. You can see the following:
   1) the recommended index
   2) the estimated percentage of performance improvement you might see after creating the index, and
   3) the list of existing indexes.

4. In the Advisor details view, examine the DDL script recommended by the index advisor by clicking **Show DDL**.
5. When tuning a real query, you can use the controls on the advisor details page to create the recommended indexes, and add your own indexes to the script. Click OK when you are done looking at the DDL.

You can also customize recommendations and the DDL creation. For example you can specify a maximum for the number of columns that can be part of an index key and change the Creator ID that is used when index DDL is generated. You can also specify the FREEPAGE, PCTFREE, and CLUSTERRATIO values to be assumed for new indexes, and whether or not index leaf page size can exceed 4 KB.
6. In the **Project Explorer**, navigate to **QT_SAMPLE_Project → Query Group 1 → Rewritten Query** and double click on **After Index Advisor**.

7. The results of creating the indexes are shown in the Query Tuner editor. Take a look at the **Access Plan Graph**. You can see that the newly created indexes are now used to access the tables.
After Index Advisor

Represents the analysis result after running an index advisor recommendation and reanalyzing the query. The annotated query text, access plan graph, and advisor recommendations reflect changes to the performance of the query from the improved indexes. The changes that were made to the query text for the query advisor are also reflected in the analysis result.

8. Scroll down and look at the **Advisor Recommendation Overview**. It is now empty because all advisor recommendations have been resolved.
Lab 3  Viewing Query Reports

3.1 Generating Query Reports
Query reports allow you to view detailed information about an SQL statement.

1. In the Project Explorer, navigate to QT_SAMPLE_Project → Query Group 1 → Rewritten Query and double click on **After Query Advisor**.

2. This will launch the Query Tuner editor, in the **Advisor Recommendation Overview** click on **Open Access Path Reports**
__3. The **Generate Access Plan Reports** pop-up will guide you through the process of creating reports. Verify that **Table report**, **Index report** and **Predicate report** options are selected and make sure that **Show in HTML format** is checked. Click **OK**.

__4. Three HTML files are generated for this example, the **Table Report**, **Predicate Report** and **Index Report**.

   **Table Report**

   (2012-01-05 17:29:46.078)
5. When done looking at the reports click on the button to close the reports.

Productivity
A record of the report files can also be found in the Project Explorer under QT_SAMPLE_Project → Query Group 1 → Rewritten Query → After Query Advisor → Query Reports.

Query reports allow you to view detailed information about an SQL statement.
Workload Tuning

This section uses the QTTUTORIAL tutorial database. You can see the sample workloads by selecting TUTORIAL_ZOS in the Subsystems list in the Workload list view. Three sample workloads are provided:

- **WorkloadWithTypicalStats** - Contains a set of typical queries before any tuning, which might not have updated statistics. You can invoke all the workload advisors to tune these queries as a group.

- **WorkloadTunedWithStatsAdvisor** - Contains the same set of queries with the updated catalog statistics. You can invoke workload advisors to get workload recommendations for creating indexes and rewriting queries.

- **WorkloadTunedWithIndexAdvisor** - Contains the same set of tuned queries after creating the indexes recommended by the workload index advisor. You can see the significant performance improvement for the entire workload after using the new indexes to process the same workload.

You can get recommendations from the Workload Statistics Advisor, Workload Index Advisor, and Workload Query Advisor.

**Workload Statistics Advisor**

The Workload Statistics Advisor considers a set of SQL statements in a query workload and looks for missing, incomplete, obsolete, and conflicting statistics that might lead to suboptimal performance for the SQL statements in the workload. The Workload Statistics Advisor provides a consolidated set of statistics recommendations that apply to the entire workload, and provides RUNSTATS jobs that you can run to improve the performance of the workload as a whole.

After collecting accurate and up-to-date statistics for the workload, you can use the Workload Index Advisor to analyze the indexes for the workload.

**Workload Index Advisor**

The Workload Index Advisor considers a set of SQL statements in a query workload, and looks for ways to improve the performance of the workload, or reduce the amount of disk space used by indexes on the data server. The Workload Index Advisor recommendations include DDL scripts for creating indexes to improve the performance of the workload as a whole, or for dropping indexes to recover disk space, depending on the scenario that you specify.

