Helsinki 08242026

Whitepaper - Predictive Database Monitoring and Performance Optimization with SQL Governor

A Technical Overview of System-Level Early Warning and Query-Level Performance Prediction

Executive Summary

Traditional database monitoring is predominantly reactive. Performance counters and query statistics are continuously collected, but operational action is usually triggered only when a metric crosses a predefined threshold or application performance has already deteriorated.

Two SQL Governor U.S. patent applications address this limitation at complementary levels:

- **US20250173203A1 — "Early warning mechanism on database management system workload"**
- **US20220309064A1 / Application US17/697,998 — "Method, a system and a computer program product for predicting database query performance"**

The first technology analyzes changes in the statistical behavior of DBMS-level workloads to identify developing performance risk. The second analyzes query-level performance time series, dynamically selects suitable forecasting models, predicts future query-performance values, and can compare future performance across alternative query plans.

Together, they establish two complementary layers of predictive database observability:

System-level prediction:
Is the database environment developing toward an abnormal or problematic state?

Query-level prediction:
Which queries or query plans are likely to develop performance problems, and when?

The combined approach supports a transition from reactive incident detection toward proactive database performance optimization.

1. The Limitation of Conventional Threshold Monitoring

Conventional database monitoring typically evaluates metrics such as CPU utilization, processor queues, memory pressure, I/O rates, throughput, and query execution statistics.

A traditional monitoring rule might be:

CPU > 80% → Warning

CPU > 90% → Alert

This is useful for identifying an existing resource problem, but it does not necessarily recognize a slowly developing one.

For example, a database operating at 65% CPU may still appear healthy against an 80% threshold. But if that database historically operated at 30–40% and utilization has been gradually increasing for several weeks, the **direction and statistical behavior of the workload** may be more important than its current absolute value.

US20250173203A1 specifically addresses this limitation by comparing recent trends in DBMS performance-counter characteristics with longer-term baseline behavior.

The conceptual shift is:

Conventional monitoring

Current value → threshold → alert

versus:

Predictive monitoring

Historical behavior → current behavior → statistical change → future risk

2. System-Level Early Warning

US20250173203A1 periodically collects DBMS performance-counter samples and derives higher-level statistical characteristics from those measurements.

The patent identifies counters including CPU usage and processor queue length and describes additional applicable metrics such as memory utilization, Page Life Expectancy, IOPS, and throughput.

Rather than analyzing only averages or peaks, the system evaluates statistical components including:

Slope

Measures whether utilization is progressively increasing or decreasing.

A positive change in slope may indicate that a resource is moving toward increasing pressure, even if its current utilization remains below a conventional alert threshold.

Volatility

Measures changes in the variability of the workload.

Increasing volatility can expose intermittent or irregular performance behavior that would be difficult to detect from averages alone.

Skewness

Measures changes in the statistical distribution of workload values.

For example, a distribution becoming increasingly concentrated at higher utilization levels may indicate developing performance pressure.

Kurtosis

Provides another measure of how the distribution and concentration of workload observations are changing.

The patent proposes evaluating several such components for a performance counter rather than relying on a single statistic.

Current Trend Versus Historical Baseline

The central methodology compares:

Current trend
Recent statistical behavior of a performance-counter component

with:

Baseline trend
Longer-term historical behavior of that same component in the same DBMS.

One example described in the patent uses performance data collected frequently, aggregated into hourly data points, with trend points calculated periodically. A current trend may cover approximately 30 days while its baseline reflects a substantially longer period, such as three months. These periods can be changed to accommodate different workload characteristics or seasonality.

Conceptually:

Raw DBMS measurements

↓

Aggregated performance data

↓

Slope / volatility / skewness / kurtosis

↓

Current trend ↔ baseline trend

↓

Deviation analysis

↓

Early warning

Multi-Signal Risk Classification

A significant characteristic of the approach is that overall warning severity can be based on several statistical components of the same performance counter.

For example:

CPU

Slope → Warning

Volatility → Alert

Skewness → Warning

Kurtosis → Normal

Rather than treating these observations independently, their combination can contribute to an overall assessment of future performance risk.

The patent describes severity classes including warnings, alerts, and critical alerts. Thresholds can be determined individually for different performance-counter components, statistically derived from baseline behavior, and potentially adjusted using machine learning trained with historical counter data and known DBMS performance problems. It also describes extrapolating recent trends to predict whether a developing condition may become problematic in the near future.

The objective is therefore to identify **complex or subtle changes before they develop into obvious resource problems**.

3. Query-Level Performance Prediction

US20220309064A1 approaches prediction at a finer level.

Instead of primarily analyzing server or DBMS resource behavior, it analyzes the historical performance of individual database queries and optionally their alternative execution plans. The patent defines a **Query Performance Counter (QPC)** as a measurable characteristic associated with query execution. Examples described include execution time, worker/CPU time, rows returned, and physical and logical reads and writes.

