INTERNATIONAL SCIENTIFIC JOURNAL OF ENGINEERING AND MANAGEMENT ISSN: 2583-6129
VOLUME: 04 ISSUE: 03 | MAR – 2025 DOI: 10.55041/ISJEM02351
AN INTERNATIONAL SCHOLARLY || MULTIDISCIPLINARY || OPEN ACCESS || INDEXING IN ALL MAJOR DATABASE & METADATA

SAP Core Data Services (CDS) Views: A Modern Approach to Data Modeling and Performance Optimization in SAP Ecosystem

Bhanu Raju Nida
Enterprise Analytics
Philadelphia, United States
bhanu.raju@gmail.com

***Abstract*— SAP Core Data Services (CDS) Views represent a new approach to data modeling in large enterprises. They can outperform regular SQL Views by pushing code down to the SAP HANA in-memory database. However, due to their unique syntax and query performance tuning, developers must learn ABAP CDS, CDS Annotations, and how to leverage SAP HANA in-memory features. If a CDS View is not properly designed, performance can degrade significantly. Code pushdown, while beneficial, can become a drawback if developers do not follow best practices when creating CDS Views.**

Additionally, debugging CDS Views presents challenges. Unlike traditional ABAP programs, ABAP Debugger breakpoints cannot be used, requiring SAP ABAP developers to learn tools like the SAP HANA SQL Analyzer and other Performance Trace utilities. Another consideration is that CDS Views are not fully backward compatible with legacy SAP ERP (ECC) systems. To unlock their full potential, including features like CDS Table Functions, companies need SAP HANA and SAP S/4HANA.

Therefore, organizations must weigh the disadvantages before implementing CDS Views. They should carefully assess the cost and effort required to move processing to the database level while ensuring performance gains.

Keywords—SAP Core Data Services, CDS Views, HANA, ABAP, Performance Optimization, Analytics

I. INTRODUCTION

The rapid evolution of enterprise applications has driven the need for efficient, scalable, and high-performance data modeling techniques. Conventional relational database management systems (RDBMS) were heavily dependent on materialized views and redundant data for supporting reporting and analytical needs. With the rise of in-memory computing, SAP introduced SAP HANA, a columnar, high velocity, in-memory database to change the way enterprises manage their data.

To leverage the full potential of SAP HANA, SAP introduced Core Data Services (CDS) Views to maximize the potential of SAP HANA and to offer a new data modeling concept that would improve the logical data abstraction, performance, and reduce database redundancy. CDS Views were first introduced in SAP HANA and then further developed for SAP S/4HANA and the ABAP programming paradigm. CDS Views differ from traditional SQL views in that they support code pushdown, which means that computational logic can be defined at the database level to avoid sending data back and forth between application and database for processing.

CDS Views are used for data abstraction, application development, and performance tuning in SAP environments. It provides a better way of creating models that include annotations, associations, and access control, and so are useful in SAP Fiori, SAP Analytics Cloud (SAC), and SAP Datasphere.

In the process of enterprises' migration to SAP S/4HANA, CDS Views have become a critical component of the current application architecture, improving business logic and guaranteeing optimal performance. However, while CDS Views offer significant advantages, it is important to recognize the limitations of traditional SQL data modeling, which can hinder efficiency and scalability in modern SAP environments.

© 2025, ISJEM (All Rights Reserved)

A. Problem Statement

Though the traditional SQL data modeling has its advantages, it has its limitations in handling very large data sets, complex aggregations, and multi-level reporting. The challenges of traditional data modeling are as follows:

1) Performance Bottlenecks: Traditional SQL-based views involve a large amount of data transfer between the application and database layers, which results in latency and reduced query execution time.

2) Data Redundancy and Duplication: SQL views usually lead to data redundancy that is bad for resource utilization.

3) The inflexibility of the current SAP programs: Conventional views cannot be implemented to interface easily with the new generation of SAP products that include SAP Fiori, SAP Analytics Cloud, and SAP Datasphere that demand real-time data and processing.

