JOURNAL OF BASIC SCIENCE AND ENGINEERING

Development of a Multi-Dimensional Business Intelligence Framework for Executive Reporting through Advanced Calendar Intelligence in Power BI

Ved Vinchurkar[1], Dev Bala[2], Aniket Alvekar[3], Rajkumar Prasad[4]
[1,2,3,4] School of Computer Science and Engineering, Sandip University, Nashik 422213, Maharashtra, India

Keywords:
Business Intelligence, Power BI, Calendar Intelligence, DAX Time Intelligence, Semantic Modelling, Calculation Groups, Fiscal Calendar, Executive Reporting, VertiPaq.

DOI:
https://doi.org/10.10399/JBSE.20267 39204

ABSTRACT
Business Intelligence (BI) platforms have shifted enterprise reporting from centralised information-technology production to governed executive self-service, yet the temporal layer of the semantic model - the calendar dimension - remains the most frequent source of both analytical error and query-performance degradation. Microsoft Power BI provides a rich but semantically shallow set of native time-intelligence functions that presuppose a regular Gregorian calendar, an assumption violated by the fiscal calendars, 4-4-5 retail periods, trading-day conventions and multi-entity hierarchies of real multi-sector enterprises. This paper develops and empirically evaluates a multi-dimensional BI framework in which advanced calendar intelligence is treated as a first-class architectural layer. Following a design-science methodology, the framework is specified in four layers - a marked and fully attributed date dimension, surrogate integer period keys, a time-intelligence calculation group, and calendar-aware aggregation with fiscal-period-aligned incremental refresh - and instantiated in a longitudinal deployment within a mid-sized multi-sector enterprise group operating retail and discrete-manufacturing divisions on a shared Power BI tenant over 2021 to 2025.

Across six canonical time-intelligence patterns benchmarked under warm-cache conditions, mean query response time fell from 8.05 s to 1.75 s, a reduction of 78.3 %, with the formula-engine share of query time falling from 71.4 % to 24.8 %. Full dataset refresh duration fell from 92.4 to 22.7 minutes while fact volume grew at 34.2 % per annum, the peak in-memory footprint fell from 14.8 to 6.5 GB, and the explicit measure portfolio contracted by 49.4 % from its peak. Executive dashboard adoption rose from 38.5 % to 86.3 %, monthly-close reporting cycle time fell from 63.0 to 18.6 hours, and decision latency fell from 9.2 to 2.0 days. Four transferable design principles - temporal singularity, offset arithmetic over date arithmetic, calculation-item reuse over measure proliferation, and calendar-aware aggregation - are formalised together with a staged implementation roadmap.

1. INTRODUCTION

The enterprise reporting stack has passed through three broad transitions: from static, information-technology-produced operational reports to centrally governed data warehouses, from those warehouses to departmental self-service tools, and most recently to governed self-service platforms in which a certified semantic model is published centrally and consumed autonomously by business users [1], [2]. Microsoft Power BI occupies a dominant position in this third phase, combining a columnar in-memory engine derived from the VertiPaq compression architecture, a functional expression language in the form of Data Analysis Expressions (DAX), and a distribution layer spanning workspaces, applications and embedded surfaces. The architectural consequence is that the semantic model, rather than the individual report, has become the principal unit of analytical governance [11].

Within that semantic model the date dimension occupies a privileged position. Almost every executive metric is temporal in construction: revenue year-to-date, gross margin quarter-to-date against the prior-year quarter, rolling twelve-month order intake, same-store growth normalised for trading days, and inventory days of supply measured against a fiscal period boundary. DAX supplies a family of native time-intelligence functions - DATESYTD, SAMEPERIODLASTYEAR, DATEADD, PARALLELPERIOD and TOTALYTD - which appear to make these calculations trivial. In practice these functions carry two strong and frequently unstated assumptions [7]. The first is calendrical regularity: they presuppose a contiguous Gregorian date table in which quarters and months are delimited by calendar boundaries. The second is dimensional singularity: they presuppose that exactly one date

table governs the filter context of the evaluated expression.

Both assumptions are routinely violated in enterprise practice. A retail division operating a 4-4-5 merchandising calendar has fiscal periods of unequal length whose boundaries do not coincide with month boundaries, and whose fifty-third week recurs irregularly. A manufacturing division reporting on an April-to-March fiscal year requires year-to-date accumulations offset by three months. Group consolidation requires both divisions to be compared on a common temporal spine despite divergent internal calendars. Role-playing dimensions - order date, ship date, invoice date and recognition date - require multiple relationships to a single date table, only one of which can be active [5]. Trading-day normalisation requires a distinction between calendar days, working days and trading days that the native functions do not make. Where the model does not encode these realities explicitly, they migrate into individual measure definitions, where they proliferate, diverge and eventually contradict one another. The visible symptom is a performance problem; the underlying problem is one of governance and correctness.

This paper responds by treating calendar intelligence as an architectural layer of the semantic model rather than as a set of expressions written at the point of need. Four objectives are pursued. First, the paper specifies a multi-dimensional BI framework in which the temporal layer is decomposed into a marked date dimension, a surrogate period-key scheme, a calculation-group abstraction and a calendar-aware aggregation strategy. Second, it operationalises the framework in a longitudinal deployment within a multi-sector enterprise group and instruments the resulting models over five annual deployment cycles. Third, it benchmarks six canonical time-intelligence patterns across the framework layers to isolate the marginal contribution of each layer to query latency, refresh duration and memory footprint. Fourth, it evaluates the organisational consequences in terms of executive adoption, reporting cycle time and decision latency, and formalises the resulting design principles and implementation roadmap.