The methodology can be represented as:

Query performance data

↓

Time-series construction

↓

Multiple forecasting models

↓

Statistical best-fit selection

↓

Future QPC prediction

↓

Threshold comparison

↓

Predictive alert

This is considerably more specific than simply extrapolating average query duration.

4. Collecting Query Performance Data

The architecture includes an analysis apparatus that obtains QPC data from the DBMS through applicable performance interfaces.

Examples identified in the patent include Dynamic Management Views, Performance Monitor interfaces, and Extended Events. Query-performance information can be stored as time-series data in a separate analysis environment or repository.

Separating the analytical workload from the monitored production DBMS is technically important: potentially expensive historical analysis and forecasting can be performed without unnecessarily adding computational load to the database system being observed.

A simplified architecture is therefore:

Production DBMS

↓

Performance APIs

↓

Query Performance Counter repository

↓

Predictive analysis apparatus

↓

Prediction / Alerting

5. Multiple Time Dimensions

A particularly interesting aspect of US20220309064A1 is that query behavior does not need to be analyzed along only one conventional chronological timeline.

The patent distinguishes between **consecutive** and **parallel** time dimensions.

A consecutive time dimension can examine adjacent periods such as:

Day 1 → Day 2 → Day 3 → Day 4

A parallel time dimension can compare recurring corresponding periods, conceptually:

Monday → next Monday → next Monday → next Monday

This makes it possible to model workload patterns that may follow business cycles, weekly behavior, or other recurring temporal structures.

For each query, QPC, time dimension, and selected time span, the system can construct a separate time series for predictive analysis. Query-plan information can also be retained as metadata.

This provides a richer forecasting model than assuming that query behavior evolves uniformly over continuous clock time.

6. Selecting the Forecasting Model Dynamically

The query-prediction methodology does not require one predetermined forecasting algorithm to be used for every workload.

Instead, the patent generates **multiple candidate trends using different mathematical forecasting methods**.

Examples include:

- linear regression;
- polynomial forecasting;
- exponential forecasting;
- and, in query-plan examples, moving-average methods.

The candidate models are compared statistically against the actual historical QPC time series. A best-fitting trend is then selected for forecasting future performance.

One described method for evaluating goodness of fit is the **coefficient of determination, R^2**. The patent also discusses alternatives or supplementary techniques such as residual analysis and RMSD.

Conceptually:

Historical query data

↓

Linear model ──────┐

Polynomial model ──────┼→ Statistical comparison → Best model

Exponential model ──┘

↓

Forecast future QPC

This allows the forecasting method to adapt to the shape of the observed performance behavior rather than assuming that every query follows a linear trajectory.

7. Predicting Future Query Performance

Once the best-fitting candidate trend has been selected, it is extended into the future to generate one or more predicted QPC data points.

For example, if **worker time** is being analyzed, the result can represent expected future processor consumption by the query.

The predicted value is then compared with an acceptable range or threshold. If a future predicted point exceeds the applicable threshold, the system can trigger an alert before the predicted condition actually occurs.

This changes query monitoring from:

"This query has become expensive."

to:

"This query's historical behavior predicts that it will become expensive."

The patent notes that a predicted threshold violation may reveal a hidden anomaly whose historical effect was relatively minor but whose future occurrence could have a more severe impact.

8. Learning From Prediction Accuracy

The methodology also includes a feedback mechanism.

After the predicted future period arrives and the query actually executes, the system can compare the predicted QPC behavior against the realized measurement.

The patent describes evaluating the difference between predicted and actualized goodness of fit and using historical prediction accuracy to refine confidence thresholds. In this way, the forecasting system can evaluate how reliably a particular model predicts a given QPC and time dimension over time.

Conceptually:

Historical data

↓

Prediction

↓

Future execution occurs

↓

Predicted value ↔ actual value

↓

Prediction accuracy

↓

Confidence / threshold adjustment

This provides an important distinction between simply fitting a curve and maintaining a predictive system whose confidence can be evaluated from realized outcomes.

9. Predicting Query-Plan Performance

The patent also extends prediction below the query level to individual **query execution plans**.

A query may have multiple execution plans, and those plans may perform differently depending on workload conditions and time dimensions.

US20220309064A1 describes generating and selecting forecasting trends separately for alternative query plans, predicting their future QPC values, and comparing the predicted performance of those plans.

For example:

Query Q

Plan A → predicted worker time: lower

Plan B → predicted worker time: higher

The predicted results can help identify which plan is expected to perform best or worst for a particular QPC and time dimension. The patent explicitly describes using predicted future performance as a preferred criterion in query-plan performance comparison.

This extends the technology from **predictive monitoring** toward **predictive optimization**.

Instead of only predicting:

This query may become problematic.

the analysis can potentially provide:

This query may become problematic, and one available execution plan is predicted to perform better than another.

10. How the Two Technologies Complement Each Other

The two patent applications operate at different analytical layers.

