



Make SQL Fly with the Speedgain for DB2 SQL Workbench

Prepared by:
Philip K. Gunning



Most industry experts agree that Suboptimal SQL accounts for at least 80% of database performance problems. Additionally, in many cases developers and DBAs lack the tools necessary to conduct SQL tuning in a proactive manner. Developers need an easy to use Graphical User Interface to analyze SQL and access paths, and to receive expert recommendations on how to improve the SQL via SQL rewrite or index redesign. Furthermore, many tools today do not contain an integrated environment and require users to go outside the tool to integrate with the Design Advisor.

The Speedgain SQL workbench solves many of these problems. The SQL workbench contains an integrated graphical explain, expert SQL tuning rules, the ability to conduct multiple "what-if" SQL drills, and seamless integration with Design Advisor. It can be launched from Speedgain for DB2 monitor or SQL statements can be loaded from an external file. Refer to Figure 1-1 for an example of launching SQL Workbench from Speedgain for DB2.

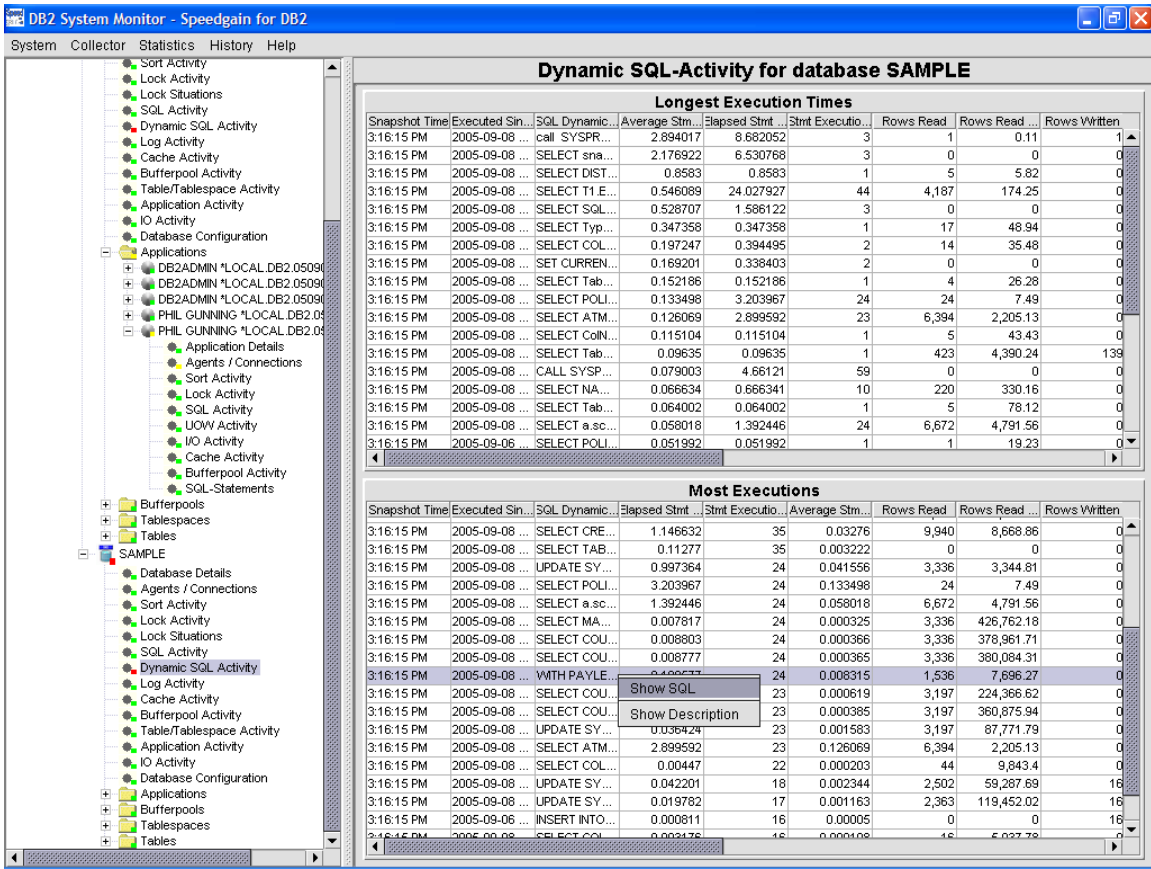


Figure 1-1. Launching SQL Workbench

As shown in Figure 1-1, SQL Workbench can be launched from the Speedgain Dynamic SQL Activity dialogue. It should be noted that Speedgain for DB2 captures dynamic SQL via classic snapshots, and stores them in the Speedgain Performance database and makes them available to the SQL Workbench.