**Workload Query Advisor**

The Workload Query Advisor compares SQL statements with a set of best practices and recommends ways that you might rewrite SQL statements in the workload to improve the performance of the workload as a whole.
Lab 4  Sample Workloads

4.1  Viewing Sample Workloads

1. To open the Workload List view, go to File → New → Query Tuner Sample Project…

2. Next the Project wizard is going to guide you through the process of creating a new workload project. When prompted to enter a project name, type in QT_Sample_Project2. Make sure that radio button is select for DB2 for z/OS sample project. In the Workload tab, select the query that has the WorkloadWithTypicalStats and click Finish.
3. You now have created a project called **WorkloadWithTypicalStats** double click the tab then you can open the Workload Tuner Editor to tune your workload.

4. You can use the Query Tuner client to capture from DB2 for z/OS a set of queries that are related in some way. You can capture the query workloads from various sources, run advisors to tune the performance of the queries in the workload as a group, and single out individual queries for detailed analysis and fine-tuning.
Lab 5  Tuning a Workload

IBM Optim Query Workload Tuner (OQWT) allows you to capture a group of SQL statements that run on a DB2 for z/OS subsystem in a workload and get recommendations for improving the SQL performance as a group.

One of the advantages of Query Workload Tuner is allowing you to collect and operate on collections of SQL statements, not only single SQL queries that are analyzed one at a time.

Advanced query tuning features are enabled in the query tuner client. In addition, the following activities related to workloads are also provided:

- Creating monitor profiles to capture information about the performance queries and query workloads that run on the data server
- Defining and capturing query workloads
- Getting expert advice, considering an entire workload, in the following areas:
  - Statistics
  - Index design
  - Query design
- Capturing and sending detailed information about workload environments to different data servers or to IBM software support.
- Creating detailed reports about tables that are referenced by queries in the workloads.
5.1 Viewing Workloads

1. We are going to clean up the workspace by closing opened editors and resetting views.

   a. Right click on QT_SAMPLE_Project_Query Group 1_Rewritten Query tab and select Close All

   b. We are also going to reset the view. To reset the view click Window the click Reset Perspective...

   c. When prompted to reset the current Data perspective to its defaults click OK.
2. From the previous lab, we have created a workload project **WorkloadWithTypicalStats**. In the **Project Explorer** navigate to the **WorkloadWithTypicalStats** project, expand **Workload Group 1** then double click on **WorkloadWithTypicalStats [TUTORIAL_ZOS – WorkloadWithTypicalStats]**.

3. This will open the Workload Tuning Editor. Notice if you right click on any query you will see the single query tuning tools available from the **Workload Statement** view. This means you can select any query in the workload to perform single query tuning.
4. To view the statement in a detached window, double click on any statement in the workload.

![View SQL Statement](image)

5. After you are done looking at the SQL statement, click Close.
5.2 Viewing Workload with Statistics Advisors Recommendations

The workload statistics advisor recommends statistics that you can capture to improve the performance of an entire SQL workload, and generates RUNSTATS control statements that you can run immediately from QWT or save to run at a later time.

__1. To view the workload advisors, in the Workload Statements section click on Run Workload Advisors, next click on Run Default Advisors.__

__2. Click OK__
3. After all the advisors have completed, you will notice that the Statistics Advisor requires your attention. Click on Statistics Advisor to get more information.

4. Click on the Workload Statistics Advisor tab
5. This will bring you to the Statistics Advisor Details page where you can take a more detailed look at each of the indexes.
5.3 Viewing Workload with Query Advisors Recommendations

The workload query advisor warns you about problematic statements that might cause poor performance from an SQL workload. You can view statements according to the severity of the warnings.

1. Click on the Workload Query Advisor tab

2. The Workload Query Advisor Recommendations Summary will open to display the total number of SQL statements that were analyzed, indicates how many warnings were generated. You can set viewing criteria by the degree of warning severity.
3. The workload query advisor does not highlight any problems with the statements of the workloads that are provided in the TUTORIAL_ZOS <enabled> database. However, you can view the statement runtime statistics by clicking the View Queries button.

4. This will open the Workload Query Advisor Details, which show the runtime statistics in the workload. If you see a problematic query, you can right click to perform tuning for a single query.

5. Double click on the first query, statement 3530.
6. A **Query** tab with the statement ID will be added next to the Query List tab. You can review the recommendations for Query 3530. Here see the details of a query advisor recommendation along with the rationale for the recommendation. The UI automatically highlights the section of the query related to the recommendation.

