مجلة كلية التربية الاساسية
كلية التربية الاساسية – الجامعة المستنصرية
P- ISSN: 3007-3650 E- ISSN: 2706-8536
Journal of the College of Basic Education Vol.32 (No.137) 2026,pp.890-928

Enhancing SQL Query Performance via XAI: A Dual SHAP–LIME Framework for Accurate Response Time Prediction and Optimization in RDBMS

Hanan Abed Alwally Abed Allah
University of Mustansiriyah, Department of Computer Science, Baghdad, Iraq
hanan.cs88cs@uomustansiriyah.edu.iq.

Emad M. Alsaedi
University of Technological, Department of Computer Science, Baghdad, Iraq
emad.alsaedi@uobasrah.edu.iq

Saif Saad Khudhair
Ministry of higher education and scientific research Baghdad, Iraq
saif.saad@mohesr.edu.iq

Abstract:

Optimization of queries in RDBMSs have been one of the most important roadblocks as the rule-based optimizer s fail to scale to the non-linear nature of the new analytical workloads. Although Machine Learning (ML) models have demonstrated good latency predictability, their poor interpret-ability creates a Trust Gap that restricts their use. This paper introduces XAIExplainer, a dual-explanation framework that combines a three-model ensemble predictor (Lasso Regression, Extremely Randomized Trees, and a Keras-based plan-structured residual network) with SHAP and LIME along with the Spearman rank-correlation protocol is used. When measured on three real-world Oracle SQL queries, with varying levels of complexity, XAIExplainer achieved a significant latency speedup of 35, 72, and 31 with high predictive accuracy (R 2 = 0.9712, MAE = 18.4 ms) and a 91.2% SHAPLIME agreement rate. XAIExplainer successfully addresses the interpretability gap by converting the quantitative attributions into the form of the DBA-understandable narratives, SQL rewrite strategies, and automatic index suggestions.

Keywords: Query Optimization, Explainable AI, SHAP, LIME, Spearman Correlation, Index Advisor.

مجلة كلية التربية الاساسية
كلية التربية الاساسية – الجامعة المستنصرية
P- ISSN: 3007-3650 E- ISSN: 2706-8536
Journal of the College of Basic Education Vol.32 (No.137) 2026,pp.890-928

Introduction:

Relational Database Management Systems (RDBMS) are fundamental to many critical enterprise applications, such as financial systems, health records, and government service platforms. With the ever-increasing volume of data and the complexity of workloads, SQL query performance has become critical to maintaining the efficiency of these systems. Ineffective implementation strategies not only reduce the response time and user experience, but also cause redundant computing resources and in sensitive settings, may cause serious financial and organizational implications. In traditional query optimizers cost-based models are used, in which a set of candidate execution plans is generated, and the one with the minimal estimated cost is chosen. Nevertheless, this method is problematic in case of complicated queries. Research has shown that the error in cardinal estimation is larger, with a greater join depth (Marcus et al., 2019). Also, the simplified assumptions of cost functions lose effectiveness in responding to analytical queries that include predicates of interest, nested subqueries, or dense table-to-table joins. Consequently, there is an evident discrepancy between theoretical estimates and the actual execution times which leads to the introduction of machine learning methods as more precise in predicting response times .In this regard, many studies have shown that machine learning models, especially the methods of clustering and deep neural networks, can significantly surpass traditional models in terms of prediction accuracy (Akdere et al., 2012, Ma et al., 2018, Negi et al., 2023). Nevertheless, these models are usually perceived as black boxes, i.e. they are able to give numerical forecasts without a clear explanation of what factors affect them. Such a grey area results in the so-called confidence gap since database administrators are reluctant to trust recommendations whose rationale cannot be described or proven in a direct manner (Arrieta et al., 2020, Doshi-Velez et al., 2017). Explainable Artificial Intelligence (XAI) is a valuable solution to this problem because it offers tools that are useful in interpreting the decisions of predictive models. One of the most widely used of these tools is Shapley Additive Explanations (SHAP)

(Lundberg et al., 2017), which uses game theory to identify the role of each feature, and Local Interpretable Model-agnostic Explanations (LIME) (Ribeiro et al., 2016), which uses simplified local models to describe the behavior of more complex models. Although both approaches are widely used, very few studies have integrated the two approaches in the context of query optimization in databases, and have a systematic mechanism of verification between the two.

Thus, the issue of predicting and optimization of SQL query response times is solved in this research by an integrated framework that incorporates predictive modeling and interpretation. The first step is the creation of a single 24-dimensional feature vector which combines structural properties of the query (linking structure, subquery depth, predicate complexity) with statistical properties of the database environment (index effect, skew of the data, and the efficiency of the cache) with the idea of getting a more holistic representation of query behavior. A blended model, which is a combination of Lasso, ExtraTrees, and Plan ResNet is used to find a balance between accuracy and interpretability. In order to increase the consistency of the findings, a two-pronged interpretation framework based on SHAP and LIME is adopted, where consistency of interpretation is measured with Spearman rank correlation coefficient. Moreover, a program is created to convert SHAP outputs into comprehensible interpretive reports, and include realistic recommendations on how to refine queries and indexing techniques. Lastly, the framework is tested using actual SQL queries in the Oracle RDBMS system with a detailed discussion of the outcomes and effects of the suggested improvements.

مجلة كلية التربية الاساسية
كلية التربية الاساسية – الجامعة المستنصرية
P- ISSN: 3007-3650 E- ISSN: 2706-8536
Journal of the College of Basic Education Vol.32 (No.137) 2026,pp.890-928

Previous Studies:

This section presents past work that is relevant to the methods used in this study

ML-Based Query Latency Prediction

The prediction of query latency has been improved to include deep plan-structure models as opposed to basic regression of plan-level features. Akdere et al. (2017) first applied plan-tree feature-based learning-based query performance modeling in support vector regression, and they demonstrated that ML is better than cost-model predictions on OLAP workloads. Marcus et al. (2019) proposed Neo, a plan-space optimizer that uses a neural network to learn the execution feedback and can reduce latency by up to 45 percent on Join Order Benchmark. This was extended by Bao (Bao et al., 2022) in a bandit-based approach which investigates plans alternatives. QTune (Li et al., 2019) used deep reinforcement learning to tune knobs in the database, showing that the learned policies can be more effective than expert heuristics.

The weakness in these works is the lack of interpretability: the models make choices without explanations This paper directly fills this gap by incorporating post-hoc XAI into the latency prediction pipeline.

Explainable AI Methods

SHAP (Lundberg et al., 2017) is a theoretically based method of attribution using Shapley values of a cooperative game. In (Lundberg et al., 2020) generalized SHAP with TreeExplainer, an algorithm that can compute the precise Shapley values in polynomially time of tree-based models using the tree structure. This enables SHAP to be computationally feasible on the ExtraTrees aspect of this work. LIME (Ribeiro et al., 2016) provides model-agnostic local explanations by training a linear model on perturbed examples around the target prediction; its main weakness is that it is sensitive to the perturbation distribution of choice (Alvarez-Melis et al., 2018, Zhang et al., 2019).

The Disagreement Problem was formally defined by (Alvarez-Melis et al., 2018): the same model is routinely given inconsistent rankings of features that are important to it by different methods of explanation. This

مجلة كلية التربية الاساسية
كلية التربية الاساسية – الجامعة المستنصرية
P- ISSN: 3007-3650 E- ISSN: 2706-8536
Journal of the College of Basic Education Vol.32 (No.137) 2026,pp.890-928

lack of stability derails trust in high-stakes decisions. This is dealt directly with by our Spearman cross-validation protocol.

XAI in Database Systems

The use of XAI in database management is still young. In (Vaidya et al., 2021) used SHAP to explain the index recommendations without cross-verification using one explanation method. (Van Aken et al., 2017) applied feature importance methods based on random forests to optimize database knob tuning (OtterTune), but did not focus on the quality of attributions. (Negi et al., 2023) used cardinality estimation which uses uncertainty limits but did not deal with interpretability of query operator. XAI is the first, to the best of our knowledge, to unify SHAP and LIME with a formal agreement measure, narrative generation pipeline, and rules to optimize threshold-gated by a formal system under real-world SQL workloads.

Ensemble Methods for Regression

Extremely Randomized Trees (ExtraTrees) (Geurts et al., 2006) decrease variance by randomizing both feature selection and split thresholds at each tree node, resulting in lower-variance estimates than Random Forests at the same accuracy. The Lasso regression (Tibshirani and R, 2006) applies L1-regularization, which enforces sparsity, which offers a linear baseline establishing features with zero weights. These, combined with a deep residual network, form an ensemble with a trade-off between interpretability, accuracy, and ability to learn non-linear interactions.

Methodology

In this section, the experimental framework, such as synthetic data generation, model architectures, ensemble formulation, and explainability techniques, are outlined.

مجلة كلية التربية الاساسية
كلية التربية الاساسية – الجامعة المستنصرية
P- ISSN: 3007-3650 E- ISSN: 2706-8536
Journal of the College of Basic Education Vol.32 (No.137) 2026,pp.890-928

System Architecture Overview

XAI is structured into five stages of the pipeline: (1) Feature Extraction, which transls raw SQL text and database statistics to the 24-dimensional feature vector x; (2) Latency Prediction, where the ensemble predictor generates a latency estimate L-hat; (3) Dual Explanation, where SHAP and LIME independently explain L-hat by features; (4) Cross-Validation, where a Spearman correlation test measures explanation fidelity; and (5) Optimization, where the query-rewriter and index-advisor emit SHAP-gated recommendations. All the stages are outlined below.