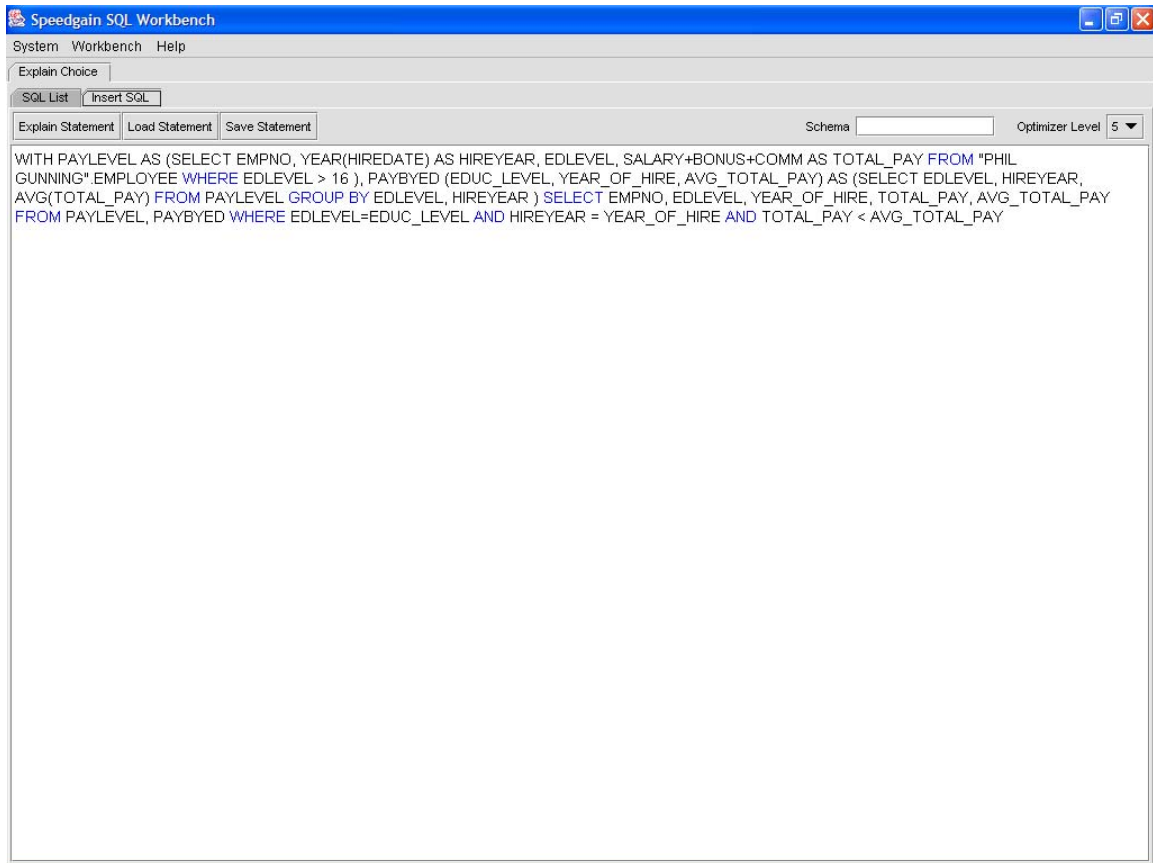


Figure 1-2. Speedgain SQL Workbench

The initial entry into SQL Workbench is as shown in Figure 1-2. From here you can launch a graphical explain or import SQL or a workload from an external file as shown in Figure 1-3.