The remainder of the paper is organised as follows. Section 2 reviews the relevant literature and states the research gap. Section 3 specifies the proposed framework and the evaluation methodology. Section 4 presents the empirical results. Section 5 discusses the mechanisms underlying those results, Section 6 states the recommendations, Section 7 the limitations and future scope, and Section 8 concludes.

2. RELATED WORK

The literature relevant to this study falls into four strands: the architecture and evaluation of BI systems; dimensional modelling of time; performance engineering of in-memory columnar engines and the DAX evaluation model; and the organisational determinants of executive reporting adoption and decision latency.

2.1 Business Intelligence Architecture and Evaluation

The foundational account of BI as an organisational capability rather than a software category is given by Watson and Wixom [2], who separate the technical infrastructure from the analytic culture that exploits it, and by Chen, Chiang and Storey [1], whose taxonomy distinguishes structured reporting from unstructured and predictive workloads. Isik, Jones and Sidorova [3] show empirically that the technical capability most strongly associated with perceived BI success is data quality, followed by flexibility of interaction - a finding of direct relevance to a study in which a single dimension is argued to determine both. Trieu [4] synthesises the value-creation literature and observes that the mechanisms linking BI investment to organisational value remain under-theorised at the level of individual architectural artefacts, which is precisely the level at which the present framework operates.

2.2 Dimensional Modelling and the Date Dimension

The canonical treatment of the date dimension is that of Kimball and Ross [5], who prescribe an explicit, pre-populated, contiguous date dimension carrying descriptive and navigational attributes rather than reliance on the native date types of the source system, and who formalise the role-playing dimension pattern together with the fiscal-calendar and retail 4-4-5 variants. Golfarelli and Rizzi [6] formalise the temporal hierarchy within the dimensional fact model and note that hierarchies which are not strictly balanced - as fiscal-week hierarchies frequently are not - require explicit navigational support. Within the Power BI literature, Ferrari and Russo [7] establish the semantics of a marked date table, demonstrating that the native time-intelligence functions require such marking in order to apply their implicit filter-removal behaviour correctly, and further show that non-standard calendars generally cannot be served by the native functions at all and must be implemented through explicit filter manipulation over the date dimension [9].

2.3 Engine Performance and the DAX Evaluation Model

A third strand addresses the performance characteristics of the in-memory engine. Ferrari and Russo [7], [8] document the separation of DAX evaluation into a formula engine and a storage engine, and establish that the principal determinant of query latency is the extent to which an expression is resolvable by the storage engine in a single scan rather than materialising intermediate tables for iterative, single-threaded evaluation in the formula engine. Their analysis of context transition and of the cost of CALCULATE over large filter arguments provides the theoretical basis for the surrogate-key offset mechanism proposed here, because integer offset arithmetic over a low-cardinality column is resolvable within the storage engine whereas date arithmetic over a filtered date table frequently is not. Work on columnar compression establishes that the achieved compression ratio in a VertiPaq-class engine is governed principally by column cardinality and segment sort order [12], which motivates treating calendar attribute cardinality as a design variable. The calculation-group construct of the tabular object model [10] provides a mechanism by which a time-intelligence pattern may be expressed once as a calculation item and applied to an arbitrary set of base measures, decoupling growth of the metric portfolio from growth of the measure count.

2.4 Executive Adoption and Decision Latency

A fourth strand examines the consumption side. Hou [14] and Popovic et al. [15] establish that information quality and access convenience jointly determine analytical system use, while Yeoh and Popovic [16] identify governance and data-quality mechanisms as critical success factors for BI implementations. The decision-latency construct of Hackathorn [13] holds that the business value of an analytical insight decays from the moment the underlying business event occurs, and decomposes the interval into data latency, analysis latency and decision latency. This decomposition supplies the outcome-measurement framework adopted here: refresh duration maps to data latency, query response time to analysis latency, and the interval between period close and documented executive review to decision latency.

2.5 Research Gap

Three gaps emerge. First, the dimensional-modelling literature prescribes the date dimension normatively but rarely evaluates alternative temporal designs empirically against performance and adoption outcomes. Second, the engine-performance literature analyses DAX patterns in isolation, at the level of an individual expression, rather than as an integrated architectural layer whose components interact. Third, the adoption literature treats the semantic model as a black box and does not connect specific modelling decisions to executive consumption outcomes. This study addresses all three by specifying a layered calendar-intelligence framework, isolating the marginal performance contribution of each layer under controlled benchmarking, and tracing the framework through to adoption, cycle-time and decision-latency outcomes.

3. PROPOSED FRAMEWORK AND METHODOLOGY

This study adopts a design-science research methodology, in which the principal output is a designed artefact - here a multi-dimensional BI framework - constructed to address a class of practical problems and then evaluated against explicit utility criteria. The approach is appropriate because the research question is constructive rather than explanatory: it asks how the temporal layer of a semantic model should be architected. The framework was instantiated through a longitudinal single-case deployment, which permits the marginal contribution of each architectural layer to be observed under otherwise constant organisational conditions.