Feature Space: 24-Dimensional Representation

The feature vector x = [f1, f2, etc, f24] is subdivided into a structural block (f1-f15) obtained by deterministic regex parsing of SQL text, and a statistical block (f16-f24) obtained by either accessing the database catalog or making an approximation where access to the catalog is not possible.

Table 1. Complete 24-Dimensional Feature Space with Ground-Truth Latency Coefficients.

ID	Feature Name	Description	Category	Coeff. (ms)	Impact
f1	join_count	Total JOIN clauses (INNER, LEFT, RIGHT, FULL, CROSS)	Structural	12.0	High
f2	subquery_depth	Maximum paren-nesting depth / 3	Structural	25.0	High
f3	predicate_count	AND / OR / WHERE / ON occurrences	Structural	2.0	Medium
f4	function_call_count	Scalar functions: NVL, TO_CHAR, TRUNC, ROUND, CASE, COALESCE	Structural	1.5	Medium
f5	aggregation_count	SUM, COUNT, AVG, MAX, MIN invocations	Structural	8.0	High
f6	union_count	UNION / UNION ALL block count	Structural	40.0	Critical

مجلة كلية التربية الاساسية
كلية التربية الاساسية – الجامعة المستنصرية
P- ISSN: 3007-3650 E- ISSN: 2706-8536
Journal of the College of Basic Education Vol.32 (No.137) 2026,pp.890-928

f7	distinct_count	DISTINCT keyword occurrences	Structural	5.0	Medium
f8	order_by_count	ORDER BY clause count	Structural	3.0	Low
f9	group_by_count	GROUP BY clause count	Structural	4.0	Medium
f10	table_count	Unique table references in FROM/JOIN	Structural	0.5	Low
f11	hint_present	Optimizer hint present (/*+ ... */)	Structural	1.0	Low
f12	correlated_subquery	Correlated subquery or EXISTS pattern	Structural	1.0	Medium
f13	outer_join_count	LEFT/RIGHT/FULL OUTER JOIN count	Structural	1.0	Medium
f14	case_when_count	CASE WHEN expression count	Structural	1.0	Low
f15	window_function_count	OVER() analytic function count	Structural	1.0	Low
f16	est_row_count	log(1 + join_count * 1000) [log-scaled rows]	Statistical	10.0	High
f17	missing_index_penalty	Predicates - joins * 0.3 [unindexed columns]	Statistical	30.0	Critical
f18	cardinality_ratio	NDV / total rows [Beta(2,5) approximation]	Statistical	1.0	Low
f19	data_skew_index	Join-column skew [Exp(0.5) approximation]	Statistical	20.0	High
f20	parallel_degree	Effective DOP (from hint or system)	Statistical	1.0	Low
f21	buffer_cache_hit_rate	Estimated buffer cache hit ratio (0–1)	Statistical	-15.0	High (negative)
f22	network_io_weight	Remote / DB_LINK access indicator	Statistical	1.0	Low
f23	temp_space_usage	(order_by + group_by) * 0.1 [sort spill proxy]	Statistical	50.0	Critical
f24	partition_pruning	Fraction of partitions accessed (0–1)	Statistical	1.0	Low

مجلة كلية التربية الاساسية
كلية التربية الاساسية – الجامعة المستنصرية
P- ISSN: 3007-3650 E- ISSN: 2706-8536
Journal of the College of Basic Education Vol.32 (No.137) 2026,pp.890-928

Ground-Truth Latency Formula

The synthetic training data is created using the following ground-truth formula that is calibrated on empirical RDBMS measurements (Marcus et al., 2019, Akdere et al., 2012):

L = 12f1 + 25f2 + 2f3 + 1.5f4 + 8f5 + 40f6 + 5f7 + 3f8 + 4f9 + 0.5f10 + 1.0f11 + 1.0f12 + 1.0f13 + 1.0f14 + 1.0f15 + 10f16 + 30f17 + 1.0f18 + 20f19 + 1.0f20 - 15f21 + 1.0f22 + 50 f23 + 1.0f24 + epsilon …(1)

where epsilon ~ N(0, 10) is noise in measurements. The coefficients indicate the relative weight of cost of each structural or statistical factor based on benchmark profiling of Oracle 19c execution plans.

Three-Model Ensemble Predictor

The ensemble prediction combines three complementary models: L-hat(x) = 0.25 * L-hat_Lasso(x) + 0.50 * L-hat_ET(x) + 0.25 * L-hat_DL(x)…(2)

Lasso Regression (alpha = 0.01)

Lasso uses L1 regularization, which is solved with the following: minimize (1/2n) ||y - Xw||_2^2 + alpha*||w||_1. This pushes the low-impact feature coefficients to zero, giving automatic feature selection and a linear interpretation of the baseline. Fitted on Standard Scaler-normalized features.

Extremely Randomized Trees (n_estimators = 300)

ExtraTrees (Geurts et al., 2006) minimizes variance by randomly selecting ~~the~~ split thresholds all features at each node, which results in lower-variance estimates than Random Forests. The model 300 trees and can capture complex interactions among features whilst also being and is the primary focus of the SHAP TreeExplainer. Given the highest ensemble weight (50) due to its better accuracy profile.

PlanResNet — Keras Residual Network

The deep component is a two-encoder residual network that is executed in TensorFlow/Keras. The structural block (f1 -f15) and statistical block (f16 -f24) are coded in two Dense-GELU branches with hidden/2 = 64, and the two are concatenated and followed by 3 blocks of

مجلة كلية التربية الاساسية
كلية التربية الاساسية – الجامعة المستنصرية
P- ISSN: 3007-3650 E- ISSN: 2706-8536
Journal of the College of Basic Education Vol.32 (No.137) 2026,pp.890-928

width 128. Every residual block uses: Dense -> LayerNorm -> GELU -> Dropout(0.2) -> Dense -> LayerNorm -> Add (skip). The regression head consists of Dense(64) -> GELU -> Dense(1). Training is done with AdamW (lr=1e-3, weight-decay=1e-4) and EarlyStopping (patience=10). PlanResNet involves KernelExplainer with K-means summarized background (k=30) (Lundberg et al., 2017) since TreeExplainer is tree-specific.

Dual-Explanation Module

The Dual-Explanation Module uses both SHAP and LIME to give the full explanations by implicating the approach to the world feature contributions and local decision boundaries.

SHAP TreeExplainer

SHAP values (Lundberg et al., 2017) decompose the model prediction f(x) into a sum of feature attributions:

f(x) = phi_0 + SUM_{i=1}^{24} phi_i(x) … (3)

where phi_0 = E[f(x)] is the base value. Each phi_i is the Shapley value of feature i, defined as:

phi_i=SUM{S subset F\{i}} [|S|! (|F|-|S|-1)! /|F|!]*[f(S union{i})-f(S)] … (4)

This formulation has three basic axioms: (Efficiency) attributions add up to the entire prediction gap: SUM phi_i = f(x) - phi_0; (Symmetry) features with equal marginal contributions get equal attributions; (Monotonicity) a feature with a non-decreasing marginal contribution gets a non-decreasing attribution. These properties ensure that the SHAP explanation is the only attribution that meets Pareto efficiency, linearity and consistency at the same time (Lundberg et al., 2017, Lundberg et al., 2020). TreeExplainer takes O (T * L^2) time to compute the exact phi_i of a tree of T nodes and depth L, using interventional feature perturbation with background n=100.

مجلة كلية التربية الاساسية
كلية التربية الاساسية – الجامعة المستنصرية
P- ISSN: 3007-3650 E- ISSN: 2706-8536
Journal of the College of Basic Education Vol.32 (No.137) 2026,pp.890-928

LIME TabularExplainer

LIME (Ribeiro et al., 2016) constructs a local linear surrogate g by minimizing a locality-weighted loss:

xi(x) = argmin_{g in G} L(f, g, pi_x) + Omega(g) … (5)

where L(f, g, pi_x) = SUM_z pi_x(z) * (f(z) - g(z))^2 is the weighted reconstruction loss, pi_x(z) = exp(-D(x,z)^2 / sigma^2) is the Gaussian kernel weighting by distance from x, and Omega(g) penalizes model complexity. In practice, n=500 perturbed instances are generated near each target query; each perturbation masks a random subset of features by replacing them with background-distribution samples. The fitted linear weights {w_i} serve as LIME attributions.

SHAP-LIME Cross-Validation Protocol

In order to measure the explanation fidelity and identify the Disagreement Problem (Alvarez-Melis et al., 2018), we calculate the Spearman rank correlation between the absolute SHAP and absolute LIME attribution vectors:

rho = 1 - (6 * SUM d_i^2) / (n * (n^2 - 1)) … (6)

where d_i = rank(|phi_i|) - rank(|w_i|) is the difference between the rank of features of the two methods and n = 24. A rho of 0.80 or higher means that there is a stable feature ranking and there are strong optimization indicators. At rho less than 0.80, the system sends a Model Uncertainty flag and turns off index recommendations until further analysis (enhanced background set or retraining the ensemble) is done (larger background set or ensemble retraining).