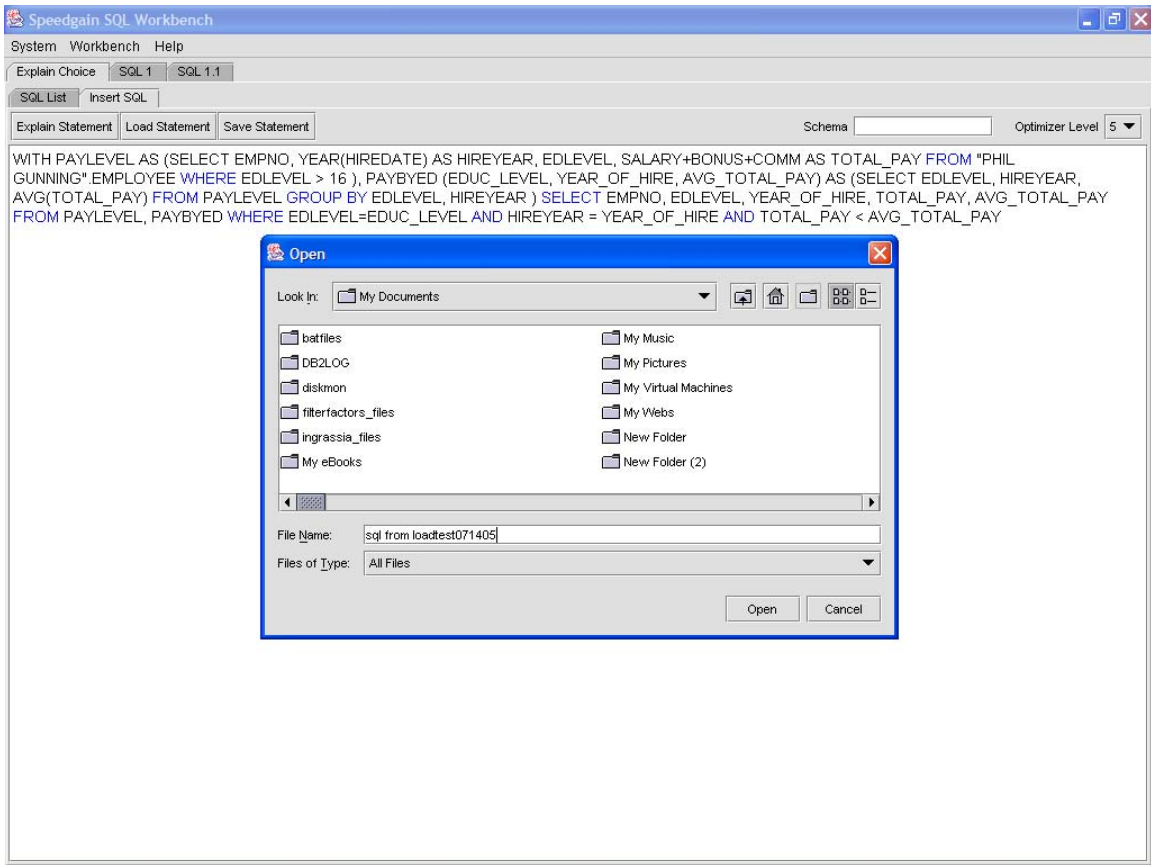


Figure 1-3. SQL Workbench in Stand-alone mode

This makes SQL Workbench flexible and easy to use for developers and DBAs; especially when developing SQL or performing multiple “what-if” drills. To review all statements associated with a particular time period, select the “**SQL List**” tab and you are presented with a list of SQL statements based on the statement and period selected as shown in Figure 2-1.