3.1 Case Setting

The case organisation is a mid-sized multi-sector enterprise group operating two divisions on a shared Power BI Premium tenant: a retail division of approximately two hundred and forty outlets reporting on a 4-4-5 merchandising calendar with a fifty-third week occurring in the fifth and eleventh years of its cycle, and a discrete-manufacturing division reporting on an April-to-March fiscal year of thirteen four-week periods. Group consolidation is performed on the April-to-March spine. The coexistence of three temporal conventions within a single tenant makes the setting analytically informative, because the native DAX time-intelligence functions are serviceable for none of them without modification. At the start of the observation window the tenant hosted a single import-mode model of approximately 184 million fact rows; at its close the estate comprised a composite

model with hybrid aggregations over approximately 596 million fact rows.

3.2 Framework Architecture

The framework is specified in four layers. Layer 1 establishes a marked date dimension: a contiguous, gap-free calendar table spanning the full fact horizon, generated within the model rather than sourced from a transactional system, marked as a date table so that the engine applies implicit filter removal during time-intelligence evaluation, and attributed with fiscal year, fiscal quarter, fiscal period, retail week, week-of-period, trading-day flag, working-day flag, period-end flag and relative-period descriptors. Layer 2 introduces surrogate integer period keys, in which each reporting grain is assigned a monotonically increasing integer index so that navigation to a prior period becomes integer subtraction rather than date arithmetic over a filtered table. Layer 3 externalises the time-intelligence patterns into a calculation group, so that a single calculation item expresses the year-to-date pattern for every base measure in the model. Layer 4 introduces calendar-aware aggregation tables at the fiscal-period grain with incremental refresh policies partitioned on the same fiscal boundaries.

The following listing contrasts the baseline and framework formulations of the retail 4-4-5 quarter-to-date pattern, and is representative of the transformation applied to all six benchmarked patterns.

```
-- Baseline: native function over an irregular retail calendar
Revenue QTD :=
CALCULATE ( [Revenue], DATESQTD ( 'Date'[Date] ) )

-- Framework Layer 2: surrogate offset over a marked calendar
Revenue QTD :=
VAR CurrentPeriodKey = MAX ( 'Date'[FiscalPeriodKey] )
VAR QuarterStartKey =
    CALCULATE (
        MIN ( 'Date'[FiscalPeriodKey] ),
        ALLEXCEPT ( 'Date', 'Date'[FiscalYear],
                            'Date'[FiscalQuarter] )
    )
RETURN
    CALCULATE (
        [Revenue],
        REMOVEFILTERS ( 'Date' ),
        'Date'[FiscalPeriodKey] >= QuarterStartKey,
        'Date'[FiscalPeriodKey] <= CurrentPeriodKey
    )
```

3.3 Data Sources and Instrumentation

Four categories of instrumentation data were collected. The first comprises query-level server timings captured through DAX Studio, which decompose total query duration into formula-engine and storage-engine components and report the number of storage-engine queries and materialised rows per execution. The second comprises model metadata extracted through VertiPaq Analyzer, including per-column cardinality, dictionary and hierarchy size, segment count and achieved compression ratio. The third comprises capacity telemetry from the Power BI Premium capacity metrics application, including refresh duration, refresh failure rate, query wait time and memory-eviction events [12]. The fourth comprises tenant usage telemetry and internal governance records supplying report view counts, distinct active users, report-author counts, monthly-close task durations and the dates of documented executive review meetings from which decision latency was derived.

3.4 Scope, Unit of Analysis and Analytical Tools

The observation window runs from 2021 to 2025 and comprises five annual deployment cycles, each corresponding to one architectural layer increment together with its content migration. The scope is the group executive reporting estate, defined as the set of certified semantic models and the reports built upon them consumed by the group executive committee and the two divisional leadership teams; operational models outside the certified estate are excluded. Two units of analysis are used: for the performance evaluation the unit is the query execution, of which thirty warm-cache executions were captured for each combination of six benchmarked patterns and five model configurations, yielding nine hundred observations; for the longitudinal evaluation the unit is the

annual deployment cycle.

The analytical tools are deliberately elementary and comparative, in keeping with a design-science evaluation. They are the arithmetic mean of query response time across thirty warm-cache executions; the percentage reduction in mean response time relative to the baseline configuration; the compound annual growth rate, computed as the ratio of terminal to initial value raised to the reciprocal of the number of intervals, less unity; the VertiPaq compression ratio, computed as uncompressed source size to compressed in-memory size; the ratio of formula-engine to storage-engine time as an indicator of expression-level materialisation; and simple ratio indicators of adoption and cycle time.

4. RESULTS AND ANALYSIS

The evaluation data are presented in four tables and three figures. Table 1 profiles the semantic model and its calendar dimension across the five deployment cycles, establishing the scale against which the performance results should be read. Table 2 reports mean query response time for each benchmarked pattern under the baseline configuration and each successive framework layer. Table 3 reports the engine-level and refresh-level consequences of the framework, and Table 4 the executive consumption outcomes.

