

Understanding Profiling

Profiling continuously samples the data source to build a statistical model of the load on the database. Profiling can be used to locate and diagnose problematic SQL code and event-based bottlenecks. Additionally, profiling enables you to investigate execution and wait time event details for individual stored routines. Results are presented in the SQL Profiling Editor, which enables users to identify problem areas and subsequently drill down to individual, problematic SQL statements.

Profiling filters out well performing, light weight SQL and collects information on heavy weight SQL. SQL that is heavy weight is either long running queries or queries that are short but run so often that they put load on the database

Profiler takes snapshots of user and session activity once a second and builds up a statistical model of the load on the database. The sampled data is displayed in three ways:

- Load on the database measured in average number of sessions active
- Top Activity - Top SQL, Event, and Sessions, Object I/O, and Procedures
- Details - Detail on a SQL statement, Session, Event, Object I/O or Procedure

The graph on the top of the screen shows the load on the database and can quickly indicate visually how the database is functioning. The database could be:

- Idle
- Lightly loaded
- Heavily loaded
- Bottlenecked

Problems can come from any one or more of four areas:

- Machine CPU or Engine, slow disks (network)
- Application locks, invalid SQL
- Database cache sizes, log files, etc
- Inefficient SQL

Details of profiling sessions can be saved to an .oar file, which you can access through the SQL Project Explorer.