Layer 1 — DBMS Workload Intelligence

US20250173203A1 examines system-level performance behavior.

It asks:

Is the statistical nature of the database workload changing in a potentially harmful direction?

Inputs include DBMS performance counters such as CPU, processor queue, memory, I/O, and throughput. Analysis focuses on characteristics such as slope, volatility, skewness, and kurtosis relative to historical baselines.

Layer 2 — Query and Query-Plan Intelligence

US20220309064A1 analyzes performance closer to the SQL workload.

It asks:

Which query or query plan is predicted to develop performance problems, in which time dimension, and according to which performance metric?

It uses QPC time series, multiple forecasting methods, statistical model selection, future-value prediction, and threshold evaluation.

Together:

DBMS telemetry

↓

Statistical workload analysis

↓

Emerging system-level risk

↓

Query/QPC time-series analysis

↓

Forecast-model selection

↓

Query and query-plan prediction

↓

Workload prioritization

↓

DBA investigation

↓

Preventive optimization

The early-warning technology can indicate **that the operating environment is moving toward risk**, while query-performance prediction can provide more granular intelligence about **where performance risk may exist inside the workload**.

11. Example Operational Scenario

Consider a database where no conventional alert threshold has yet been exceeded.

System-level analysis detects that:

- CPU slope is increasing;
- I/O volatility is rising;
- processor-queue behavior differs from its historical baseline;
- multiple statistical components now indicate deterioration.

The early-warning layer therefore indicates an emerging DBMS-level performance risk.

Query-level analysis then examines historical QPC data.

For one high-volume SQL workload, several forecasting models are generated. Exponential forecasting fits poorly, polynomial forecasting provides a moderate fit, while linear regression provides the strongest statistical fit.

The selected model predicts that worker time will exceed its acceptable range during the forecast horizon.

The same analysis is performed for alternative execution plans, and one plan is predicted to consume substantially less worker time than another.

The operational workflow becomes:

1. Detect
Statistical changes indicate developing system-level risk.

2. Predict
Query analysis forecasts a future performance issue.

3. Localize
The relevant query, QPC, time dimension, and potentially problematic execution plan are identified.

4. Diagnose
The DBA investigates query logic, execution plans, indexing, statistics, resource contention, or configuration.

5. Optimize
Corrective action can occur before application performance is materially affected.

This transforms the typical workflow from:

Incident → diagnosis → remediation

toward:

Early warning → prediction → diagnosis → prevention

12. Technical and Operational Value

The combination provides several important advantages.

Earlier Detection

Changes can potentially be identified before resource consumption or query duration crosses conventional production thresholds.

Environment-Specific Intelligence

System-level baselines are derived from the monitored DBMS's own historical behavior, while query predictions are based on the actual historical performance of individual workloads.

Adaptive Forecasting

Query prediction can evaluate several forecasting methods and statistically select the one that best represents a specific time series rather than forcing all workloads into the same model.

Better Root-Cause Prioritization

System-level early warning can identify developing resource risk, while query-level prediction can focus investigation on workloads and execution plans whose future characteristics warrant attention.

Predictive Query-Plan Optimization

Forecasting alternative execution plans provides an additional opportunity to move from identifying future performance problems toward selecting or recommending better execution behavior.

Foundation for Automation

The early-warning patent describes providing warning information to resource-management systems capable of taking automated preventive measures. Combined conceptually with query and query-plan prediction, this creates a foundation for increasingly automated performance-management workflows.

Conclusion

Predictive database management requires more than observing current utilization.

It requires understanding:

- **How is the database changing?**
- **Is that change likely to become harmful?**
- **Which queries are likely to become problematic?**
- **Which execution plans are predicted to perform better or worse?**

US20250173203A1 addresses the first two questions by comparing current statistical workload characteristics—including slope, volatility, skewness, and kurtosis—with longer-term DBMS-specific baselines.

US20220309064A1 addresses the latter questions through query-performance time series, multiple time dimensions, competing forecasting models, statistical model selection, future QPC prediction, alert thresholds, prediction-accuracy feedback, and query-plan comparison.

Together, the technologies describe a complementary architecture:

System-level early warning

-

Query- and query-plan-level prediction

=

Predictive database performance intelligence

The resulting objective is not merely to detect database problems faster. It is to provide enough advance information to investigate and optimize the environment **before the predicted performance problem materially affects production workloads**.

Patent References

US20250173203A1
Early warning mechanism on database management system workload
U.S. Application 19/042,061; inventor Jani K. Savolainen; assignee listed by Google Patents as SQL Governor Oy.

US20220309064A1
Method, a system and a computer program product for predicting database query performance
U.S. Application 17/697,998; inventor Jani K. Savolainen; assignee listed by Google Patents as SQL Governor Oy.

This whitepaper provides a technical overview of the published patent applications and should not be interpreted as a legal analysis of patent claims, validity, enforceability, or scope.

Jani K. Savolainen
Founder & CTO,
SQL Governor