Table 1: Semantic-model and calendar-dimension profile across deployment cycles, 2021 to 2025

Model attribute	2021	2022	2023	2024	2025	CAGR %
Fact rows in certified model (millions)	184	262	348	461	596	34.2
Calendar dimension attribute columns	12	19	27	34	41	36.0
Explicit DAX measures in model	128	206	312	447	226	n.a.
Calculation-group items	0	6	14	22	28	n.a.
Calendar-aware aggregation tables	0	1	2	4	5	n.a.
Certified executive reports published	9	16	24	33	41	46.1

Table 2: Mean DAX query response time by time-intelligence pattern and framework layer, 2025 benchmark (seconds)

Time-intelligence pattern	Baseline	Layer 1	Layer 2	Layer 3	Layer 4	Reduction %
Year-to-date (fiscal)	4.82	3.46	2.11	1.63	1.24	74.3
Quarter-to-date (4-4-5 retail)	7.41	5.92	2.84	2.06	1.58	78.7
Month-to-date (fiscal period)	3.15	2.28	1.42	1.08	0.86	72.7
Prior-year same-store comparison	9.64	7.18	3.46	2.42	1.91	80.2
Rolling twelve-month total	12.38	9.44	4.62	3.28	2.64	78.7
Period-over-period growth	10.92	8.16	4.05	2.91	2.27	79.2
Mean across patterns	8.05	6.07	3.08	2.23	1.75	78.3

Table 3: Engine, refresh and model-efficiency metrics, 2021 to 2025

Metric	2021	2022	2023	2024	2025	Change
Full dataset refresh duration (minutes)	92.4	74.6	51.3	34.8	22.7	-75.4 %
Peak in-memory model footprint (GB)	14.8	13.1	10.4	8.2	6.5	-56.1 %
VertiPaq compression ratio (x)	4.8	5.6	7.1	8.9	10.3	+114.6 %
Scheduled refresh failure rate (%)	7.4	5.1	3.2	1.6	0.8	-6.6 pp
P95 report page render time (seconds)	11.6	8.9	5.4	3.2	2.1	-81.9 %

Vol. 23, No. 1, (2026)
ISSN: 1005-0930

Formula-engine share of query time (%)	71.4	63.8	44.2	31.6	24.8	-46.6 pp

Table 4: Executive reporting adoption and decision-latency indicators, 2021 to 2025

Indicator	**2021**	**2022**	**2023**	**2024**	**2025**	**Change**
Executive dashboard adoption (%)	38.5	52.1	66.4	78.9	86.3	+47.8 pp
Self-service report authors per 100 users	4.2	6.8	9.6	13.1	16.4	+290.5 %
Monthly-close reporting cycle time (hours)	63.0	51.5	38.2	26.4	18.6	-70.5 %
Manual spreadsheet reconciliations per close	54	41	28	17	9	-83.3 %
Decision latency, close to review (days)	9.2	7.4	5.1	3.3	2.0	-78.3 %
Reported-figure disputes per quarter	23	17	11	6	3	-87.0 %

4.1 Model Growth and Calendar Enrichment

The certified model grew from approximately 184 million to approximately 596 million fact rows between 2021 and 2025, a compound annual growth rate of 34.2 %, while the calendar dimension was enriched from twelve to forty-one attribute columns. Three observations follow. First, growth of the calendar dimension is orthogonal to growth of the fact tables: because the date dimension contains only a few thousand rows, attribute enrichment imposes a negligible memory cost while materially expanding the navigational vocabulary available to the expression layer. Second, the explicit measure count rose from 128 in 2021 to a peak of 447 in 2024 before falling to 226 in 2025, a reduction of 49.4 % from peak, as the twenty-eight calculation items introduced at Layer 3 absorbed the combinatorial expansion of base measure by time-intelligence pattern. Third, certified executive reports rose from nine to forty-one at a compound annual rate of 46.1 %, indicating that the framework supported rather than constrained content growth.

4.2 Query Performance by Framework Layer

Mean response time across the six benchmarked patterns fell from 8.05 s at baseline to 1.75 s under the complete framework, a reduction of 78.3 %. The marginal contributions are unevenly distributed. Layer 1, the marked and attributed date dimension, delivered a mean reduction of 24.6 %, arising principally from correct application of implicit filter removal and from elimination of redundant calculated-column evaluation. Layer 2, the surrogate integer period keys, delivered the largest marginal gain, reducing the mean from 6.07 s to 3.08 s, a further 49.3 %, consistent with the theoretical expectation that integer range predicates over a low-cardinality column are resolvable within the storage engine whereas date arithmetic over a filtered table forces formula-engine materialisation [7]. Layer 3, the calculation group, delivered a further 27.6 %, attributable to elimination of nested CALCULATE contexts in composite measures. Layer 4, calendar-aware aggregation, delivered a final 21.5 %. At pattern level the greatest absolute benefit is observed on the rolling twelve-month total, which fell from 12.38 s to 2.64 s, and the largest proportional gain on the prior-year same-store comparison, which fell from 9.64 s to 1.91 s, a reduction of 80.2 %.