7. To review other queries in the list, click on the **Query List** tab.
5.4 Viewing Workload with Index Advisors Recommendations

The workload index advisor recommends indexes that you might create, modify, or remove to improve the performance of an SQL workload. It also generates the CREATE, ALTER, and DROP INDEX statements that you can run them immediately in QWT or save for later.

1. Create a new project, go to File → New → Query Tuner Sample Project ...

2. Next the Project wizard is going to guide you through the process of creating a new project. When prompted to enter a project name, type in QT_Sample_Project3. Make sure that radio button is select for DB2 for z/OS sample project. In the Workload tab, select the query that has the WorkloadTunedWithStatsAdvisor and click Finish.
3. In the Workload Tuning Editor for **WorkloadWithStatsAdvisor**, **Run Default Advisors** to tune the workload.

4. In the Advisor window, click on the **Details…** recommendation for **Index Advisor**.
5. The Workload Index Advisor Recommendations tab shows the recommended indexes for the entire workload. It also lists the existing indexes. You can set the recommendation parameters to limit the amount of disk space to allocate new indexes, and the estimated number of indexes allowed per table.
Review Workload Advisor Recommendations

This page shows the recommendations from the advisors that you ran.

Workload name: WorkloadTunedWithStatsAdvisor
Number of statements: 22
Workload owner: BSQSC12
Status: contains a set of SQL statements with the updated catalog statistics. You can invoke workload advisors to get further

Estimated performance improvement: 86.52%
Disk space required (DASD space): 675.08 MB
Total I/O Cost: 0.0 MS

Recommendations:

Create Indexes
- CUSTOMER
  - Index: Create CLSTOMEN
  - Index: Create C_MCTSEGMENT
- ORDER
  - Index: Create ORDER_V
  - Index: Create O_CUSTKEY
- LINEITEM
  - Index: Create LINEITEM

Existing Indexes

<table>
<thead>
<tr>
<th>Index</th>
<th>Table</th>
<th>Creator</th>
<th>Index Columns</th>
<th>Used After</th>
<th>Used Before</th>
<th>Foreign Key Index</th>
</tr>
</thead>
<tbody>
<tr>
<td>PKS_SUP</td>
<td>SUPPER</td>
<td>SYSDM</td>
<td>S_SUPKEY, S_NATIONKEY</td>
<td>Y</td>
<td>Y</td>
<td>N</td>
</tr>
<tr>
<td>PKS_PART</td>
<td>PARTSUPP</td>
<td>SYSDM</td>
<td>PS_PARTKEY, PS_SUPKEY</td>
<td>Y</td>
<td>Y</td>
<td>N</td>
</tr>
<tr>
<td>PKS_BPRTPN</td>
<td>PART</td>
<td>SYSDM</td>
<td>P_PARTKEY, P_TYPE, P_N...</td>
<td>N</td>
<td>Y</td>
<td>Y</td>
</tr>
<tr>
<td>PKS_CSTNKS</td>
<td>CUSTOMER</td>
<td>SYSDM</td>
<td>C_CUSTKEY, C_NATIONKEY</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
</tr>
<tr>
<td>PKS_ORDERS</td>
<td>ORDER</td>
<td>SYSDM</td>
<td>O_ORDERKEY, O_ORDERDATE</td>
<td>Y</td>
<td>Y</td>
<td>N</td>
</tr>
<tr>
<td>PKS_NATION</td>
<td>NATION</td>
<td>SYSDM</td>
<td>N_NATIONKEY, N_NAME</td>
<td>Y</td>
<td>Y</td>
<td>N</td>
</tr>
<tr>
<td>PKS_LIG</td>
<td>LINEITEM</td>
<td>SYSDM</td>
<td>L_ORDERKEY, L_SHIPDATE</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
</tr>
</tbody>
</table>
6. You can preview the DDL to create the recommended index. Click **Show DDL...** to preview the generated DDL.
__7. Examine the DDL. If you click **Save**, you can run the DDL at another time for this lab we will not save the DDL at this time. Click **OK** to continue.

__8. Click **Change Constraints** … on the Recommendation tab to help you decide whether to change settings to produce a different set of recommendations.
Caution
Since this is a tutorial you will not be able to run a what-if analysis

9. If you have a live connection to the target DB2, you can perform workload What-If analysis using the recommended indexes and recommendation parameters to limit the resources used.