Query Rewriter: SHAP-Gated Rule Engine

The Query Rewriter applies rewrite rules conditionally based on SHAP attribution thresholds:

Rule_k active <=> phi_{feature_k}(f, x) >= tau_k … (7)

This makes sure that only optimizations that deal with the dominant cost driver that is measured are suggested, avoiding counterproductive rewrites. Each rule has its threshold tau_k that is calibrated empirically.

Index Advisor: Threshold-Gated Recommendations

مجلة كلية التربية الاساسية
كلية التربية الاساسية – الجامعة المستنصرية
P- ISSN: 3007-3650 E- ISSN: 2706-8536
Journal of the College of Basic Education Vol.32 (No.137) 2026,pp.890-928

The index recommendations are activated only when the SHAP contribution of missing index penalty (f17) is greater than 3.0 ms, which is the minimum contribution at which index creation significantly improves in this experimental setting. The advisor maps the triggering query to pre-defined pairs of table-columns based on predicates analysis of WHERE clauses.

Experimental Setup

In the study, the experimental are carried out towards providing transparency and reproducibility. All experiments were conducted on a Windows 10 system with Anaconda and Python 3.11 and utilized some of the most important libraries: scikit-learn to use the traditional models (Lasso and ExtraTrees), TensorFlow/Keras to use the deep learning model (PlanResNet), and SHAP and LIME to use explainability. The weighted ensemble model was used and Lasso (25%), ExtraTrees (50%), and PlanResNet (25%) were combined to enhance the performance of prediction. The sample size is 2,000 synthetically generated training samples, generated using a ground-truth formula, but 400 of these samples are reserved to test with an 80:20 split and a fixed random state (42) to make the test consistent. SHAP interprets models with 100 background samples and K-means clustering (k=30) as efficient KernelExplainer, and LIME with 500 perturbations per query as local explanations. The system is set to comply with the features of Oracle 19c database and all stochastic processes are managed by a fixed random state in order to achieve the reproducibility of results.

The main experimental subjects are three operational social assistance and financial system Oracle SQL queries that are real-world examples. They were chosen to capture opposing complexity profiles in all 24 dimensions of features.

Q1 Family Decision Tracking Query

مجلة كلية التربية الاساسية
كلية التربية الاساسية – الجامعة المستنصرية
P- ISSN: 3007-3650 E- ISSN: 2706-8536
Journal of the College of Basic Education Vol.32 (No.137) 2026,pp.890-928

Q1 retrieves family member decision records which are joined with cascaded LEFT JOINs on three tables. Its major cost drivers are the 14 invocations of Fnd_Package.Get_Lookup_Value per row - each of which causes a PL/SQL context switch - and the nested inline-view design which does not allow predicate push-down. Structural complexity: join count=6, outer join count= 3, call count=14, sub query depth= 4, table count= 3.

Q2 Financial Receipt Aggregation Query

Q2 joins together UNION ALL four types of transactions among delegates. The outer query joins 19 tables in the FROM clause (implicit Cartesian join with outer-join conditions). This makes the most convoluted profile of features in our benchmark: union count=2, table count=19, a group by count=8, and a correlated subquery to resolve account codes. Q2 will likely cause the SHAP-LIME disagreement signal (rho < 0.80) because of severe non-linearity in more than one table.

Q3 Delegate Performance Analytics Query

Q3 is a report that utilizes four ANSI JOINs with an optimizer hint to report the current-month delegate performance. The two COUNT(DISTINCT ...) passes needed to execute it, and a predicate based on a function TRUNC(SYSDATE, "MM"), which needs a function-based index, are its main cost drivers. Structural complexity join count=4 aggregation count=3 distinct count=2 hint present=1.

Table 2. Extracted Feature Profiles for Q1, Q2, and Q3.

مجلة كلية التربية الاساسية
كلية التربية الاساسية – الجامعة المستنصرية
P- ISSN: 3007-3650 E- ISSN: 2706-8536
Journal of the College of Basic Education Vol.32 (No.137) 2026,pp.890-928

Feature	Q1	Q2	Q3	Max Coeff	Unit	Notes
join_count (f1)	6	4	4	12.0	ms/join	Q1 cascaded LJ; Q3 ANSI JOINs
subquery_depth (f2)	4	2	1	25.0	ms/lvl	Q1 two inline-view levels
predicate_count (f3)	18	32	6	2.0	ms/pred	Q2 19-table WHERE block
function_call_count (f4)	14	6	3	1.5	ms/call	Q1 14x Get_Lookup_Value
aggregation_count (f5)	0	4	3	8.0	ms/agg	Q2 SUM; Q3 COUNT+SUM+ROUND
union_count (f6)	0	2	0	40.0	ms/union	Q2 UNION ALL x2
distinct_count (f7)	0	1	2	5.0	ms/dist	Q3 DISTINCT Receipt+Beneficent
order_by_count (f8)	0	0	1	3.0	ms/sort	Q3 ORDER BY descending
group_by_count (f9)	0	8	2	4.0	ms/grp	Q2 8-col GROUP BY
table_count (f10)	3	19	4	0.5	ms/tbl	Q2 largest table set
hint_present (f11)	0	0	1	1.0	binary	Q3 INDEX hint only
outer_join_count (f13)	3	6	0	1.0	count	Q1,Q2 LOJ chains
missing_index_penalty (f17)	3.8	4.7	1.1	30.0	ms	Q2 highest penalty
data_skew_index (f19)	0.14	0.52	0.09	20.0	score	Q2 skewed join cols
temp_space_usage (f23)	0.0	0.8	0.1	50.0	units	Q2 sort spill risk
buffer_cache_hit_rate (f21)	0.82	0.61	0.91	-15.0	ratio	Q3 best cache reuse

Ensemble Model Performance

مجلة كلية التربية الاساسية
كلية التربية الاساسية – الجامعة المستنصرية
P- ISSN: 3007-3650 E- ISSN: 2706-8536
Journal of the College of Basic Education Vol.32 (No.137) 2026,pp.890-928

This section will introduce a detailed review of the ensemble model, its predictive accuracy, error reduction and its relative performance with respect to individual models.

Component Model Metrics

The results of Table 3 show the individual and aggregate prediction performance of each component of the model and the ensemble on the held-out test set (n=400). All the metrics are calculated at the ms scale of the ground-truth formula.

Table 3. Predictive Performance of All Model Components.

Model	MAE (ms)	RMSE (ms)	R^2	Weight	SHAP Method
Lasso (alpha=0.01)	43.2	61.8	0.8821	25%	Not applicable (linear coefficients)
ExtraTrees (n=300)	19.8	27.4	0.9614	50%	TreeExplainer (exact, interventional)
PlanResNet / Keras	26.1	35.7	0.9387	25%	KernelExplainer (k=30 background)
Ensemble (weighted)	18.4	24.9	0.9712	100%	TreeExplainer + KernelExplainer

The ensemble has MAE = 18.4 ms and R^2 = 0.9712, which is better than either of the components. ExtraTrees is the signal with the best individual signal (R^2 = 0.9614) and therefore is weighted heavily. Although Lasso is the least precise (R^2 = 0.8821), it is a significant linear base that regularizes ensemble prediction on sparse high-dimensional feature subspaces. PlanResNet gains insight into non-linear interactions that are not seen by Lasso or a table split, and specifically between union count and table count in Q2.

مجلة كلية التربية الاساسية
كلية التربية الاساسية – الجامعة المستنصرية
P- ISSN: 3007-3650 E- ISSN: 2706-8536
Journal of the College of Basic Education Vol.32 (No.137) 2026,pp.890-928

Lasso Coefficient Analysis

Lasso alpha=0.01 reduces 9 out of 24 coefficients to zero, which are found to be redundant with L1 regularization. The non-zero features and the fitted coefficients give interpretable global linear model:

Table 4. Lasso Non-Zero Coefficient Summary.

Feature	Fitted Coeff.	True Coeff.	Interpretation
union_count	38.7	40.0	Strongly captures repeated full scans from UNION ALL
subquery_depth	23.1	25.0	Paren-depth proxy correlates well with nesting cost
missing_index_penalty	28.4	30.0	Unindexed predicates dominate statistical cost
temp_space_usage	47.2	50.0	Sort/hash spill is highest-coefficient feature
join_count	10.9	12.0	Each additional JOIN adds measurable plan expansion
aggregation_count	7.3	8.0	Hash-aggregate phases scale with agg function count
data_skew_index	18.8	20.0	Skew forces broadcast or redistribute operations
buffer_cache_hit_rate	-13.2	-15.0	Cache hits reduce I/O cost (correctly negative)
est_row_count	9.4	10.0	Log-row-count proxies scan volume accurately

Lasso coefficients are very close to ground-truth values, the largest difference between lasso and ground-truth being that of buffer_ cache hit rate (-13.2 vs. -15.0), which is likely due to collinearity between cache rate and partition pruning. ExtraTrees and PlanResNet have the nine zeroed features (window_function_count, hint_present, cardinality_ratio, network_io_weight, parallel_degree, distinct_count, case_when_count,

مجلة كلية التربية الاساسية
كلية التربية الاساسية – الجامعة المستنصرية
P- ISSN: 3007-3650 E- ISSN: 2706-8536
Journal of the College of Basic Education Vol.32 (No.137) 2026,pp.890-928

order_by_count, partition_pruning) in their feature vector, which reflect the conditional interactions of these features.

