SALT-KG: A Benchmark for Semantics-Aware Learning on Enterprise Tables

Isaiah Onando Mulang'
SAP SE
mulang.onando@sap.com

Felix Sasaki
SAP SE
felix.sasaki@sap.com

Tassilo Klein
SAP SE
tassilo.klein@sap.com

Jonas Kolk
SAP SE
jonas.kolk@sap.com

Nikolay Grechanov
SAP SE
nikolay.grechanov@sap.com

Johannes Hoffart
SAP SE
johannes.hoffart01@sap.com

Abstract

Building upon the SALT benchmark for relational prediction [Klein et al., 2024], we introduce SALT-KG, a benchmark for semantics-aware learning on enterprise tables. SALT-KG extends SALT by linking its multi-table transactional data with a structured **O**perational **B**usiness Knowledge represented in a Metadata **K**nowledge **G**raph (**OBKG**) that captures field-level descriptions, relational dependencies, and business object-types. This extension enables evaluation of models that jointly reason over tabular evidence and contextual semantics—an increasingly critical capability for foundation models on structured data. Empirical analysis reveals that while metadata-derived features yield modest improvements in classical prediction metrics, these metadata features consistently highlight gaps in models' ability to leverage semantics in relational context. By reframing tabular prediction as semantics-conditioned reasoning, SALT-KG establishes a benchmark to advance tabular FMs grounded in declarative knowledge, providing the first empirical step toward semantically linked tables in structured data at enterprise scale.[1]

1 Introduction

We introduce SALT-KG, a benchmark for *semantics-aware learning on enterprise tables*. The SALT dataset [Klein et al., 2024] established a standardized setting for learning multi-table enterprise data through attribute autocompletion tasks, but was limited to relational structure, focusing on how models infer missing attributes from transactional evidence. SALT-KG extends this foundation by linking the same relational schema to a structured OBKG that captures field-level semantics, declarative business knowledge, and hierarchical object types. Declarative metadata describes what entities and attributes *mean*, beyond how they are *joined*, enabling models to condition tabular prediction on contextual semantics instead of mere statistical correlations.

Despite remarkable progress in language, vision, and multimodal learning, tabular data—particularly multi-table, relational datasets typical of enterprise environments—remain among the most challenging modalities for machine learning [Grinsztajn et al., 2022, Bodensohn et al., 2024]. Recent advances in tabular foundation models, such as CARTE [Kim et al., 2024], TABPFN [Hollmann et al., 2025, 2023], TABICL [Qu et al., 2025], PORTAL [Spinaci et al., 2024] and CONTEXTTAB [Spinaci et al., 2025], have improved data efficiency and in-context reasoning. However, these models are typically trained and evaluated on benchmarks that represent relational structure but lack explicit semantic grounding or declarative context. Conversely, enterprise data encodes rich dependencies between attributes—such as *Shipping Point*, *Plant*, and *Payment Terms*—yet the semantics of these relationships are implicit, buried in metadata descriptions or business logic. As a result, even strong

[1] The data set, benchmark splits, and scripts to reproduce all results are publicly available at: https://github.com/SAP-samples/salt-kg.

AI for Tabular Data workshop at EurIPS 2025

tree-based or neural predictors rely on the memorization of local correlations rather than reasoning. Existing benchmarks such as RELBENCH [Robinson et al., 2024], TALENT [Liu et al., 2024], SALT [Klein et al., 2024], and TABARENA [Erickson et al., 2025] have advanced evaluation for structured prediction on relational data. Yet, they do not expose the metadata layer where domain meaning and declarative knowledge reside. In parallel, knowledge graph (KG) and data integration communities have explored connecting tables to semantic graphs through systems such as JENTAB [Abdelmageed et al., 2024], CORECOLUMNMATCH [Qiu et al., 2024], and LLM-based table-to-KG alignment methods [Vandemoortele et al., 2024]. These advances have yet to be systematically incorporated into tabular learning benchmarks, leaving a gap between relational representation learning and semantics-aware reasoning. SALT-KG bridges this gap by enriching enterprise relational data with an explicit semantic layer that links tables, fields, and business object types through declarative relationships in a coherent OBKG. This integration enables controlled evaluation of models that combine relational reasoning with declarative schema semantics for contextual understanding beyond feature-level learning. Empirical results show that metadata features yield only modest gains in classical prediction metrics but reveal consistent differences in the models' ability to exploit semantic alignment and contextual relationships. This work operationalizes the declarative layer of the broader *Foundation Models for Semantically Linked Tables (FMSLT)* framework [Klein and Hoffart, 2025], which envisions integrating declarative, procedural, and operational knowledge; SALT-KG focuses solely on the declarative dimension—capturing what entities and attributes *mean* through structured metadata rather than how they *behave* in processes or systems.