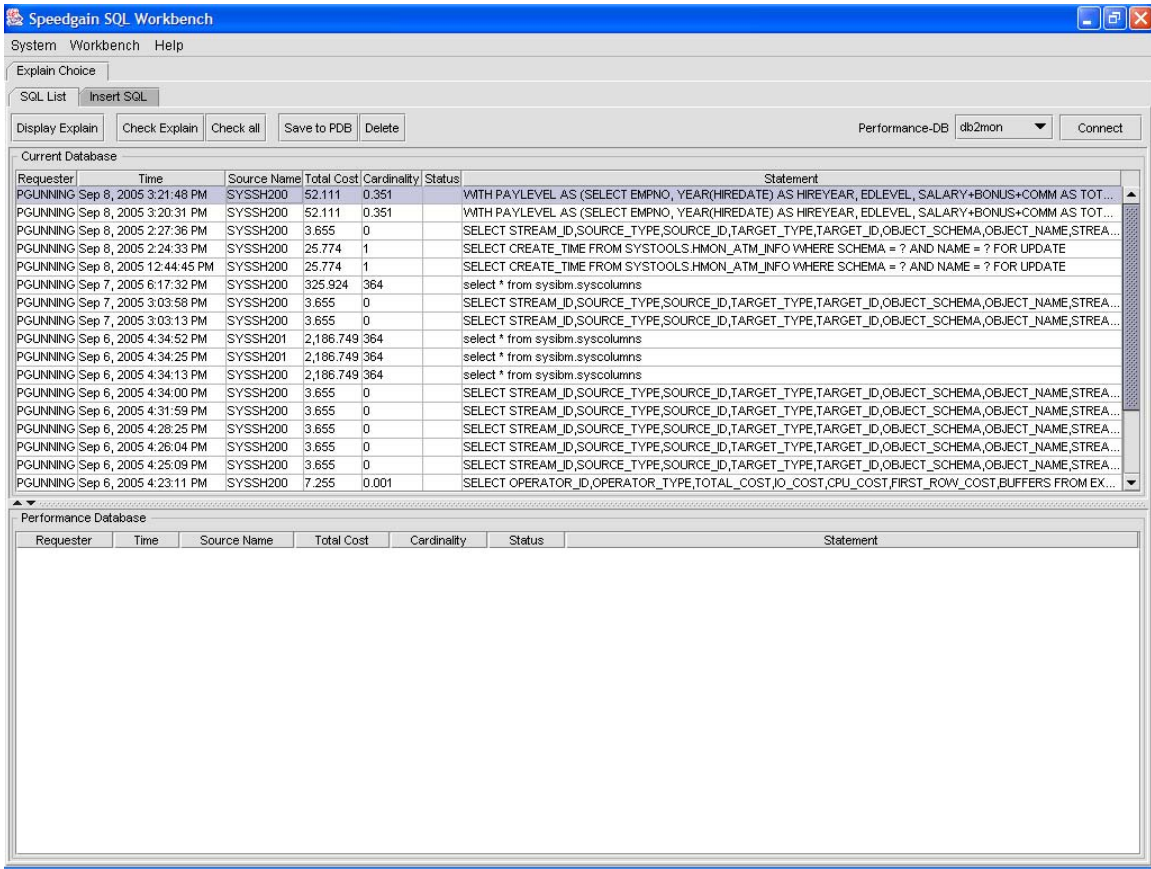


Figure 2-1. Selecting SQL to tune from a list

To conduct SQL tuning, simply select the high-cost SQL to be tuned in the lower half of the panel (highlighted in gray), and select the **“Display Explain”** tab and a graphical explain and associated object statistics are shown as demonstrated in Figure 2-2.

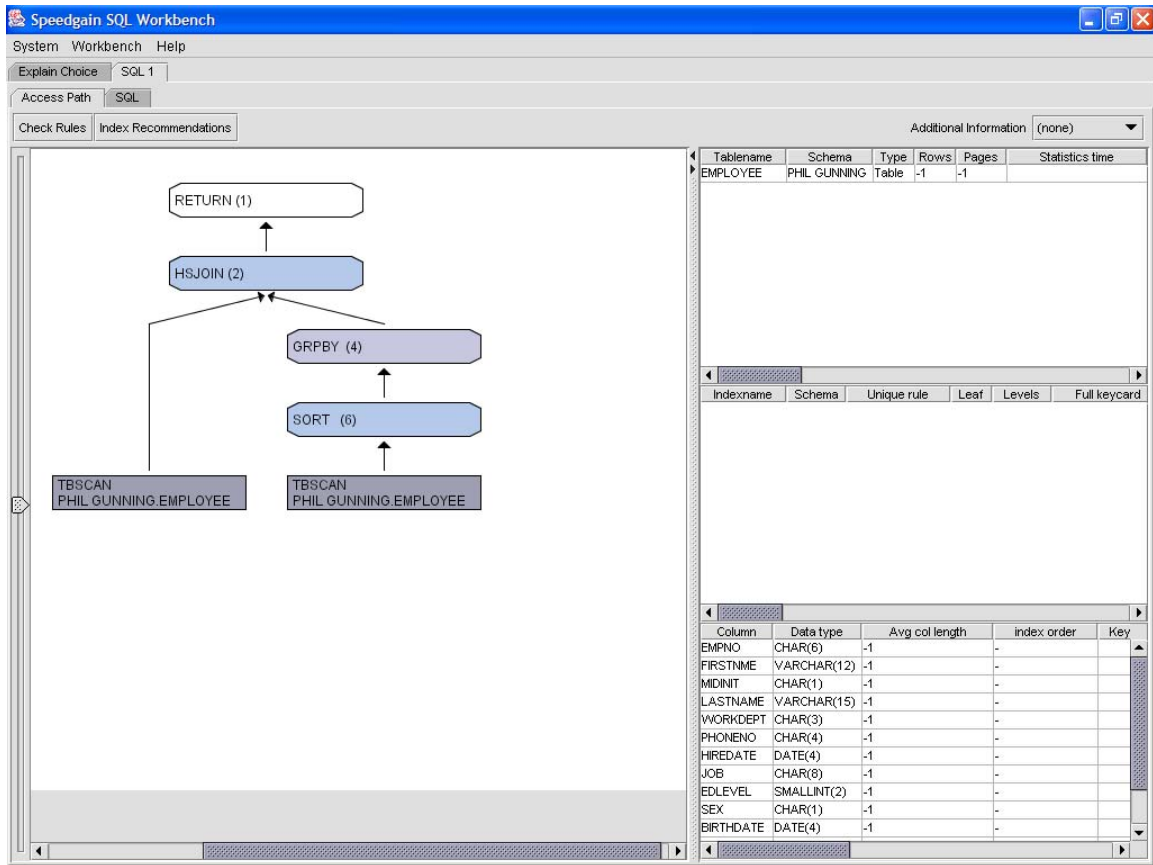


Figure 2-2. SQL Workbench Graphical Explain

You are then presented with a tailored graphical explain which is displayed in the left pane and complete object statistics are displayed in the right pane. The graphical explain is tailored such that duplicate or extraneous explain information has been eliminated. This enables developers and DBAs to focus-in on only the essential explain information. To review the statistics and cost information associated for any step in the explain, double-click on the operator of interest and object information is shown as demonstrated in Figure 2-3.