10. Click Cancel.
Lab 6 Viewing Query Environment Capture

6.1 Viewing Query environment capture

When you are working with IBM Support to resolve a problem with tuning an SQL statement, or when you want to recreate the environment of an SQL statement on a different DB2 for z/OS subsystem, you can capture the statement's environment in a set of files. You can then upload the files to IBM Support or to the other subsystem.

This feature is available only when you are connected to a DB2 for z/OS subsystem It is not available when you are connected to a DB2 for Linux, UNIX, and Windows database.

Query environment capture allows you to provide IBM support with information required to re-create a problem. It generates DDL, Catalog statistics, Zparms (if DSNWZP stored procedure is available) and Explain tables info (PLAN_TABLE, DSN_DETCOST_TABLE, DSN_PREDICAT_TABLE) (if SQL was used as input).
1. In the Query Workflow Assistant click on **Capture Workload Environment**.
2. In the PMR field, “1111, 1111, 1111”, “910” for the version, and click the **Generate Report** button.

3. Click **Upload Files**
Caution
Because no data server connection is available for the sample projects, any actions that require an actual data server connection, including capturing a query environment, are disabled.

4. In upload file list area, you can find the generated service files. If this were a real environment, you could directly click the Upload Files button to upload these files to specified FTP server. Also you could zip and send these files to another subsystem for problem re-creation.

Caution
This use to demonstrate how to send files for support. Since this a tutorial you can not click Finish.

5. Since this is for demonstration purposes, click Cancel.
6.2 Optional – Reference ONLY : Define a Workload Using the Workload Wizard

As mentioned previously, you can view and analyze a collection of statements as a group called a workload. You can collect workload from various sources including statement cache, DB2 catalog, QMF, a file, and more.

---

**Caution**
The tutorial database has limited functionality and simulates what you might see in a production DB2 for z/OS system. So you will not be able to create a Workload using the Workload Wizard.

---

**You cannot save…**
The workload into the tutorial database because there is no live-connection for the tutorial database. Subsequent workload exercises will use the predefined workloads stored in the tutorial database.

---

6.2.1 Workload Source: Dynamic statement cache

The dynamic statement cache captures all dynamic SQL, and if IFCID 0318 is enabled, performance data as well. Open the Manage Tab

1. Open the Manage Tab,

2. This will open a view called **Workload List** located in the bottom half of the Query Tuner client. Double click on **Workload List** tab to expand the view.
3. In the Workload List view, make sure that the subsystem TUTORIAL_ZOS <enabled> is selected. In the Workloads List section, click on New Workload, which causes a drop-down menu to appear. Select Create… from that menu. In the Manage section of the editor, click the Create Workload button to open the Create Workload wizard.
4. In the Workload Wizard, take the default Workload name: **Workload_1** and click **Next >**.
5. For the source criteria, take the default Source name: **Source_0**. In the drop-down menu for Source type: select **Statement cache** and click **Next >**.

The tutorial database simulates the ability to capture workload queries; however, **Statement Cache** is the only source type supported by the tutorial database. When connected to a live DB2 subsystem, you can choose from the following source types from **View Queries**:

- Statement cache
- Catalog plan and package
- QMF
- Monitor
- File
- SQL procedure

Additional source types supported in **Project → Query**:

- Category
- Function table
- Plan Table
- Statement table
- User-entered text
6. Filter options are provided in the wizard; we are not going to provide a filter so click **Next >**.
7. Specify the **Capture profile** as **immediately**. You can also specify the capture to be at the client or at the server.
8. You can specify members in a data sharing group, or you can configure data sharing group to be included in the Group Members. To configure data sharing group, click on **Configure Group**.
9. When **Configure Group** wizard pops up, click on **New Group** and type in **DemoGroup** for the Group alias; and for the Group name:, type in **OQTGroup1**. Then click **Add Member**.

**Caution**
The following is for demonstration purposes; you will **not** be able to save.
Conclusion

Congratulations! You’ve completed the introductory IBM Optim Query Workload Tuner 4.14.1 lab which walked you through the general process for tuning a single query, tuning a query workload, managing the monitor profile, and configuring the product.

Visit us on the web for more in-depth articles and tutorials at http://www.ibm.com/developerworks/spaces/optim