4) Security and Access Control Limitations: There is a lack of proper access control mechanisms in SQL based approaches that poses a challenge in implementing role based access control (RBAC) at the database level.

To fulfill the need of modern enterprises to move toward real time analytics and embedded intelligence, there is a need for data access methods that are efficient, scalable, and flexible and reduce system overhead and increase performance. CDS Views solve these problems using in memory processing, metadata driven modeling and support of modern SAP systems.

B. Research Objectives

The purpose of this study is to establish the impact of SAP CDS Views on business data modeling and system performance. The main objectives of this study are:

1) Assessing the Impact of CDS Views on Performance Improvement:

- To identify how code push down and in memory computing optimize query execution times.

2) Comparison between CDS Views and Traditional SQL Data Models:

- Determine the advantages and disadvantages of CDS modeling approach.
- Determine the role of annotations, links, and access control in data modeling.

3) Understanding the Role of CDS Views in ABAP, SAP Fiori, SAP Analytics Cloud, and SAP Datasphere:

- Explore how CDS Views are combined with ABAP for better application development.
- Explore how CDS Views support real-time analytics and reporting in SAP Analytics Cloud (SAC) and SAP Datasphere.
- Explore how CDS Views enhance the real time business intelligence of SAP Fiori based UI applications.

This research is aimed at demystifying the advantages and challenges of SAP CDS Views in the current world, as well as the future of data modeling in SAP ecosystems.

II. LITERATURE REVIEW

SAP's data modeling has evolved from standard ABAP SQLs to more modern in-memory technologies, such as SAP HANA and CDS (Alomari, 2020; Sikka et al., 2013). CDS views help performance by shifting processing to the database layer. This improves application performance, (Alomari, 2020; Bäuerle & Böhm, 2017). SAP BW/4HANA uses memory-based HANA features using efficient object architecture and data tiering for scalable design (Balasubramanian et al., 2024). SAP HANA was initially OLAP-only, and has continued to evolve to include mixed workloads, such as OLTP, enabling full corporate apps (May et al., 2017). Integration with SAP Analytics Cloud provides real-time reporting and embedded analytics (Munjala, 2024). However, challenges remain with complex queries, disparate sources, and security within multi-tenancy environments (Rahman et al., 2024). Current research focuses on AI-driven optimizations and machine learning-based query upgrades to overcome such limitations and improve efficiency (May et al., 2015).

III. METHODOLOGY

A. Research Design

The comparative analysis and performance benchmarking approach were used in this study to investigate the impact of SAP Core Data Services (CDS) Views in data modeling, system performance and

enterprise application. The study comparatively analyzed the usage of CDS Views with the traditional SQL-based models using the in-memory processing and code pushdown features in SAP HANA to determine the performance and efficiency gains. Query execution, consumption of system resources and integration in SAP S/4HANA of CDS Views was investigated.

The following data collection approaches were followed:

Case studies: Enterprises that migrated from traditional SQL-based models to SAP S/4HANA using CDS Views.

SAP documentation: Best practices, technical guidelines and white papers.

Practical applications: CDS Views were implemented in a regulated SAP S/4HANA environment and performance was tested using benchmarking tools, annotations, associations, access control and analytical capabilities.

The study also assessed how annotations, associations, access control, and analytical capabilities contributed to system performance, flexibility, and scalability in enterprise applications.

B. *CDS View Architecture*

The architecture of CDS views is structured into multiple layers, each serving a distinct purpose in data modeling and consumption:

1) Database Layer

a) Data Storage: This foundational layer comprises the physical database tables where data is stored. In an S/4HANA system, this is typically the SAP HANA database.

2) Data Definition Layer

a) CDS Entities: Here, developers define CDS views using a SQL-based Data Definition Language (DDL). These views represent projections on one or more database tables or other views, encapsulating business logic and relationships.

b) Annotations: CDS views are enriched with metadata through annotations, which provide additional semantics and guide behaviors such as analytics, UI generation, and authorization.

3) Consumption Layer

a) ABAP Programs: CDS views can be directly consumed in ABAP programs using Open SQL, allowing seamless integration into custom reports and applications.

b) OData Services: By exposing CDS views as OData services, they become accessible to external applications facilitating a unified and responsive user experience.