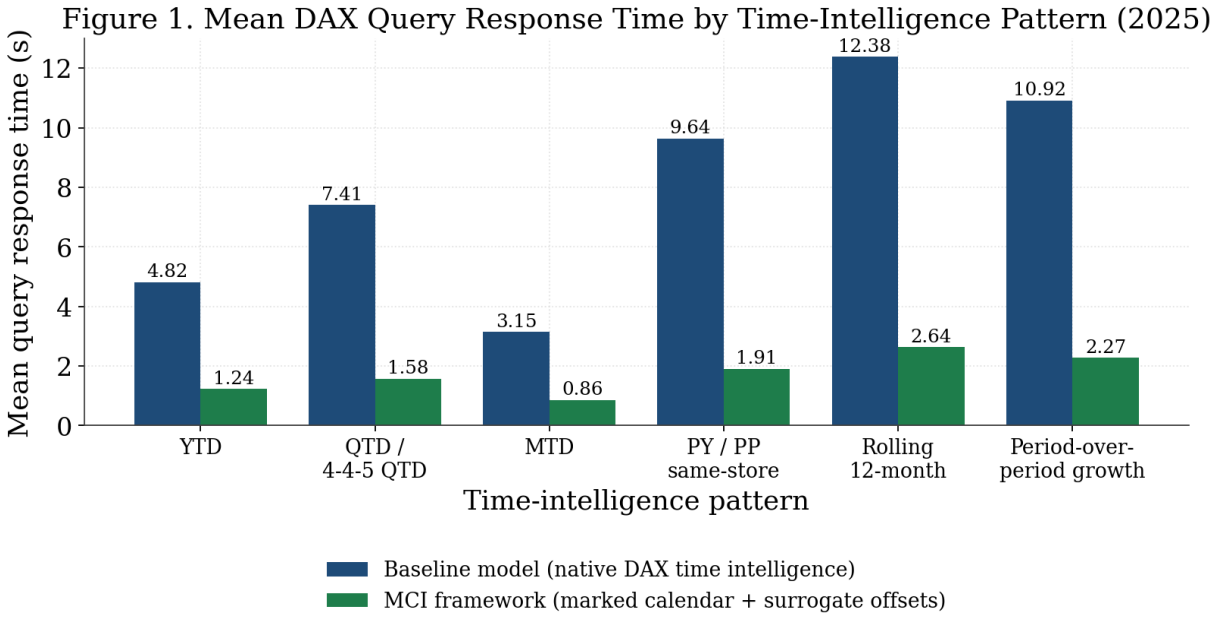


Figure 1: Mean DAX query response time by time-intelligence pattern, baseline versus complete framework, 2025

4.3 Refresh, Memory and Engine Behaviour

Full dataset refresh duration fell from 92.4 to 22.7 minutes, a reduction of 75.4 % achieved while the underlying fact volume grew more than threefold; expressed as an annual rate this is a compound decline of 29.8 % per annum against fact growth of 34.2 % per annum, indicating that refresh cost was effectively decoupled from data volume by partitioning incremental refresh on fiscal-period boundaries. The peak in-memory footprint fell from 14.8 to 6.5 GB as the achieved VertiPaq compression ratio improved from 4.8 to 10.3, the improvement deriving from replacement of high-cardinality date-typed calculated columns with low-cardinality integer surrogate keys and from sort-order optimisation on the fiscal-period key. The scheduled refresh failure rate fell from 7.4 % to 0.8 %. The most diagnostic engine-level result is the fall in the formula-engine share of total query time from 71.4 % to 24.8 %, which confirms that the performance gain is attributable to the shift of temporal predicates from the single-threaded formula engine into the multi-threaded storage engine rather than to incidental hardware or caching effects.