Related Work: The prediction of tabular data has traditionally been dominated by tree-based ensemble methods such as XGBoost [Chen and Guestrin, 2016], LightGBM [Ke et al., 2017], and CatBoost [Prokhorenkova et al., 2018]. Recently, this landscape has shifted with the emergence of deep learning approaches tailored to relational databases, including CARTE [Kim et al., 2024], AutoGluon [Erickson et al., 2020], and GraphSAGE [Hamilton et al., 2017], evaluated on benchmarks such as RelBench [Robinson et al., 2024] and TabArena [Erickson et al., 2025]. The rise of tabular foundation models—including transformer-based in-context learners such as TabPFN [Hollmann et al., 2025, 2023], TabICL [Qu et al., 2025], and ContextTab [Spinaci et al., 2025]—marks a significant shift toward generalizable tabular learning. In parallel, knowledge graphs (KGs) have gained prominence for their ability to contextualize data for deep learning [Mulang' et al., 2020]. Based on this, systems such as JenTab [Abdelmageed et al., 2024] have taken initial steps toward bridging the gap between raw tables and semantic graphs. Nevertheless, more progress is needed to ground tabular foundation models in operational business semantics [Klein and Hoffart, 2025], enabling models that jointly reason over relational evidence and contextual knowledge. SALT-KG is designed to catalyze this research direction by providing a benchmark to evaluate how contextual knowledge of KG can improve tabular learning and representation.

2 Dataset Design

Background: SALT-KG extends SALT [Klein et al., 2024] by enriching multi-table enterprise data with a structured layer of KG-metadata. The dataset captures a representative *sales-order creation process* from a real-world transactional system, where each order consists of a document header and multiple line items linked through foreign keys. This reflects enterprise data characteristics: multi-relational dependencies, temporal dynamics, categorical imbalance, and hierarchical structure.

Task Definition: The SALT-KG benchmark defines a suite of predictive tasks designed to assess *semantics-aware tabular learning*. Aligned with SALT setup [Klein et al., 2024], the benchmark simulates missing-field autocompletion in transactional records, to infer key attributes given partial table information. Twenty-one features are provided as inputs, and eight variables serve as multiclass targets, including sales indicators such as `SalesOffice`, `SalesGroup`, `CustomerPaymentTerms`, `ShippingCondition`, `Plant`, `ShippingPoint`, and both header- and item-level `IncotermsClassification`.

The OBKG data infused with the tabular SALT is obtained from an underlying RDF-Based Enterprise Metadata KG. The KG models the ontological concepts for business data, and instances of these concepts are represented in a hierarchical network beginning with metadata about the underlying tables (e.g., table names, descriptions, column descriptions, type, length) and other metadata such as property domains and ranges that define RDF classes to which these objects belong. For example, in the table "I_SALESDOCUMENT" will be represented as a node in the graph as an instance of the class for tables. The fields are independent one-hop nodes, allowing for more expressivity at