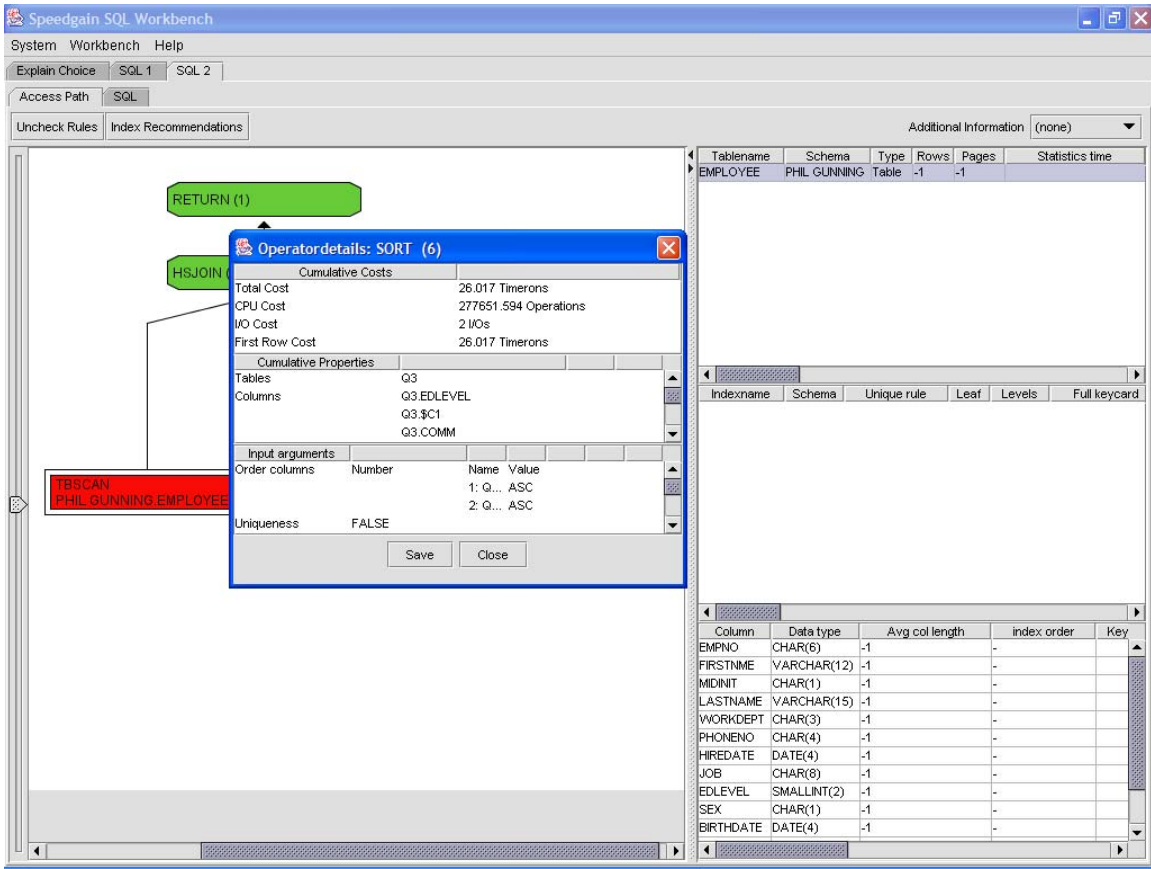


Figure 2-3. Object detailed cost and predicate information

After reviewing object cost and statistics information, the next step is to apply the expert rules by selecting the **"Check Rules"** button. You are then presented with a graphical explain with rules checked and violated rules displayed as shown in Figure 3-1.