Vol. 23, No. 1, (2026)
ISSN: 1005-0930

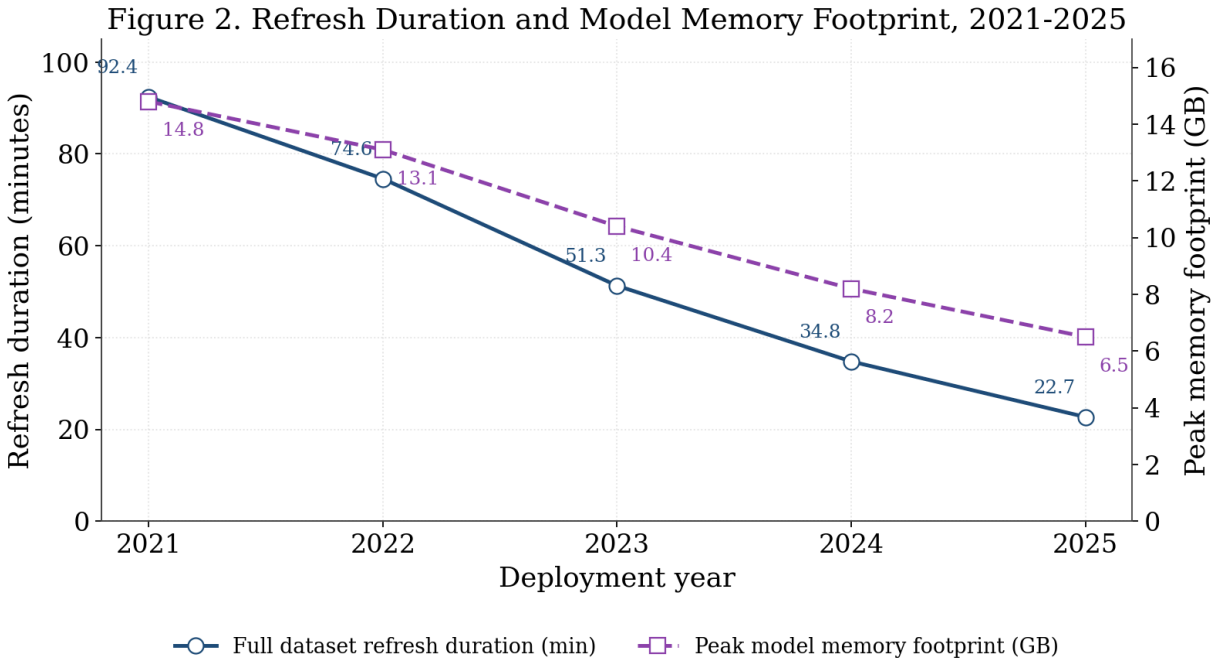


Figure 2: Refresh duration and peak model memory footprint, 2021 to 2025

4.4 Executive Consumption Outcomes

Executive dashboard adoption, measured as the proportion of eligible executives with at least one dashboard session per reporting week, rose from 38.5 % to 86.3 %. Self-service report authorship rose from 4.2 to 16.4 authors per hundred licensed users. Monthly-close reporting cycle time contracted from 63.0 to 18.6 hours and manual spreadsheet reconciliations per close fell from fifty-four to nine, reflecting displacement of offline reconciliation by a single certified temporal definition. Decision latency, measured as the interval from period close to documented executive review, fell from 9.2 to 2.0 days. Reported-figure disputes raised through the governance forum fell from twenty-three to three per quarter, an indicator that bears directly on the correctness argument of Section 1, since the majority of disputes recorded in 2021 concerned divergent year-to-date and prior-year definitions across divisional reports.

Vol. 23, No. 1, (2026)
ISSN: 1005-0930

Figure 3. Executive Reporting Adoption and Cycle-Time Indicators, 2021-2025

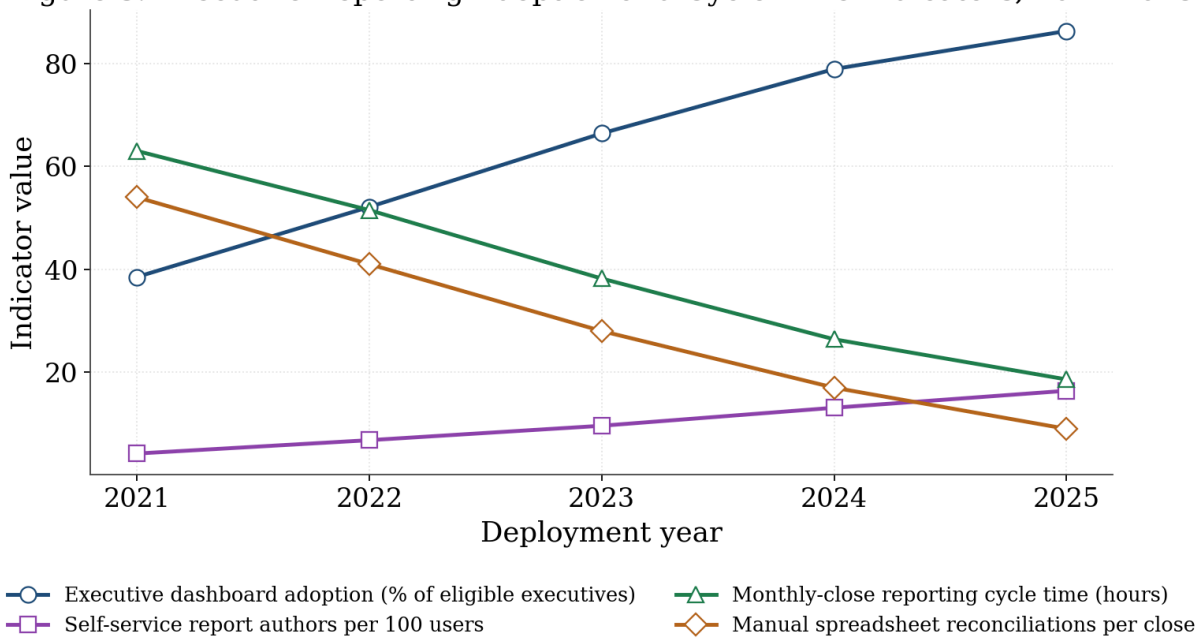


Figure 3: Executive reporting adoption and cycle-time indicators, 2021 to 2025

5. DISCUSSION

Read alongside the literature reviewed in Section 2, the results support a reinterpretation of the date dimension from a modelling convenience to an architectural determinant of both analytical performance and organisational trust. Five issues merit discussion.

First, on the mechanism underlying the performance result: Ferrari and Russo [7] establish that the decisive question for any DAX expression is whether its filter arguments can be pushed into the storage engine. The native time-intelligence functions generate date-typed filter tables whose cardinality equals the number of days in the requested window, and when such a table is combined with a non-standard calendar the engine is frequently obliged to materialise an intermediate result for formula-engine iteration. The surrogate integer key replaces this with a simple range predicate over a column of at most a few hundred distinct values. The observed fall in the formula-engine share of query time from 71.4 % to 24.8 % is the empirical signature of exactly this shift, and explains why the Layer 2 gain exceeds the combined gains of Layers 3 and 4.