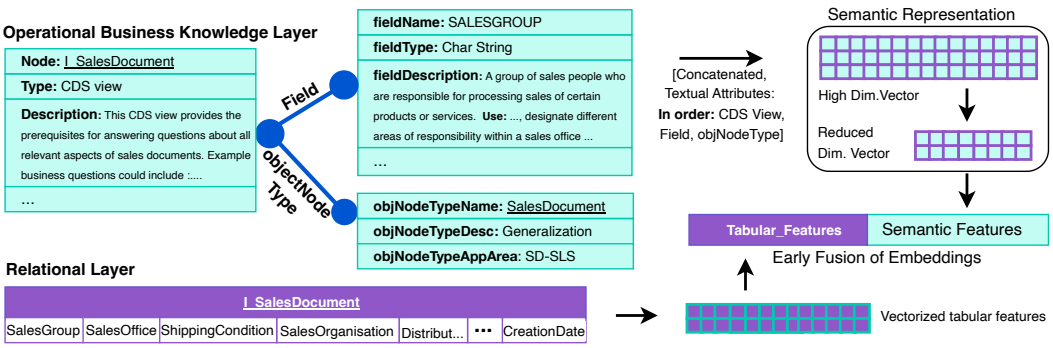

Figure 1: Construction of semantically linked tables. A sample OBKG (blue) aligned with its relational schema. Graph context is encoded and merged with tabular features.

the table and field levels. Further metadata includes value help data and the business application area from where the table has been derived. The next layer in the network involves inter-table relationships, specifically those involving foreign keys and joins within the network. Core Data Services Views (known as CDS VIEWS[2])are projections of other entities, like tables, to capture specific data needs. In the OBKG, these are data abstraction nodes with associated fields, labels, associations, data classes, reference fields, and other elements. Intrinsically, every underlying table has a matching view in the KG. To align OBKG with the underlying tabular data, metadata context is fetched using SPARQL queries in which KG nodes corresponding to each of the tables in SALT are subjects of the first triple patterns in the SPARQL basic graph pattern. Triple patterns for one-hop triples that link to two major business objects, namely: The Fields and the Object Node Types are fetched. Fig. 1 illustrates three types of interconnected information. First, the CDS Views provides data model abstractions (semantic definitions without actual tabular instances. Second, the fields augment the columns in the SALT table, e.g., I_SALESDOCUMENT in our example has a column ACCOUNTINGDOCEXTERNALREFERENCE and the OBKG triples provide descriptions or data type information for this column. column. All tabular columns have a corresponding field in the KG. Third, Object Node Types provide further semantic metadata through technical definitions, business object descriptions, and additional information such as the application area of the business object.

The overall dataset comprises exact tabular statistics in SALT [Klein et al., 2024]. These tables are normalized through foreign-key joins and flattened to the item level for supervised learning. Approximately 990 schema fields are explicitly mapped to corresponding tabular columns, along with their associated business object nodes and textual descriptions. Additionally, 1,954 semantic object types enrich the metadata layer. Connectivity varies across schema elements: object nodes related to address entities exhibit higher out-degree, as address views connect to numerous other schema components across diverse contexts. These nodes reference a shared set of higher-level objects. Details of extracted semantic information available in appendix B. This semantic overlay transforms the relational dataset into a graph-linked benchmark, enabling contextual representation learning and semantics-aware evaluation across tabular and graph modalities.

Data Insights and Challenges: SALT-KG inherits the same real-world properties of SALT [Klein et al., 2024]: high cardinality, class imbalance, and temporal drift. Several categorical attributes, such as `SalesOffice` and `ShippingPoint`, are dominated by few frequent classes, whereas others (e.g., `Customer` or `ProductID`) contain tens of thousands distinct values. Temporal drift is evident across the three-year window, as organizational structures and categorical hierarchies evolve with time. Introducing the KG layer opens new opportunities to address these challenges. By linking tabular fields to their semantic object types and descriptive definitions, models can leverage graph-based regularization, hierarchical grouping, or transfer between semantically related columns. E.g. graph relations connecting address-related entities across schema components like `Customer`, `AddressLine`, `PostalLocation` can facilitate domain adaptation or infer meaningful join paths between related tables. This design encourages research on semantics-informed learning strategies that transcend feature engineering, emphasizing generalization through contextual and relational understanding.

[2] https://help.sap.com/docs/SAP_HANA_PLATFORM/09b6623836854766b682356393c6c416/b4080c0883c24d2dae38a60d7fbf07c8.html

Table 1: **Effect of KG context on baseline models (ΔMRR).** Blue values indicate improvement due to KG context; red values indicate degradation; black values denote no change.