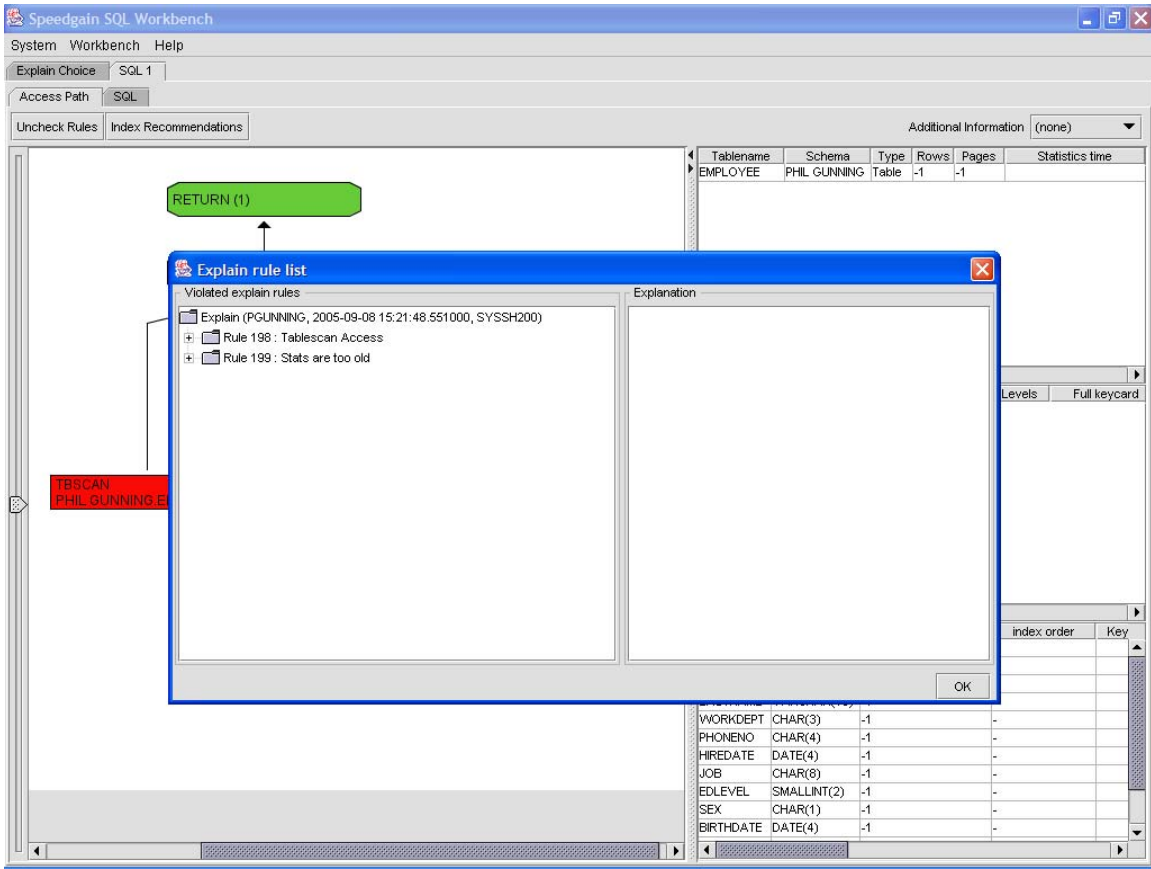


Figure 3-1. Graphical Explain with Rules Checked

As shown in Figure 3-1, a list of violated explain rules are shown in the pop-up box. In this example the rules indicated that a table scan is involved and that the statistics are too old. After closing the “**Explain rule list**” dialogue, you are presented with a color-coded graphical explain with yellow and/or red colors pinpointing the hotspots.