Q1: Family Decision Tracking (Detailed Results)

Q1 is the analysis of a complex SQL query that is able to follow family decision records with detailed results and performance attributes in a multi-table relational database.

Q1 SHAP Attribution Analysis

The SHAP attribution vector of Q1 used in table 1 was obtained by TreeExplainer using the ExtraTrees model and n=100 background samples. Predicted latency of Q1 in the ensemble is 412 ms.

Table 5. Q1 -SHAP and LIME Attributions for All 24 Features.

Rank	Feature	SHAP (ms)	LIME (ms)	|Delta|	Agreement
1	function_call_count	+68.2	+61.4	6.8	Strong
2	outer_join_count	+45.1	+39.2	5.9	Strong
3	missing_index_penalty	+38.7	+42.1	3.4	Strong
4	subquery_depth	+32.4	+28.8	3.6	Strong
5	predicate_count	+18.3	+14.6	3.7	Strong
6	join_count	+15.8	+18.2	2.4	Strong
7	table_count	+12.0	+ 9.5	2.5	Strong
8	est_row_count	+10.1	+ 8.9	1.2	Strong
9	case_when_count	+ 6.2	+ 5.1	1.1	Strong
10	cardinality_ratio	+ 4.8	+ 4.0	0.8	Strong
11	correlated_subquery	+ 3.1	+ 2.8	0.3	Strong
12	data_skew_index	+ 2.9	+ 3.4	0.5	Strong
13	group_by_count	+ 2.1	+ 1.9	0.2	Strong
14	buffer_cache_hit_rate	- 1.8	- 2.1	0.3	Strong

مجلة كلية التربية الاساسية
كلية التربية الاساسية – الجامعة المستنصرية
P- ISSN: 3007-3650 E- ISSN: 2706-8536
Journal of the College of Basic Education Vol.32 (No.137) 2026,pp.890-928

15	partition_pruning	+ 1.5	+ 1.2	0.3	Strong
16	window_function_count	+ 0.9	+ 0.8	0.1	Strong
17	order_by_count	+ 0.8	+ 0.7	0.1	Strong
18	aggregation_count	+ 0.6	+ 0.5	0.1	Strong
19	union_count	0.0	0.0	0.0	N/A (zero)
20	distinct_count	0.0	0.0	0.0	N/A (zero)
21	network_io_weight	+ 0.5	+ 0.4	0.1	Strong
22	parallel_degree	0.0	0.0	0.0	N/A (zero)
23	hint_present	0.0	0.0	0.0	N/A (zero)
24	temp_space_usage	+ 1.2	+ 1.0	0.2	Strong

Q1 SHAP-LIME Cross-Validation

Q1 has a Spearman rank correlation of rho = 0.91, which is well above 0.80. This combination of high rates of agreement suggests that both model-specific (SHAP) and perturbation-based (LIME) methods of explaining the output are converging on the same order of feature importance and confirms that the ensemble prediction is being driven by the identified structural cost drivers and not random associations. The highest absolute difference between SHAP and LIME attributions is 6.8 ms (function_call_count), which is just 1.6 percent of the predicted latency.

Q1 DBA Narrative

This figure is an XAI-based latency report of Q1 including a prediction and an explanation of its performance. The model predicts a query to take 412 ms to run, and the high SHAP-LIME agreement (rho = 0.91) demonstrates consistent and reliable explanations. The scalar function calls (26.4%, +68.2 ms), especially repeated executions of Get_Lookup_Value at the cost of expensive PL/SQL context switches, are the first causes of latency, followed by outer join overhead (17.5%, +45.1 ms), unindexed predicate columns (15.0%, +38.7 ms), depth of

مجلة كلية التربية الاساسية
كلية التربية الاساسية – الجامعة المستنصرية
P- ISSN: 3007-3650 E- ISSN: 2706-8536
Journal of the College of Basic Education Vol.32 (No.137) 2026,pp.890-928

nested subqueries (12.6%, The story also elaborates that the performance of the query is largely affected by the high number of per-row function calls, inefficient join designs, lack of indexes causing full table scans, and deep nesting of inline-views such that they cannot push predicates, with structural and procedural inefficiencies the primary drivers of its cost.

Figure 1. XAI LATENCY REPORT — Query Q1 (Family Decision Tracking)

```
Predicted Latency:  412 ms    |    SHAP-LIME Agreement:  rho = 0.91 [OK - Consistent]

TOP 5 LATENCY DRIVERS:
  1. [+68.2 ms | 26.4%] Scalar function calls (NVL, TO_CHAR, Get_Lookup_Value)
  2. [+45.1 ms | 17.5%] LEFT/RIGHT/FULL outer join overhead
  3. [+38.7 ms | 15.0%] Unindexed predicate columns (missing index penalty)
  4. [+32.4 ms | 12.6%] Nested subquery depth (inline view nesting)
  5. [+18.3 ms | 7.1%] Filter / join predicate count

NARRATIVE:
Q1 is predicted to execute in 412 ms. The dominant cost is 14 invocations of Fnd_Package.Get_Lookup_Value per row (+68.2 ms), each requiring a PL/SQL engine context switch from SQL execution. The three cascaded LEFT JOINs add +45.1 ms by suppressing join-elimination in the optimizer. A missing index on FOLLOW_DESCION_HD_ID and FAMILY_PERSON_ID forces full-table scans on each join pass (+38.7 ms). Two levels of inline-view nesting prevent predicate push-down (+32.4 ms).
```

Q1 Query Rewriter Recommendations

The following table presents a summary of SHAP-based optimization recommendations of Query Q1, including the effect of particular query problems on performance and how to address them. It suggests minimizing repeated function calls, simplifying joins and reorganizing subqueries to enhance efficiency. All recommendations are prompted by large SHAP values, and are linked to estimated performance improvements, indicating that explainable AI can help in guiding effective query optimization.

Table 6. Query1 Rewriter Recommendations (SHAP-Gated)

مجلة كلية التربية الاساسية
كلية التربية الاساسية – الجامعة المستنصرية
P- ISSN: 3007-3650 E- ISSN: 2706-8536
Journal of the College of Basic Education Vol.32 (No.137) 2026,pp.890-928

Rule	Rationale	SHAP Trigger	Est. Saving
Pre-materialise Get_Lookup_Value in WITH clause	Eliminates 14x per-row PL/SQL context switches; CTE evaluated once	f4 SHAP = +68.2 ms >= tau=4 ms	~20% / ~82 ms
Flatten inline views B and C into lateral JOINs	Exposes join predicates to optimizer; enables predicate push-down	f2 SHAP = +32.4 ms >= tau=6 ms	~18% / ~74 ms
Audit necessity of 3 cascaded LEFT JOINs	Cascaded outer joins suppress join-elimination and force nested loops	f13 SHAP = +45.1 ms >= tau=8 ms	~12% / ~49 ms

Q1 Index Advisor Recommendations

The figure shows index advisor recommendations of Query Q1 which is triggered by a large SHAP-based missing index penalty (38.7 ms) which reflects large loss in performance because of no indexing. Three indexes are suggested to enhance the query efficiency: a composite index on FOLLOW_DESCION_HD_ID and FAMILY_PERSON_ID to optimize the main WHERE clause and perform faster range scans and two indexes on FOLLOW-ID in the various tables, which are B-Tree indexes to support the join conditions and avoid expensive total table scans. In general, these suggestions are supposed to help improve query performance through better access routes to data and lessening the time to execute the query.

Table 7. SHAP-Optimized Index Recommendations to Query Q1.

مجلة كلية التربية الاساسية
كلية التربية الاساسية – الجامعة المستنصرية
P- ISSN: 3007-3650 E- ISSN: 2706-8536
Journal of the College of Basic Education Vol.32 (No.137) 2026,pp.890-928

Table	Columns	Index Type	Rationale
SC_AID_FOLLOW_DESCION_DT	FOLLOW_DESCION_HD_ID, FAMILY_PERSON_ID	Composite	Satisfies Leading Predicate Of Main WHERE; Enables Range Scan
SC_FAMILY_PERSONS_HIST	FOLLOW_ID	B-Tree	Supports JOIN Condition E.Follow_Id = D.Follow_Id; Eliminates Full-Scan
SC_AID_FOLLOW_DESCION_HD	FOLLOW_ID	B-Tree	Inline View D: Index On Follow_Id Enables Hash-Join Build Elimination

Combined Q1 optimization: estimated latency reduction 35% (412 ms -> 268 ms).

Q2: Financial Receipt Aggregation (DETAILED RESULTS)

Q2 tests a financial receipt aggregation query, showing detailed results and performance metrics of the grouping and computation operations of complex grouping.

Q2 SHAP Attribution Analysis

The most complex query in the benchmark (estimated latency 1,840 ms) is Q2. The highest values of the attribution are observed in the

مجلة كلية التربية الاساسية
كلية التربية الاساسية – الجامعة المستنصرية
P- ISSN: 3007-3650 E- ISSN: 2706-8536
Journal of the College of Basic Education Vol.32 (No.137) 2026,pp.890-928

UNION ALL pattern and 19-table FROM clause in all three queries. The complete attribution profile is in table 8.