Method \ Target	Plant	ShipPt	Item Inc.	Hdr Inc.	Sales Off.	Sales Grp.	PayTrm	ShipCond	Avg
Random Classifier	+0.03	+0.02	0.00	0.00	+0.01	-0.01	-0.02	+0.01	+0.01
Majority Class Baseline	+0.03	-0.05	0.00	0.00	+0.01	-0.01	0.00	0.00	+0.00
XGBoost ([Chen and Guestrin, 2016])	0.00	0.00	+0.01	+0.01	+0.01	0.00	0.00	+0.01	+0.01
LightGBM ([Ke et al., 2017])	+0.01	0.00	+0.03	+0.04	+0.01	+0.01	+0.08	-0.31	+0.00
CatBoost ([Prokhorenkova et al., 2018])	0.00	-0.09	-0.02	0.00	+0.01	0.00	0.00	0.00	-0.01
CARTE ([Kim et al., 2024])	0.00	+0.01	+0.02	+0.02	+0.01	-0.01	+0.03	+0.04	+0.02
AutoGluon ([Erickson et al., 2020])	0.00	0.00	+0.02	+0.03	+0.01	+0.02	-0.04	0.00	+0.03
GraphSAGE ([Hamilton et al., 2017])	0.00	+0.01	0.00	-0.01	+0.01	-0.02	+0.01	-0.10	-0.01

3 Experiments and Results

We evaluate SALT-KG using standard tabular baselines trained on the joined relational schema, similar to the setup in SALT Klein et al. [2024]. Since OBKG provides schema rather than record level information, we independently encode the information (concatenated in the order: CDS View, Fields, objNodeTypes), using TEXT-EMBEDDING-3-LARGE model, which we chose as baseline method for simplicity of implementation. The encoded descriptors are then projected into a compact latent space via PCA. We perform early fusion of the two representations, where tabular features are encoded into unified row vectors by retaining both the numerical and categorical features for tree based methods, while relying on intrinsic representations for the neural models. The resultant reduced-dimensional semantic representation is concatenated with tabular features for every row. This final representation is then used to train all eight benchmark models. Empirically, retaining between 16 and 64 components provides a stable trade-off for expressiveness and generalization, while higher dimensions may lead to noise amplification, weak signal utilization, and increased feature space complexity that hinders robust learning leading to degraded performance.

Deep learning baselines benefit modestly from this semantic context, while graph-based models, such as GRAPHSAGE [Hamilton et al., 2017], show inconsistency, highlighting the need for architectures that directly integrate declarative schema semantics. Across all baselines, incorporating metadata-derived features rarely changes overall ranking metrics but consistently alters the relative performance of different model families, as shown by the slight deviations in Tab. 1. Tree-based methods naturally handle heterogeneous data types and can split on categorical features without requiring heavy embedding hence, they remain relatively stable, whereas neural baselines exhibit greater sensitivity to this alignment with operational data. Deep learning on tabular data often struggles when the number of samples is moderate, when features vary in type and scale, and when the target function is irregular (non-smooth) Shwartz-Ziv and Armon [2022]. This pattern suggests that semantic grounding modifies inductive biases rather than raw predictive accuracy. The magnitude of these effects also reveals a structural limitation of the underlying dataset: while SALT-KG introduces declarative semantics through OBKG, the relational scaffold inherited from SALT provides limited ontological depth (where the data exhibits only direct relationships between tables and fields, but does not capture higher-order abstractions such as class hierarchies, class-instance relationships, and rich expressivity through transitivity, inverse, reflexivity) and cross-entity abstraction. As a result, available semantics cannot fully propagate through the relational topology, constraining the degree to which current models can internalize and exploit higher-order context. The modest gains observed indicate that semantics-aware learning also demands intrinsically structured datasets with a relational design that supports the emergence of semantic generalization.

4 Conclusion

By explicitly linking relational schema elements to declarative metadata in the OBKG, SALT-KG closes a critical gap between purely structural benchmarks for tabular learning and the semantics-grounded evaluation needed for the next generation of tabular foundation models. Empirical results, proffer a need for future work on architectures that unify relational, semantic, and linguistic understanding. The declarative semantics encoded in SALT-KG remain bounded by the descriptive depth of the underlying SALT dataset, where the OBKG semantics are defined using existing enterprise schema and glossary, limiting the extent of cross-domain or hierarchical reasoning. SALT-KG thus provides the first benchmark for semantics-aware tabular learning and offers a foundation for systematically studying how contextual knowledge shapes predictive behavior.