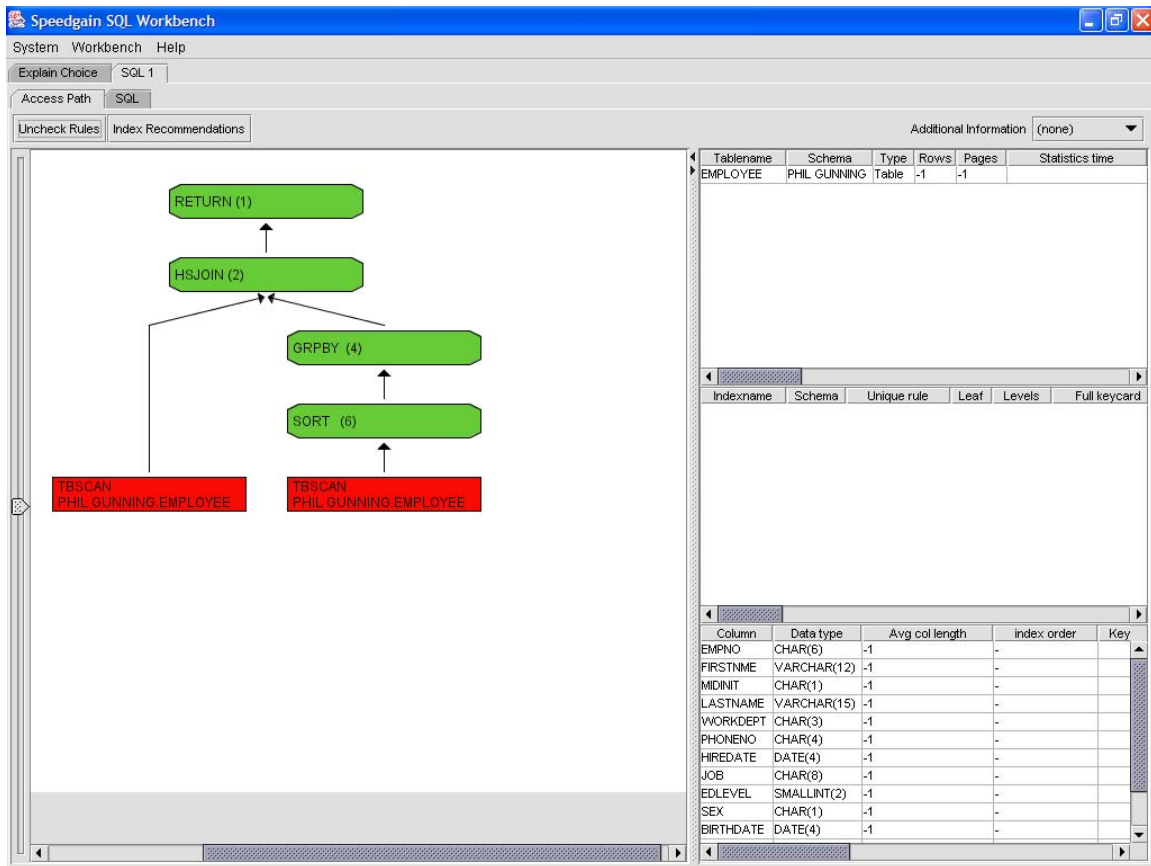


Figure 3-2. Graphical Explain with Problems Highlighted

As shown in Figure 3-2, the portions of the access path that violate expert SQL rules are highlighted in yellow or red depending on the severity. A developer or DBA can manually review the graphical explain and object statistics, but with SQL Workbench all you really have to do is select the **“Check Rules”** tab and the SQL Workbench engine automatically analyzes the SQL, applies the rules, creates another SQL trial (SQL Tab 1.1), and produces a rule violation pop-up as previously demonstrated, if expert SQL rules have been violated. Since the SQL rules are stored in the Speedgain Performance Database (PDB), DBAs and developers can write their own SQL rules and store them in the PDB. The SQL Workload analysis engine will then consider them when analyzing access paths, in addition to the default rules already provided.

After reviewing the rules report, you can conduct what-if drills by creating another SQL trial and making changes to the alternative by changing the SQL as indicated by the expert rules. In a future release this will be fully automated and the rules will be applied and SQL rewritten based on the rules. After changing the SQL based on the rule recommendations, you can then re-explain the modified SQL and review the explain, and even apply the rules again to verify

that you have made the correct changes and that the access path is what you want.

Sometimes developers and DBAs have no ability to change the SQL. With the advent and popularity of third party vendor packages such as SAP, Siebel, PeopleSoft, and others, developers and DBAs no longer have the option to simply rewrite the SQL. The only other solution in these cases, is to conduct what I call "index redesign". Index redesign is best conducted using the Design Advisor. SQL Workbench integrates seamlessly with the Design Advisor. In other words, you don't have to leave SQL Workbench to interface with it. To evaluate the adequacy of existing indexes and to obtain recommendations for new or modifications to indexes, just select the "**Index Recommendations**" button and the SQL Workbench analysis engine will interact with the Design Advisor and return recommended indexes if found. Recommendations are displayed in the statistics pane highlighted in red, as shown in Figure 4-1.