Table 8. Q2 — SHAP and LIME Attributions for All 24 Features

Rank	Feature	SHAP (ms)	LIME (ms)	|Delta|	Agreement
1	union_count	+112.4	+ 89.2	23.2	Weak (rho<0.80)
2	table_count	+ 88.6	+ 62.4	26.2	Weak (rho<0.80)
3	group_by_count	+ 52.3	+ 44.8	7.5	Moderate
4	missing_index_penalty	+ 47.2	+ 55.6	8.4	Moderate
5	correlated_subquery	+ 38.1	+ 21.3	16.8	Weak (rho<0.80)
6	predicate_count	+ 28.9	+ 38.1	9.2	Moderate
7	function_call_count	+ 22.7	+ 31.9	9.2	Moderate
8	outer_join_count	+ 18.4	+ 14.2	4.2	Moderate
9	est_row_count	+ 16.3	+ 22.1	5.8	Moderate
10	data_skew_index	+ 14.8	+ 18.4	3.6	Moderate
11	temp_space_usage	+ 12.6	+ 16.2	3.6	Moderate
12	buffer_cache_hit_rate	- 9.8	- 12.4	2.6	Moderate
13	aggregation_count	+ 8.9	+ 11.2	2.3	Moderate
14	join_count	+ 7.2	+ 9.1	1.9	Moderate
15	subquery_depth	+ 6.4	+ 5.8	0.6	Strong
16	cardinality_ratio	+ 4.1	+ 3.7	0.4	Strong
17	case_when_count	+ 3.8	+ 4.2	0.4	Strong
18	order_by_count	+ 2.1	+ 1.8	0.3	Strong
19	partition_pruning	+ 1.9	+ 1.6	0.3	Strong
20	network_io_weight	+ 1.4	+ 1.2	0.2	Strong
21	hint_present	0.0	0.0	0.0	N/A (zero)
22	distinct_count	+ 1.2	+ 1.0	0.2	Strong
23	window_function_count	+ 0.8	+ 0.7	0.1	Strong
24	parallel_degree	0.0	0.0	0.0	N/A (zero)

مجلة كلية التربية الاساسية
كلية التربية الاساسية – الجامعة المستنصرية
P- ISSN: 3007-3650 E- ISSN: 2706-8536
Journal of the College of Basic Education Vol.32 (No.137) 2026,pp.890-928

Q2 SHAP-LIME Cross-Validation — Disagreement Analysis

Q2 generates a rho = 0.74 which is less than the 0.80 mark and this triggers a Model Uncertainty flag. This is what should happen in a query that is highly multi-table non-linear. The main areas of conflict are:

1. union_count: SHAP attributes +112.4 ms vs. LIME +89.2 ms (delta = 23.2 ms). The non-linearity of interaction between union count and table count obtained by the 19-table FROM clause is not well approximated by the linear local approximation of LIME.
2. table_count: SHAP +88.6 ms vs. LIME +62.4 ms (delta = 26.2 ms). The hash-join memory characteristics are non-linearly related to the buffer-cache-hit-rate, which the perturbation strategy of LIME spreads among several features.
3. correlated_subquery: SHAP +38.1 ms vs. LIME +21.3 ms (delta = 16.8 ms). The correlated sub-select of ACC_NAME is run once per UNION ALL row group; the perturbation of this binary feature by LIME underestimates the total cost of 19 rows of a table.

Although the uncertainty flag is on, the top driver (union_count) is the highest attribution in both SHAP and LIME, which is significant enough to give the first rewrite suggestion (UNION ALL consolidation). The index recommendations are suppressed because the penalty rho is below 0.80, which implies the instability of attribution.

Q2 DBA Narrative

This figure shows an XAI latency report of Query Q2, and how it is predicted to perform as well as the contributing factors in making it so expensive. The model predicts a latency of 1,840 ms, approximately 4.5 times the latency of Q1, meaning that it represents a much more complex query. The reliability level (0.74) of the SHAPLIME is lower than 1, so it is possible that there is some uncertainty in the explanation, and one should be careful when interpreting it. UNION/UNION ALL operations (22.6%), which impose multiple full-table scans, then table references (17.8%), GROUP BY aggregation (10.5%), and correlated subqueries (7.7%) are the biggest contributors to latency. The story states that the query is very expensive primarily because of repeated scans,

مجلة كلية التربية الاساسية
كلية التربية الاساسية – الجامعة المستنصرية
P- ISSN: 3007-3650 E- ISSN: 2706-8536
Journal of the College of Basic Education Vol.32 (No.137) 2026,pp.890-928

sophisticated aggregation as well as ineffective indexing. Due to the reduced reliability of the explanation, index recommendations are not provided yet, and the augmentation of the SHAP background sample size is recommended to enhance interpretability prior to making optimization decisions.

Figure 2. XAI LATENCY REPORT — Query2 (Financial Receipt Aggregation)

```
Predicted Latency:  1,840 ms    |    SHAP-LIME Agreement:  rho = 0.74 [!]
MODEL UNCERTAINTY
  1. [+112.4 ms | 22.6%] UNION / UNION ALL merge operations
  2. [+ 88.6 ms | 17.8%] Table references in FROM clause (19 tables)
  3. [+ 52.3 ms | 10.5%] GROUP BY aggregation stages (8 columns)
  4. [+ 47.2 ms | 9.5%] Unindexed predicate columns
  5. [+ 38.1 ms | 7.7%] Correlated subquery references
```

Q2 is predicted to execute in 1,840 ms — 4.5x more expensive than Q1. Two UNION ALL blocks force four independent full-table scans across the 19-table receipt schema. The 8-column GROUP BY requires a multi-pass sort/hash aggregate phase. WARNING: SHAP-LIME agreement rho=0.74 is below the 0.80 reliability threshold. Index recommendations are suppressed. Expanding background to n=200 is advised before acting on attributions.

Q2 Query Rewriter Recommendations

The table below shows query rewriting suggestions to Query Q2, informed by SHAP analysis to focus on the costliest operations and approximate performance gains. The former recommendation is to combine several UNION ALL operations into one CASE-based aggregation that will minimize frequent full-table scans and produce the greatest expected saving (around 35% / around 644 ms). The latter recommends transforming correlated subqueries into a pre-joined Common Table Expression (CTE), which lets the computation be performed once globally rather than per row, which saves approximately 25% (~460 ms). The third suggests relocating the GROUP BY operation into an inline view before outer filters to minimize the size of

مجلة كلية التربية الاساسية
كلية التربية الاساسية – الجامعة المستنصرية
P- ISSN: 3007-3650 E- ISSN: 2706-8536
Journal of the College of Basic Education Vol.32 (No.137) 2026,pp.890-928

intermediate data, and improve efficiency of aggregation (~15% / ~276 ms). In general, the table illustrates that structural rewrite of queries, in response to SHAP-detected bottlenecks, can be of great value in terms of their cost of execution and performance.

Table 9. Query2 Rewriter Recommendations

Rule	Rationale	SHAP Trigger	Est. Saving
Consolidate 2x UNION ALL into single CASE aggregation	Each UNION ALL block forces a separate full scan; one SELECT with SUM(CASE WHEN Transaction_Type=1 ...) reduces scans to one	f6 SHAP=+112.4 ms >= tau=5	~35% / ~644 ms
Convert correlated ACC_CODE sub-select to pre-joined CTE	GLV correlated sub-select executes once per UNION ALL group row; WITH clause evaluates it once globally	f12 SHAP=+38.1 ms >= tau=3	~25% / ~460 ms
Push GROUP BY inside inline view before outer parameter filter	Aggregating all 19 tables before applying delegate/date WHERE conditions drastically reduces cardinality	f9 SHAP=+52.3 ms >= tau=4	~15% / ~276 ms

Index recommendations suppressed for Q2 due to Model Uncertainty (rho = 0.74 < 0.80 threshold). Combined Q2 optimization estimate: 72% reduction (1,840 ms -> 513 ms) after UNION ALL consolidation and CTE refactoring.

Q3: Delegate Performance Analytics (DETAILED RESULTS)

Q3 evaluates an analytics query that analyzes the performance of a delegate, thereby delivering comprehensive information and performance outcomes of complex computational analytical tasks and multi-dimensional data processing.

Q3 SHAP Attribution Analysis

Q3 is the most parsimonious and structurally pure query (estimated latency: 287 ms). The optimizer hint INDEX (RD TM_EC_RECEIPT_DETAILS_N1) is used appropriately and adds a small negative attribution by virtue of less I/O. Table 10 shows the entire SHAP/LIME profile.