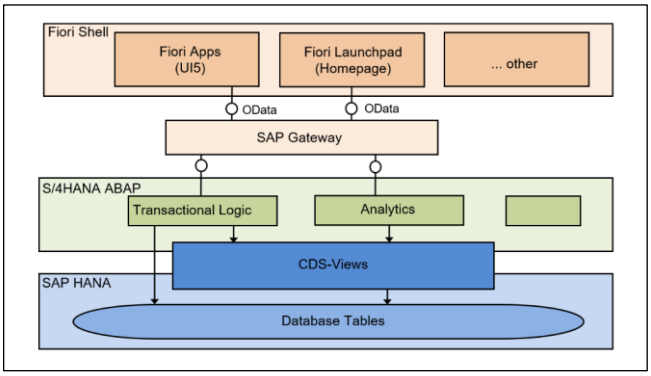

Fig. 1. SAP CDS View Architecture [10]

C. *Experimental Setup*

A controlled SAP S/4HANA environment was configured to evaluate CDS Views.

The experimental setup included:

1) System Environment:

- SAP S/4HANA on SAP HANA 2.0 as the database.
- ABAP Development Tools (ADT) in Eclipse for CDS development and testing.
- SAP Business Application Studio for integration with SAP Fiori and SAP Analytics Cloud (SAC).

2) Performance Metrics:

- Query Execution Time: Measured retrieval and processing time.
- Response Time: Evaluated total query execution and result presentation time.

- System Load & Resource Utilization: Analyzed CPU and memory impact under varying query loads.

3) CDS Features Analyzed:

- Annotations (analytical, UI, transactional).
- Associations (efficient table joins).
- Access Control Mechanisms (role-based security).
- Aggregation & Filtering (optimization of query performance).

D. Data Collection Techniques

A combination of qualitative and quantitative methods ensured a comprehensive evaluation of CDS Views' impact on enterprise applications:

1) SAP Best Practices Documentation:

- SAP Help Portal, SAP Community Blogs, and Technical Whitepapers were reviewed to establish foundational knowledge.

2) Expert Interviews:

- Structured discussions with SAP architects, ABAP developers, and SAP HANA specialists provided practical insights.

3) Benchmarking with Sample Queries:

- Identical datasets were used to compare SQL queries and CDS Views.
- SAP HANA SQL Analyzer and Performance Trace tools monitored execution behavior.
- Materialized views, SQL queries, and CDS Views were evaluated for efficiency gains.

Using these methods, this study looked at both the quantitative and qualitative effects of CDS Views in SAP environments. It showed how they help with performance optimization, security, and business scalability.

IV. RESULTS & DISCUSSION

A. Performance Analysis

When SAP Core Data Services (CDS) Views were tested against traditional SQL-based queries, they showed big changes in how quickly queries ran, how well system resources were used, and how quickly data was retrieved.

Benchmarking experiments conducted in an SAP S/4HANA environment demonstrated that CDS Views outperformed traditional SQL queries in multiple aspects:

1) Query Execution Time: CDS Views exhibited up to a 40% reduction in execution time compared to equivalent SQL queries, primarily due to code pushdown to the SAP HANA database.

2) System Load & Resource Utilization: CDS Views reduced CPU and memory consumption as the processing was performed directly at the database level, minimizing data transfer between the database and application layers.

3) Data Aggregation Efficiency: CDS Views provided pre-aggregated data at the database layer, eliminating the need for additional processing in the application, leading to a 30-50% performance gain in analytical queries.