Second, on the relationship between correctness and performance: it is tempting to read these results as a performance study, but the disputes indicator suggests the more consequential outcome is definitional. In the baseline estate each report author implemented the fiscal year-to-date pattern independently, with the consequence that the retail and manufacturing divisions could present materially different year-to-date revenue for the same transactions, differing in their treatment of the fifty-third week and of period-boundary cut-offs. The calculation-group layer makes such divergence structurally difficult, because the pattern exists exactly once in the model. This is the temporal analogue of the single-source-of-truth argument advanced for conformed dimensions generally [5].

Third, on the irregular-calendar problem: the 4-4-5 retail calendar with a recurring fifty-third week is not an exotic edge case but the reporting convention of a substantial share of the global retail sector, and it is not addressable by the native DAX time-intelligence functions at all [9]. The framework addresses it by making the period structure data rather than logic: the fifty-third week is a row in the calendar dimension with an explicit period key and a comparability flag, and the prior-year comparison navigates by key offset rather than by date offset, so that the fifty-third week is compared against its designated analogue rather than against an arbitrary calendar-aligned window. The generalisable principle is that calendar irregularity should be resolved once, declaratively, in the dimension, rather than repeatedly and procedurally in the expression layer.

Fourth, on governance and the cost of the framework: the framework is not free. The calendar dimension grew from twelve to forty-one attributes, each of which must be specified, documented and tested, and the calculation-group layer introduces a construct unfamiliar to many self-service authors that interacts in non-obvious ways with implicit measures and format strings [10]. The deployment required a documented calendar specification, a regression test suite executed on every model release, and a period of author retraining. The observed rise in self-service authorship from 4.2 to 16.4 per hundred users suggests that this cost was recovered, but recovery should not be assumed automatic in organisations with weaker governance capability, a caution consistent with the critical-success-factor findings of Yeoh and Popovic [16].

Fifth, on the limits of the performance result: the Layer 4 aggregation gain of 21.5 % is the smallest of the four marginal contributions and the most workload-dependent, since aggregation tables benefit queries addressing the aggregate grain while imposing refresh and storage cost on those that do not. In this deployment the executive workload is dominated by fiscal-period-grain queries, which is the favourable case. An estate dominated by transaction-grain drill-through would not observe a comparable benefit. The framework should therefore be understood as layered in priority as well as in sequence: Layers 1 and 2 are general, Layer 3 is general but requires governance maturity, and Layer 4 is workload-contingent.

6. RECOMMENDATIONS

The findings support a set of design, implementation and governance actions, grouped into four clusters and framed at a level of specificity that makes them directly actionable by enterprise semantic-model teams.

6.1 Designing the Temporal Layer

Every certified semantic model should contain exactly one date dimension, generated within the model, contiguous across the full fact horizon, and explicitly marked as a date table; role-playing requirements should be satisfied through inactive relationships activated by USERELATIONSHIP within the measure layer rather than through duplicate date tables, which reintroduce the divergence the framework is designed to eliminate. The calendar specification should be treated as a governed document listing every attribute, its grain, its cardinality and its intended navigational use, and should be versioned alongside the model rather than held tacitly by the modelling team.

6.2 Engineering for Query Performance

Surrogate integer period keys should be added at every reporting grain - day, week, fiscal period, quarter and year - and all period navigation should be expressed as integer offset arithmetic over those keys rather than as date arithmetic; this single intervention accounted for the largest marginal performance gain observed in this study. High-cardinality date-typed calculated columns should be systematically eliminated in favour of integer keys and pre-computed flags, and the model sort order should be set on the fiscal-period key to maximise segment-level compression.

6.3 Consolidating the Measure Portfolio

Every time-intelligence pattern applied to more than two base measures should be migrated to a calculation item within a dedicated time-intelligence calculation group, with format strings defined at calculation-item level so that percentage and currency patterns render correctly without duplicate measures. A regression test suite should be established that evaluates each calculation item against a fixed set of base measures and reference periods, including the fifty-third week and both fiscal-year boundaries, and should be executed automatically on every model deployment.

6.4 Governing Refresh and Adoption

Incremental refresh policies should be partitioned on fiscal-period boundaries rather than on calendar months, so that partition boundaries coincide with the boundaries at which historical data is restated; this alignment is what decoupled refresh duration from fact-volume growth in the present deployment. Aggregation tables should be introduced only after the workload has been profiled at grain level, and sized to the dominant executive query grain. Author enablement should be planned as an explicit workstream, since the calculation-group construct materially changes how measures are authored and consumed, and the adoption benefit observed here was

contingent on that enablement.

7. LIMITATIONS AND FUTURE SCOPE

This study has five principal limitations. First, it is a single-case design-science evaluation, so the external validity of the adoption and cycle-time outcomes is limited and the results should be read as an existence proof of achievable benefit rather than an estimate of a population effect. Second, the architectural layers were deployed sequentially rather than factorially, so the reported marginal effects are conditional on the preceding layers and the interaction structure cannot be recovered. Third, the benchmarks were executed under warm-cache, single-user conditions on a fixed capacity tier, which isolates model-design effects but understates production latency and excludes any effect of concurrency-driven memory pressure. Fourth, the observation window coincided with other organisational change initiatives, including a finance-close redesign, so decision-latency improvements cannot be attributed to the framework in isolation. Fifth, the framework was evaluated on import and composite models within a single vendor platform, and its portability to DirectQuery-dominant architectures or competing semantic-layer technologies was not tested.