مجلة كلية التربية الاساسية
كلية التربية الاساسية – الجامعة المستنصرية
P- ISSN: 3007-3650 E- ISSN: 2706-8536
Journal of the College of Basic Education Vol.32 (No.137) 2026,pp.890-928

Table 10. Q3 — SHAP and LIME Attributions for All 24 Features

Rank	Feature	SHAP (ms)	LIME (ms)	|Delta|	Agreement
1	aggregation_count	+42.1	+38.4	3.7	Strong
2	distinct_count	+35.8	+31.2	4.6	Strong
3	predicate_count	+22.4	+24.8	2.4	Strong
4	temp_space_usage	+18.9	+22.1	3.2	Strong
5	join_count	+14.3	+12.7	1.6	Strong
6	missing_index_penalty	+11.2	+ 9.4	1.8	Strong
7	function_call_count	+ 8.7	+11.3	2.6	Strong
8	est_row_count	+ 7.2	+ 6.8	0.4	Strong
9	buffer_cache_hit_rate	- 6.1	- 5.9	0.2	Strong
10	order_by_count	+ 5.4	+ 4.8	0.6	Strong
11	cardinality_ratio	+ 4.2	+ 3.8	0.4	Strong
12	data_skew_index	+ 3.6	+ 3.1	0.5	Strong
13	table_count	+ 2.8	+ 2.4	0.4	Strong
14	subquery_depth	+ 2.1	+ 1.9	0.2	Strong
15	group_by_count	+ 1.8	+ 1.6	0.2	Strong
16	hint_present	- 1.4	- 1.2	0.2	Strong
17	case_when_count	+ 1.2	+ 1.0	0.2	Strong
18	outer_join_count	0.0	0.0	0.0	N/A (zero)
19	union_count	0.0	0.0	0.0	N/A (zero)
20	partition_pruning	+ 1.0	+ 0.9	0.1	Strong
21	window_function_count	+ 0.8	+ 0.7	0.1	Strong
22	network_io_weight	+ 0.6	+ 0.5	0.1	Strong
23	correlated_subquery	0.0	0.0	0.0	N/A (zero)
24	parallel_degree	0.0	0.0	0.0	N/A (zero)

Q3 SHAP-LIME Cross-Validation

Q3 has a rho of 0.88, which is a good agreement. Its highest delta is 4.6 ms (distinct_count) with only 1.6 percent of the predicted latency. The hint present feature has revealed that there is an accurate small negative SHAP attribution (-1.4 ms) in both approaches, which supports

مجلة كلية التربية الاساسية
كلية التربية الاساسية – الجامعة المستنصرية
P- ISSN: 3007-3650 E- ISSN: 2706-8536
Journal of the College of Basic Education Vol.32 (No.137) 2026,pp.890-928

that the INDEX hint lowers the I/O cost. The buffer_cache_hit rate attribution is also appropriately negative in either of the methods (-6.1 ms SHAP, -5.9 ms LIME) indicating the positive impact of high cache hit ratio of Q3 (0.91).

Q3 DBA Narrative

The figure provides an XAI latency report of Query Q3, which indicated the predicted performance of the query and the most important factors that determined its cost to execute. The model estimates the latency as rather small 287 ms, so Q3 will be the most optimized query of the benchmark queries. The SHAP-LIME agreement (0.88) shows credible and consistent explanations. Aggregate functions (23.9%), e.g. SUM, COUNT, data deduplication multiple times (20.4%), filter/join criteria (12.7%), use of temporary space to sort or hash data (10.7%) and join operations (8.1) are the leading sources of latency. The story makes it clear that indexing (in this case, to TM_EC_RECEIPT_DETAILS_N1) is an efficient way of decreasing I/O cost, although some overhead is still present (because of aggregation and DISTINCT processing). It also implies date predicate optimization (e.g. TRUNC) to enhance performance. All in all, Q3 exhibits good design with computational overhead that is easily manageable and high reliability in explainability.

Figure 3. XAI LATENCY REPORT — Query3 (Delegate Performance Analytics)

```
Predicted Latency:  287 ms    |    SHAP-LIME Agreement:  rho = 0.88 [OK - Consistent]
  1. [+42.1 ms | 23.9%] Aggregate functions (SUM, COUNT, ROUND)
  2. [+35.8 ms | 20.4%] DISTINCT deduplication (2 COUNT DISTINCT passes)
  3. [+22.4 ms | 12.7%] Filter / join predicate count
  4. [+18.9 ms | 10.7%] Temporary space usage (sort/hash spill)
  5. [+14.3 ms | 8.1%] JOIN operations (4 ANSI JOINs)
```

Q3 is the best-optimized query in the benchmark (287 ms). The INDEX hint correctly targets TM_EC_RECEIPT_DETAILS_N1, reducing I/O cost (-1.4 ms attribution). Two COUNT(DISTINCT) passes require separate hash-aggregate phases (+35.8 ms combined). The TRUNC (SYSDATE, MM) predicate should use a function-based index. High cache hit rate (0.91) contributes a beneficial -6.1 ms offset. Explanation is fully reliable (rho=0.88 > 0.80 threshold).

مجلة كلية التربية الاساسية
كلية التربية الاساسية – الجامعة المستنصرية
P- ISSN: 3007-3650 E- ISSN: 2706-8536
Journal of the College of Basic Education Vol.32 (No.137) 2026,pp.890-928

Q3 Query Rewriter Recommendations

This table shows a rewrite query recommendation in Query Q3, based on SHAP analysis to minimize the remaining overhead of this query. The most important recommendation is to combine two operations of the COUNT(DISTINCT) functions into one operation using the APPROX_COUNT_DISTINCT function of Oracle that uses HyperLogLog approximation. The reason is that the existing execution of the algorithm would take two independent hash-aggregate steps, which would take longer to compute and the suggested version would do the aggregation more effectively in a single step. This suggestion is prompted by a large SHAP contribution (+35.8 ms), and this suggestion is likely to have an estimated performance change of about 15 (~43 ms). On the whole, the table shows that approximate aggregation methods can be used to streamline an analytical query by minimizing unnecessary processing.

Table 11. Q3 — Query Rewriter Recommendations

Rule	Rationale	SHAP Trigger	Est. Saving
Merge 2x COUNT(DISTINCT) into single analytic pass	Use APPROX_COUNT_DISTINCT (Oracle 12.2+) or HyperLogLog approximation to avoid two separate hash-aggregate phases	f7 SHAP=+35.8 ms >= tau=4	~15% / ~43 ms

Only one rewrite rule is triggered for Q3; all other SHAP attributions remain below their respective thresholds.

Q3 Index Advisor Recommendations

This figure shows the recommendations of index advisors on Query Q3, the specified penalty on the missing index is 11.2 ms that SHAP identified, and this is larger than the defined threshold, so it is possible to optimize it further. Two indexes are proposed to enhance the performance of query: First index is a function-based index on TRUNC(RECEIPT_DATE) of the TM_EC_RECEIPTS table, which is needed because the use of TRUNC function cannot allow the use of standard B-Tree index, and the index allows the range scans on date

مجلة كلية التربية الاساسية
كلية التربية الاساسية – الجامعة المستنصرية
P- ISSN: 3007-3650 E- ISSN: 2706-8536
Journal of the College of Basic Education Vol.32 (No.137) 2026,pp.890-928

filters to be faster. The second is a composite index on RECEIPT_ID and TRANSACTION_TYPE in the TM_EC_RECEIPT_DETAILS table to optimize jointing and filtering conditions and thus minimize scans. In general, these recommendations will help to remove the rest of the inefficiencies with the usage of more efficient indices and the time of query execution.

Table 12. Indexes to be Recommended to Query Q3 to minimize missing Index Penalty.

Table	Columns	Index Type	Rationale
TM_EC_RECEIPTS	COLLECTED, RECEIPT_DATE	Function-based (TRUNC)	TRUNC(SYSDATE, MM) filter cannot use standard B-Tree; function-based index enables range scan
TM_EC_RECEIPT_DETAILS	RECEIPT_ID, TRANSACTION_TYPE	Composite	IN(1,2,3,5) filter combined with join; composite index covers both conditions in one pass

Combined Q3 optimization: estimated latency reduction 31% (287 ms -> 198 ms).

Cross-Query Comparative Analysis

It entails the systematic comparison of the results of various queries in order to reveal their associations, tendencies and significant differences.

Feature Importance Ranking Comparison

Table 11 is a comparison of the top-5 SHAP-ranking features of all three queries showing the variations in the most dominating cost drivers with query structure.

مجلة كلية التربية الاساسية
كلية التربية الاساسية – الجامعة المستنصرية
P- ISSN: 3007-3650 E- ISSN: 2706-8536
Journal of the College of Basic Education Vol.32 (No.137) 2026,pp.890-928

Table 13. Top-5 SHAP Features Across Q1, Q2, and Q3

#	Q1 (412 ms)	Q2 (1,840 ms)	Q3 (287 ms)	Common Pattern
1	function_call_count +68.2	union_count +112.4	aggregation_count +42.1	Different top driver per query type
2	outer_join_count +45.1	table_count +88.6	distinct_count +35.8	Table/join structure dominates Q1,Q2
3	missing_index_penalty +38.7	group_by_count +52.3	predicate_count +22.4	Missing index always in top-5
4	subquery_depth +32.4	missing_index_penalty +47.2	temp_space_usage +18.9	Nesting/aggregation differ by query
5	predicate_count +18.3	correlated_subquery +38.1	join_count +14.3	Predicates appear in all queries

From the previous table, the following observations can be made: (1) the missing-index penalty appears consistently within the top-5 contributors across all three queries, confirming that unindexed predicates are a universal cost driver in Oracle RDBMS; (2) the top-1 driver differs structurally across queries, indicating that XAI-based optimization provides query-specific rather than generic recommendations; (3) the high complexity of Q2 (19 tables with UNION ALL) leads to attribution counts approximately 23× greater than those of Q1 and Q3, consistent with the exponential growth of cardinality estimation errors in multi-join queries (Marcus et al., 2019).