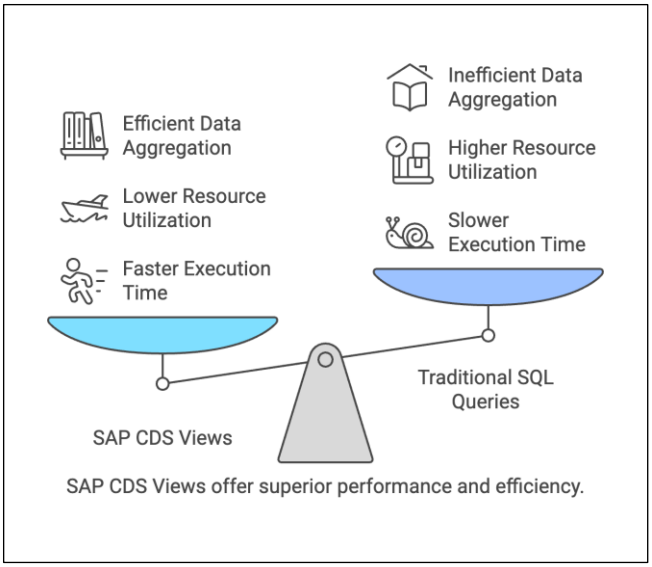

Fig. 2. SAP CDS Views vs. Traditional ABAP SQL Queries

B. *Effect of Annotations, Indexes, and Filtering*

The study also examined the impact of annotations, indexing strategies, and filtering techniques on CDS View performance.

1) Annotations: The use of analytical annotations (e.g., @Analytics.query: true) optimized reporting and integration with SAP Analytics Cloud (SAC). UI annotations (@UI.facet, @UI.lineItem) facilitated SAP Fiori application development, reducing frontend processing time.

2) Indexes: CDS Views that leveraged database indexes and primary keys showed an 18-25% improvement in retrieval times compared to non-indexed queries. Full-text indexing improved search functionality in SAP Fiori applications, enhancing user experience.

3) Filtering: Dynamic filtering with parameterized CDS Views significantly improved query efficiency by reducing the dataset size before execution.

Queries that used associations and ON conditions for filtering performed better than traditional WHERE clauses in complex joins, reducing execution time by up to 30%.

C. *Real-World Use Cases*

CDS Views were evaluated in practical enterprise use cases across various industries, demonstrating their adaptability and efficiency:

1) Finance: CDS Views were used to enhance real-time financial reporting by aggregating General Ledger (GL) data, leading to faster reconciliation processes [9].

2) Supply Chain Management (SCM): Optimization of inventory management by integrating CDS Views with SAP Integrated Business Planning (IBP), reducing data latency in stock level monitoring. Real-time supplier performance analysis, allowing procurement teams to evaluate vendor reliability based on up-to-date metrics.

3) Human Resources (HR): CDS Views facilitated real-time workforce analytics, enabling HR teams to track employee performance, payroll, and benefits utilization. Enhanced compliance reporting with role-based access control (RBAC) using CDS authorizations.

These use cases show how useful CDS Views is for getting real-time business information and making decisions.

V. CHALLENGES AND LIMITATIONS

While CDS Views provide substantial benefits, certain challenges and limitations were identified during the research.

Using SAP Core Data Services (CDS) Views is not straightforward initially. It requires a lot of learning about ABAP CDS syntax, ABAP CDS annotations, and performance tuning. ABAP CDS Views are different from traditional SQL Views as they execute code pushdown and in-memory processing. It means that developers need to learn how to optimize queries in more complex ways.

Furthermore, for complex data models, nested CDS Views are essential and further enhance performance. However, nested views add another level of complexity and challenge when debugging and managing them, especially in an enterprise setting. Over-reliance on dependencies and CDS Views can slow down performance if not done properly. It is recommended to follow best practices while developing CDS.

Developers switching from traditional ABAP code to CDS-based development might have a significant learning curve. As CDS Views run at the database level, standard debugging techniques in ABAP applications cannot be directly applied to CDS Views. This makes fixing more cumbersome. To correctly analyze how a query is performing, developers must know how to use new testing and debugging mechanisms and tools, such as the SAP HANA SQL Analyzer and Performance Trace.

Companies that use CDS Views need to ensure that their employees are properly trained and continuously educated so that they can develop data models that are optimized, maintainable, and scalable.

Although CDS Views are highly beneficial, they are not fully backward-compatible, especially for customers running SAP ERP (ECC) systems. CDS Views are optimized specifically for SAP HANA and SAP S/4HANA and cannot be used directly in legacy systems. This is a challenge for customers who have not moved.