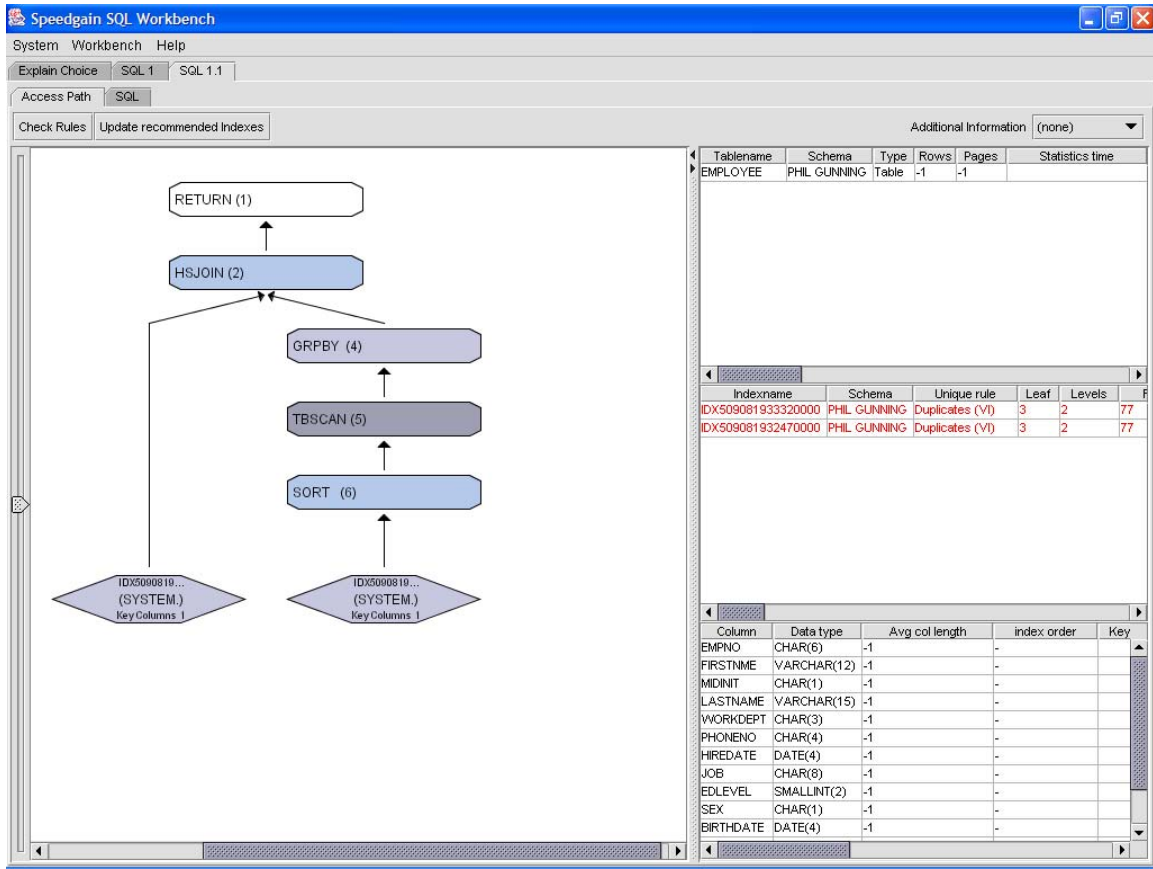


Figure 4-1. Design Advisor Index Recommendations

Note that since the integration has been already done with the Design Advisor in this implementation of SQL Workbench, future releases will display additional details on recommendations from Design Advisor, such as MDC index recommendations and MQT recommendations.

Recommended indexes are highlighted in RED and displayed in the statistics pane as shown in Figure 4-1. The recommended indexes are automatically created as virtual indexes when the “**Index Recommendations**” tab is selected. If necessary, several SQL trials can be created and different types of SQL modifications and optimization classes can be tried, and the process iterated until the best solution is found. This capability goes a long way in helping DBAs and developers to quickly and confidently solve suboptimal SQL problems. All without leaving the tool!

NOTE: Caution must be taken when evaluating single SQL statements with Design Advisor as the impact of any index recommendations on other SQL that may access the tables involved is unknown.

SQL Workbench ameliorates these concerns by providing the capability to load a file containing a workload of SQL statements for analysis by Design Advisor, and this is the recommended approach. This can be done by selecting "**Load Statement**" button from the menu and loading the SQL file. This way you can be assured of a complete index solution for the workload and tables involved. Additionally, in a future enhancement to SQL Workbench, the Workload Extraction feature of Speedgain for DB2 will be fully integrated into the workbench and will make it easy to capture and evaluate complete workloads.

However, while it is best to tune a workload, many times workloads are not available and only single SQL statements captured by snapshot monitoring are available. With input from developers, DBAs can still make significant performance improvements by tuning single SQL statements as long as all input from developers has been considered prior to making index changes. In a later paper, I will address how event monitoring can be used to accomplish this, but at an increased overhead.

What I have just described is an iterative SQL tuning process using SQL Workbench. This process or a similar process should be used by developers and DBAs to develop top performing SQL during all phases of application development, from construction to implementation. The Speedgain SQL Workbench, with its significant integration with Design Advisor, can help you and your company "Make SQL Fly".