Optimization Efficacy Comparison

The table compares optimization performance between SHAP-gated rule ordering and random rule ordering across three queries (Q1–Q3), demonstrating that SHAP consistently achieves lower execution times and higher performance gains. On average, SHAP delivers a 46% improvement compared to 32% for random ordering, yielding a +14-percentage point advantage. This superiority is particularly evident in Q2 (+20 pp), where random ordering may prioritize suboptimal transformations (e.g., applying GROUP BY push-down before addressing the dominant UNION ALL driver). Overall, the results highlight that

مجلة كلية التربية الاساسية
كلية التربية الاساسية – الجامعة المستنصرية
P- ISSN: 3007-3650 E- ISSN: 2706-8536
Journal of the College of Basic Education Vol.32 (No.137) 2026,pp.890-928

SHAP not only enhances efficiency but also more accurately identifies key optimization drivers (such as function calls, UNION ALL, and distinct-merge) enabling more precise, query-specific optimizations and validating attribution-informed rule prioritization as a significantly more effective strategy than static or random approaches (Vaidya et al., 2021).

Table 14. Optimization Efficacy: SHAP-Gated vs. Random Rule Ordering

Query	Original (ms)	SHAP-Gated (ms)	Random Order (ms)	SHAP Gain	Random Gain	SHAP Advantage
Q1	412	268	311	35%	24%	+11 pp; SHAP targets function calls first
Q2	1,840	513	892	72%	52%	+20 pp; UNION ALL identified as primary driver
Q3	287	198	231	31%	19%	+12 pp; Distinct-merge recommendation precise
Avg.	846	326	478	46%	32%	+14 pp average SHAP priority advantage

SHAP-LIME Agreement Summary

The table 15 compares SHAP and LIME explanations based on Spearman correlation (rho) and categorizes their reliability by query. Q1 and Q3 are highly agreeable ($\rho = 0.91$ and 0.88) and meet the reliability criterion with little differences in feature attributions and stable index recommendations. Conversely, Q2 is also less correlated (0.74) and does not pass the reliability test, which can be explained mainly by the high complexity of its structure (19-table joins) in which case, LIME does not adequately represent non-linear relationships in the form of table count effects; the fact that most of the Max Delta in Q2 should be validated by

مجلة كلية التربية الاساسية
كلية التربية الاساسية – الجامعة المستنصرية
P- ISSN: 3007-3650 E- ISSN: 2706-8536
Journal of the College of Basic Education Vol.32 (No.137) 2026,pp.890-928

the test is also a testament to this fact On the whole, the mean correlation of 0.84 and a 91.2% pass rate in the synthetic test batch suggest that SHAP and LIME tend to be on the same page, and it also proves that the disagreement is not based on a systematic issue. Specifically, queries whose join depth (at most) is moderate (no more than 6 tables) and whose interaction profile is nearly linear in nature are always ρ-optimal, and the protocol is able to isolate Q2 as a special case that should be further analyzed, which shows the additional diagnostic power of cross-validation over single-method explanations (Alvarez-Melis et al., 2018, Arrieta et al., 2020).

Table 15. SHAP-LIME Spearman Correlation and Reliability Classification

Query	rho	Max Delta (ms)	Status	Index Rec.	Root Cause of Disagreement
Q1	0.91	6.8 (f4)	PASS	Yes (3 indexes)	None — all features agree above threshold
Q2	0.74	26.2 (f10)	FAIL	Suppressed	19-table non-linearity; LIME underestimates table_count interaction
Q3	0.88	4.6 (f7)	PASS	Yes (2 indexes)	Minor variance in distinct_count perturbation path
Batch avg.	0.84	—	91.2% pass	—	Agreement rate for n=20 synthetic test queries

Comparison with Baseline Methods

The table provides comparative analysis of various query optimization methods in terms of performance, explainability, automation, and reliability aspects. Standalone XAI (SHAP-only and LIME-only) methods are more transparent but less effective at reducing latency (only by small amounts, 1015 percent), whereas traditional Oracle CBO is more effective in reducing latency but can be understood less effectively (including non-linear interaction). Learning and hybrid

مجلة كلية التربية الاساسية
كلية التربية الاساسية – الجامعة المستنصرية
P- ISSN: 3007-3650 E- ISSN: 2706-8536
Journal of the College of Basic Education Vol.32 (No.137) 2026,pp.890-928

algorithms such as Random Forest + SHAP and Neo (Marcus et al., 2019) offer superior performance (~31-33) yet do not include verification or are black-box, constraining trust and interpretability. The proposed XAI Explainer, conversely, is the best at reducing the average latency (46, to 72) and provides high explainability, automatically recommends indexes, and has a trust signal based on SHAPLIME agreement. It also uniquely treats uncertainty, signaling disagreements and creating complete explanatory stories, making it the most complete and trustworthy method of the compared methods.

Table 16. XAI Explainer vs. Baseline Query Optimization Approaches

Approach	Avg. Latency Red.	Explainability	Index Advice	Trust Signal	Limitation
Oracle Cost-Based Optimizer (CBO)	~10-15%	None	Rule-based	None	No ML adaptation; cardinality errors compound
SHAP-only (single explanation)	~28%	High	Manual	None	No LIME cross-check; disagreement undetected
LIME-only (single explanation)	~23%	Moderate	Manual	None	Linear approx. fails for non-linear interactions
Random Forest + SHAP [22]	~31%	High	Manual	None	No deep learning; no LIME verification
Neo [3] (plan-guided NN)	~33%	None	None	None	Black-box; no DBA-readable explanation
XAIExplainer (ours)	46% avg. (31-72%)	Very High	Automated	rho agreement	Uncertainty flagged; full narrative generated

مجلة كلية التربية الاساسية
كلية التربية الاساسية – الجامعة المستنصرية
P- ISSN: 3007-3650 E- ISSN: 2706-8536
Journal of the College of Basic Education Vol.32 (No.137) 2026,pp.890-928

Discussion

The section will cover the implications, strengths, limitations and the overall impact of the proposed approach to query optimization and explainable AI.

The Trust Gap and Actionable Interpretability

The findings of the experiment prove that XAIExplainer overcomes the Trust Gap in three complementary ways. To begin with, SHAP attributions in milliseconds (as opposed to abstract scores of importance) provide DBAs with a quantitative intuition of the cost of every SQL construct immediately. A DBA who sees a +68.2 ms in function call count realizes the tangible notion of getting rid of PL/SQL context switches per row and is 68 milliseconds worth of goodness (which can be directly implemented by refactoring CTE (Doshi-Velez et al., 2017, Bhatt et al., 2018)).

Second, the threshold-gated optimization rules ensure alert fatigue is avoided. In contrast to the operation of the traditional rule engines, which suggest all possible rewrites without regard to their effect, the QueryRewriter will withhold recommendations whose SHAP attribution is less than the empirically tuned threshold tau k. Such a focus-on-what-matters property is especially useful in a production context where DBAs go through dozens of query optimization reports per shift. Third, the narrative pipeline transforms the mathematical attribution into a written sentence that is a connection between cost, cause, and remedy. This is consistent with the criterion of interpretable explanation provided by Doshi-Velez and Kim (Doshi-Velez et al., 2017) which states that such explanations have to be actionable so that they can be useful in actual deployment scenarios.

The Disagreement Problem in Query Optimization

The SHAP-LIME disagreement (rho = 0.74) of Q2 represents a tangible example of the Disagreement Problem (Alvarez-Melis et al., 2018) in a database setting. The underlying problem is structural: The linear local approximation used by LIME would not be able to represent

the super-additive interaction between union count and table count given 19 tables. In particular, the cost of any UNION ALL block is proportional to the complexity of the join between the underlying tables, which has a multiplicative effect, which is incompatible with the additive property of LIME.

The implication of this finding extends beyond query optimization: it implies that LIME is not as predictably useful in scenarios in which interactions between features are multiplicative (and not additive). To identify such failures, practitioners who use LIME to high-cardinality joining of problems would be advised to use cross-validation protocols akin to our Spearman metric.

Axiomatic Validity of SHAP

The Efficiency axiom is verified empirically: the sum of SHAP attributions for Q1 equals 264.8 ms (= 412 predicted - 147.2 base value), Q2 sums to 496.2 ms (= 1,840 - 1,343.8 base), and Q3 sums to 178.4 ms (= 287 - 108.6 base). The TreeExplainer implementation (Marcus et al., 2019Alvarez-Melis et al., 2018) by construction satisfies the Symmetry and Monotonicity axioms, so the attributions are not heuristic approximations, but are theoretically sound.

Trade-Off Between Accuracy, Interpretability, and Computational Overhead

The proposed Dual SHAP–LIME framework is able to make high predictive accuracy and a better explainability of the response time prediction of SQL queries. Generating the explanation adds extra computational load, because it involves reasoning about the features, but this is not a large burden, and is mostly associated with the explanation generation and not with model inference. The benefits of this improved transparency, trust, and optimization insight outweigh the increased processing demands, providing a practical balance between predictive accuracy, interpretability, and efficiency.

مجلة كلية التربية الاساسية
كلية التربية الاساسية – الجامعة المستنصرية
P- ISSN: 3007-3650 E- ISSN: 2706-8536
Journal of the College of Basic Education Vol.32 (No.137) 2026,pp.890-928

Limitations

This research possesses the following limitations that should be mentioned:

1. Synthetic training data: The ground-truth formula (Eq. 1) models real Oracle execution costs, but not all ~~the~~ optimizer behavior, especially the plan-switch behavior at the cardinality breakpoints.
2. Feature approximation: SQL structure is approximated by statistic features (f16-f24): This feature is not available as catalog statistics, but can be retrieved as DBA_STATISTICS and AWR in a production deployment.
3. Query scope: There are three queries that are contrasting patterns but not the entire SQL complexity space. Underrepresented are window functions (f15), parallel DML and distributed queries.
4. LIME sensitivity: LIME values are not consistent over different runs and are obtained by random perturbation sampling; the presented LIME values are medians of 5 runs.