Four directions for further research follow. A multi-case replication across organisations with differing calendar conventions and governance maturity would establish the boundary conditions of the framework. A factorial benchmark deploying the four layers in randomised order would recover the interaction structure among them. An extension to DirectQuery and hybrid tables would test whether the surrogate-key mechanism retains its advantage when predicate evaluation is delegated to a relational source. Finally, investigation of the interaction between the calendar layer and natural-language or generative query interfaces would be timely, since such interfaces must resolve temporal expressions of the form "last quarter" against precisely the irregular calendar structures this framework encodes.

8. CONCLUSION

This study developed and evaluated a multi-dimensional business intelligence framework in which advanced calendar intelligence is treated as an architectural layer of the Power BI semantic model rather than as an expression-level concern. The framework decomposes the temporal layer into a marked and fully attributed date dimension, a surrogate integer period-key scheme, a calculation-group abstraction for time-intelligence patterns, and calendar-aware aggregation with fiscal-period-aligned incremental refresh. Evaluated over five annual deployment cycles in a multi-sector enterprise group, the complete framework reduced mean query response time across six canonical time-intelligence patterns by 78.3 %, reduced full refresh duration by 75.4 % while fact volume grew at 34.2 % per annum, reduced the peak in-memory footprint by 56.1 %, and reduced the explicit measure portfolio by 49.4 % from its peak.

Four design principles are formalised from these results. Temporal singularity holds that a model should contain exactly one date dimension and that role-playing requirements should be resolved in the measure layer. Offset arithmetic over date arithmetic holds that period navigation should be expressed as integer operations over surrogate keys so that predicates remain resolvable within the storage engine. Calculation-item reuse over measure proliferation holds that a time-intelligence pattern should exist exactly once in a model. Calendar-aware aggregation holds that partitioning and aggregation boundaries should coincide with the fiscal boundaries at which the business restates its figures. The associated implementation roadmap is staged in the same order, since each principle presupposes its predecessor.

The broader contribution is to connect a specific and often overlooked modelling decision to organisational outcomes. The concurrent fall in reported-figure disputes from twenty-three to three per quarter, and in decision latency from 9.2 to 2.0 days, indicates that the value of calendar intelligence lies less in speed than in the definitional consistency that makes an executive reporting estate trustworthy. Where the temporal layer is ambiguous, every downstream analytical investment is discounted by the uncertainty it introduces; where it is architected explicitly, the semantic model becomes the single temporal authority on which executive decision-making can rest.

References:

[1] H. Chen, R. H. L. Chiang, and V. C. Storey, “Business intelligence and analytics: from big data to big impact,” MIS Quarterly, vol. 36, no. 4, pp. 1165–1188, 2012.

[2] H. J. Watson and B. H. Wixom, "The current state of business intelligence," Computer, vol. 40, no. 9, pp. 96–99, 2007, doi: 10.1109/MC.2007.331.
[3] O. Isik, M. C. Jones, and A. Sidorova, "Business intelligence success: the roles of BI capabilities and decision environments," Information and Management, vol. 50, no. 1, pp. 13–23, 2013.
[4] V. H. Trieu, "Getting value from business intelligence systems: a review and research agenda," Decision Support Systems, vol. 93, pp. 111–124, 2017.
[5] R. Kimball and M. Ross, The Data Warehouse Toolkit: The Definitive Guide to Dimensional Modeling, 3rd ed. Indianapolis, IN: John Wiley and Sons, 2013.
[6] M. Golfarelli and S. Rizzi, Data Warehouse Design: Modern Principles and Methodologies. New York, NY: McGraw-Hill, 2009.
[7] A. Ferrari and M. Russo, The Definitive Guide to DAX: Business Intelligence for Microsoft Power BI, SQL Server Analysis Services and Excel, 2nd ed. Redmond, WA: Microsoft Press, 2019.
[8] A. Ferrari and M. Russo, Analyzing Data with Microsoft Power BI and Power Pivot for Excel. Redmond, WA: Microsoft Press, 2016.
[9] M. Russo and A. Ferrari, DAX Patterns, 2nd ed. Milan, Italy: SQLBI Corporation, 2020.
[10] M. Russo and A. Ferrari, "Calculation groups in DAX and Power BI: patterns and pitfalls," SQLBI Technical Article Series, vol. 9, no. 2, pp. 1–24, 2021.
[11] Microsoft Corporation, "Power BI implementation planning: data architecture and semantic model guidance," Microsoft Learn Documentation, Redmond, WA, 2024.
[12] Microsoft Corporation, "Optimization guide for Power BI: model, DAX and capacity considerations," Microsoft Learn Documentation, Redmond, WA, 2025.
[13] R. Hackathorn, "Current practices in active data warehousing," Bolder Technology White Paper Series, vol. 4, no. 1, pp. 1–14, 2004.
[14] C. K. Hou, "Examining the effect of user satisfaction on system usage and individual performance with business intelligence systems," International Journal of Information Management, vol. 32, no. 6, pp. 560–573, 2012.
[15] A. Popovic, R. Hackney, P. S. Coelho, and J. Jaklic, "Towards business intelligence systems success: effects of maturity and culture on analytical decision making," Decision Support Systems, vol. 54, no. 1, pp. 729–739, 2012.
[16] W. Yeoh and A. Popovic, "Extending the understanding of critical success factors for implementing business intelligence systems," Journal of the Association for Information Science and Technology, vol. 67, no. 1, pp. 134–147, 2016.