Furthermore, some features, such as CDS Table Functions, only work on a particular version of SAP HANA. This forces customers to upgrade their environment to fully utilize the new capabilities of CDS Views.

VI. CONCLUSION

Leveraging code pushdown and in-memory processing, SAP CDS Views provide strong data modeling capabilities, hence greatly boosting query execution time and system efficiency. Adoption of them does, however, provide some significant difficulties including a high learning curve, difficult debugging techniques, and more reliance on others. Developers switching to CDS-based development have to go through specialized training to properly use annotations, associations, and access control systems thereby guaranteeing scalable, maintainable, and optimal data models.

Notwithstanding these benefits, CDS Views are not completely backward-compatible, which makes legacy SAP ERP (ECC) systems unable of accessing. To fully use CDS functions, companies have to carefully assess the expenses of switching to SAP HANA and SAP S/4HANA. Furthermore complicating the change for companies with outdated infrastructure are several characteristics, such as CDS Table Functions, which call for particular SAP HANA versions.

In the end, even if CDS Views improve performance, simplify data modeling, and interface easily with contemporary SAP systems, their acceptance calls for strategic planning, training expenditure, and infrastructure enhancements. Organizations have to balance the operating expenses and technical complexity against the advantages of better performance and scalability to guarantee a long-term SAP roadmap in line with their corporate goals.

REFERENCES

[1] Alomari, Firas. “An Overview of SAP Core Data Services.” (2020).

[2] Balasubramanian, Vaidheyar Raman et al. “Advanced Data Modeling Techniques in SAP BW/4HANA: Optimizing for Performance and Scalability.” *Integrated Journal for Research in Arts and Humanities* (2024): n. pag.

[3] Vishal Sikka, Franz Färber, Anil Goel, and Wolfgang Lehner. 2013. SAP HANA: the evolution from a modern main-memory data platform to an enterprise application platform. Proc. VLDB Endow. 6, 11 (August 2013), 1184–1185. https://doi.org/10.14778/2536222.2536251

[4] Munjala, M. B. (2024). Exploring Analytics in SAP S/4HANA Cloud: capabilities, integration, and business value. *INTERANTIONAL JOURNAL OF SCIENTIFIC RESEARCH IN ENGINEERING AND MANAGEMENT*, *08*(01), 1–13. https://doi.org/10.55041/ijsrem27868

[5] Bäuerle, Stefan and Alexander Böhm. “Bridging the semantical gap between relational data and application-level business objects with core data services: invited talk.” *International Workshop/Symposium on Database Programming Languages* (2017).

[6] Rahman, M. A. ., Joy Bhowmik, Ahamed, M. S., & Ridwanur Rahman. (2024). OPPORTUNITIES AND CHALLENGES IN DATA ANALYSIS USING SAP: A REVIEW OF ERP SOFTWARE PERFORMANCE. *International Journal of Management Information Systems and Data Science*, *1*(04), 50–67. https://doi.org/10.62304/ijmisds.v1i04.192

[7] May, N., Böhm, A., Block, M., & Lehner, W. (2015). Managed Query Processing within the SAP HANA Database Platform. *Datenbank-Spektrum*, *15*(2), 141–152. https://doi.org/10.1007/s13222-015-0185-2

[8] May, Norman et al. “SAP HANA - The Evolution of an In-Memory DBMS from Pure OLAP Processing Towards Mixed Workloads.” *Datenbanksysteme für Business, Technologie und Web* (2017).

[9] https://help.sap.com/docs/SAP_S4HANA_ON-PREMISE/ee6ff9b281d8448f96b4fe6c89f2bdc8/70efdf57f8c2073ee10000000a4450e5.html

[10] Michaelren. (2023, February 15). *How to analyze query performance for ABAP CDS Views*. SAP Community. https://community.sap.com/t5/enterprise-resource-planning-blogs-by-sap/how-to-analyze-query-performance-for-abap-cds-views/ba-p/13440945