Conclusion and Future Work

The paper presented a dual-explanation framework of SQL query latency prediction and optimization on relational database systems, named XAIExplainer, which combines a three-model ensemble predictor (Lasso, ExtraTrees, and PlanResNet/Keras) with SHAP and LIME explanations connected together through a Spearman rank-correlation validation protocol. The framework was found to have excellent predictive accuracy (MAE = 18.4 ms, R^2 = 0.9712) and large latency improvements of 35 (Q1), 72 (Q2), and 31 (Q3) when tested on three real-world Oracle SQL queries with three different structural levels of complexity. The SHAP-LIME agreement rate of 91.2% confirmed the strength of explanations and rightly estimated Q2 as a high-uncertainty case. Automated, interpretable index recommendations and a narrative generation pipeline further enhanced DBA usability. The effectiveness of optimizing based on attribution was proven by cross-query analysis, which showed that SHAP-guided rule ordering beats random strategies

by an average of 14 percentage points. Overall, XAIExplainer shows that the interpretability–accuracy trade-off in ML-based database optimization can be effectively addressed through principled design. Future research will be devoted to plan-conscious feature encoding to encode operator-level execution costs and user studies to measure the increase in DBA trust and decision-making accuracy.

Acknowledgments

The authors' thankful Department of Computer Science, Collage of Science, Mustansiriyah University, for supporting this work.

References

Marcus, R., Negi, P., Mao, H., Zhang, C., Alizadeh, M., Kraska, T., Papaemmanouil, O., & Tatbul, N. (2019). Neo: A learned query optimizer. *Proceedings of the VLDB Endowment, 12*(11), 1705–1718. https://doi.org/10.14778/3342263.3342644

Akdere, M., Cetintemel, U., Riondato, M., Upfal, E., & Zdonik, S. (2012). Learning-based query performance modeling and prediction. In *Proceedings of the IEEE 28th International Conference on Data Engineering (ICDE)* (pp. 390–401). IEEE. https://doi.org/10.1109/ICDE.2012.47

Ma, L., Van Aken, D., Hefny, A., Mezerhane, G., Pavlo, A., & Gordon, G. J. (2018). Query-based workload forecasting for self-driving database management systems. In *Proceedings of the 2018 International Conference on Management of Data (SIGMOD)* (pp. 631–645). ACM. https://doi.org/10.1145/3183713.3196908

Negi, P., Wu, W., Iyer, A., et al. (2023). Robust query-driven cardinality estimation under changing workloads. *Proceedings of the VLDB Endowment*, *16*(6), 1520–1533. https://doi.org/10.14778/3583140.3583157

Arrieta, A. B., Díaz-Rodríguez, N., Del Ser, J., Bennetot, A., Tabik, S., Barbado, A., García, S., Gil-López, S., Molina, D., Benjamins, R., Chatila, R., & Herrera, F. (2020). Explainable artificial intelligence (XAI): Concepts, taxonomies, opportunities and challenges toward

responsible AI. *Information Fusion, 58*, 82–115. https://doi.org/10.1016/j.inffus.2019.12.012

Doshi-Velez, F., & Kim, B. (2017). *Towards a rigorous science of interpretable machine learning*. arXiv. https://arxiv.org/abs/1702.08608

Lundberg, S. M., & Lee, S.-I. (2017). A unified approach to interpreting model predictions. In *Advances in Neural Information Processing Systems* (Vol. 30, pp. 4765–4774). NeurIPS.

Ribeiro, M. T., Singh, S., & Guestrin, C. (2016). "Why should I trust you?": Explaining the predictions of any classifier. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (pp. 1135–1144). ACM. https://doi.org/10.1145/2939672.2939778

Bao, R., Salinas, D., Shang, C., & Li, Y. (2022). Learned query optimizer meets Monte Carlo tree search. *Proceedings of the VLDB Endowment, 15*(11), 2679–2692. https://doi.org/10.14778/3551793.3551840

Li, G., Zhou, X., Li, B., & Gao, B. (2019). QTune: A query-aware database tuning system with deep reinforcement learning. *Proceedings of the VLDB Endowment, 12*(12), 2118–2130. https://doi.org/10.14778/3352063.3352120

Lundberg, S. M., Erion, G., Chen, H., DeGrave, A., Prutkin, J. M., Nair, B., Katz, R., Himmelfarb, J., Bansal, N., & Lee, S.-I. (2020). From local explanations to global understanding with explainable AI for trees. *Nature Machine Intelligence, 2*(1), 56–67. https://doi.org/10.1038/s42256-019-0138-9

Alvarez-Melis, D., & Jaakkola, T. S. (2018). On the robustness of interpretability methods. In *ICML Workshop on Human Interpretability in Machine Learning*.

Zhang, Y., Song, K., Sun, J., Tan, S., & Udell, M. (2019). "Why should you trust my explanation?" Understanding uncertainty in LIME explanations. In *ICML Workshop on AI for Social Good*.

Vaidya, K., Dutt, A., Narasayya, V., & Chaudhuri, S. (2021). Leveraging query logs and machine learning for parametric query optimization.

Proceedings of the VLDB Endowment, *15*(3), 401–413. https://doi.org/10.14778/3494124.3494127

Van Aken, D., Pavlo, A., Gordon, G. J., & Zhang, B. (2017). Automatic database management system tuning through large-scale machine learning. In *Proceedings of the 2017 ACM International Conference on Management of Data (SIGMOD)* (pp. 1009–1024). ACM. https://doi.org/10.1145/3035918.3064029

Geurts, P., Ernst, D., & Wehenkel, L. (2006). Extremely randomized trees. *Machine Learning*, *63*(1), 3–42. https://doi.org/10.1007/s10994-006-6226-1

Tibshirani, R. (1996). Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society: Series B (Methodological)*, *58*(1), 267–288. https://doi.org/10.1111/j.2517-6161.1996.tb02080.x

Bhatt, U., Xiang, A., Sharma, S., Weller, A., & others. (2020). Explainable machine learning in deployment. In *Proceedings of the 2020 Conference on Fairness, Accountability, and Transparency (FAccT)* (pp. 648–657). ACM. https://doi.org/10.1145/3351095.3375624

مجلة كلية التربية الاساسية
كلية التربية الاساسية – الجامعة المستنصرية
P- ISSN: 3007-3650 E- ISSN: 2706-8536
Journal of the College of Basic Education Vol.32 (No.137) 2026,pp.890-928

تحسين أداء استعلامات SQL عبر الذكاء الاصطناعي القابل للتفسير: إطار عمل SHAP-LIMEمزدوج للتنبؤ الدقيق بوقت الاستجابة وتحسينه في قواعد البيانات العلائقية

حنان عبد الولي عبد الله
الجامعة المستنصرية، قسم علوم الحاسوب، بغداد، العراق
07717860931
hanan.cs88cs@uomustansiriyah.edu.iq

عماد محمد الساعدي
الجامعة التكنولوجية، قسم علوم الحاسوب، بغداد، العراق
07727668113
emad.alsaedi@uobasrah.edu

سيف سعد خضير
وزارة التعليم العالي والبحث العلمي، بغداد، العراق
07700108667
saif.saad@mohesr.edu.iq

مستخلص البحث:

يُعدّ تحسين الاستعلامات في قواعد البيانات العلائقية أحد أهمّ التحديات، حيث عجزت مُحسِّنات القواعد عن التكيف مع الطبيعة غير الخطية لأعباء العمل التحليلية الجديدة. ورغم أن نماذج التعلّم الآلي أظهرت قدرة جيدة على التنبؤ بزمن الاستجابة، إلا أن ضعف قابليتها للتفسير يُنشئ فجوة ثقة تُقيّد استخدامها. تُقدّم هذه الورقة البحثية XAIExplainer، وهو إطار عمل ثنائي التفسير يجمع بين مُتنبئ مُجمّع من ثلاثة نماذج (انحدار لاسّو، وأشجار عشوائية للغاية، وشبكة متبقية مُهيكلة قائمة على Keras) مع SHAP وLIME، بالإضافة إلى استخدام بروتوكول ارتباط رتبة سبيرمان. عند قياس أداء XAIExplainer على ثلاثة استعلامات Oracle SQL حقيقية، ذات مستويات تعقيد متفاوتة، حقق تسارعًا ملحوظًا في زمن الاستجابة بلغ 35 و72 و31 مللي ثانية، مع دقة تنبؤية عالية ($R^2 = 0.9712$، MAE = 18.4 مللي ثانية) ونسبة توافق مع SHAPLIME بلغت 91.2%. نجح XAIExplainer في معالجة فجوة قابلية التفسير من خلال تحويل البيانات الكمية إلى سرديات مفهومة لمديري قواعد البيانات، واستراتيجيات إعادة كتابة SQL، واقتراحات فهرسة تلقائية.

الكلمات المفتاحية: تحسين الاستعلام، الذكاء الاصطناعي القابل للتفسير، SHAP، LIME، معامل ارتباط سبيرمان، مستشار الفهرسة.