References

Nora Abdelmageed, Sirko Schindler, and Birgitta König-Ries. Jentab: Bridging tabular data and knowledge graphs – a detailed system overview. *Semantic Web Journal*, 2024.

Jan-Micha Bodensohn, Ulf Brackmann, Liane Vogel, Matthias Urban, Anupam Sanghi, and Carsten Binnig. Llms for data engineering on enterprise data. In *VLDB Workshops (Tabular Data Analysis Track)*, 2024.

Tianqi Chen and Carlos Guestrin. Xgboost: A scalable tree boosting system. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '16, New York, NY, USA, 2016. ACM. ISBN 9781450342322.

Nick Erickson, Jonas Mueller, Alexander Shirkov, Hang Zhang, Pedro Larroy, Mu Li, and Alexander Smola. Autogluon-tabular: Robust and accurate automl for structured data, 2020. URL `https://arxiv.org/abs/2003.06505`.

Nick Erickson, Lennart Purucker, Andrej Tschalzev, David Holzmüller, Prateek Mutalik Desai, David Salinas, and Frank Hutter. Tabarena: A living benchmark for machine learning on tabular data, 2025. URL `https://arxiv.org/abs/2506.16791`.

Leo Grinsztajn, Edouard Oyallon, and Gael Varoquaux. Why do tree-based models still outperform deep learning on typical tabular data? In *Thirty-sixth Conference on Neural Information Processing Systems Datasets and Benchmarks Track*, 2022.

William L. Hamilton, Rex Ying, and Jure Leskovec. Inductive representation learning on large graphs. In *NIPS*, 2017.

Noah Hollmann, Samuel Müller, Katharina Eggensperger, and Frank Hutter. Tabpfn: A transformer that solves small tabular classification problems in a second. In *ICSL'23*, 2023.

Noah Hollmann, Samuel Müller, Lennart Purucker, Arjun Krishnakumar, Max Körfer, Shi Bin Hoo, Robin Tibor Schirrmeister, and Frank Hutter. Accurate predictions on small data with a tabular foundation model. *Nature*, 2025.

Guolin Ke, Qi Meng, Thomas Finley, Taifeng Wang, Wei Chen, Weidong Ma, Qiwei Ye, and Tie-Yan Liu. Lightgbm: a highly efficient gradient boosting decision tree. NIPS'17, Red Hook, NY, USA, 2017. Curran Associates Inc. ISBN 9781510860964.

Myung Jun Kim, Léo Grinsztajn, and Gaël Varoquaux. Carte: pretraining and transfer for tabular learning. *arXiv preprint arXiv:2402.16785*, 2024.

Tassilo Klein and Johannes Hoffart. Foundation models for tabular data within systemic contexts need grounding, 2025. URL `https://arxiv.org/abs/2505.19825`.

Tassilo Klein, Clemens Biehl, Margarida Costa, Andre Sres, Jonas Kolk, and Johannes Hoffart. SALT: Sales autocompletion linked business tables dataset. In *NeurIPS 2024 Third Table Representation Learning Workshop*, 2024. URL `https://openreview.net/forum?id=UZbELpkWIr`.

Si-Yang Liu, Hao-Run Cai, Qi-Le Zhou, and Han-Jia Ye. Talent: A tabular analytics and learning toolbox. *arXiv preprint arXiv:2407.04057*, 2024.

I. Onando Mulang', Kuldeep Singh, Chaitali Prabhu, Abhishek Nadgeri, Johannes Hoffart, and Jens Lehmann. Evaluating the impact of knowledge graph context on entity disambiguation models. In *CIKM*, 2020.

Liudmila Prokhorenkova, Gleb Gusev, Aleksandr Vorobev, Anna Veronika Dorogush, and Andrey Gulin. Catboost: unbiased boosting with categorical features. In *Proceedings of the 32nd International Conference on Neural Information Processing Systems*, NIPS'18, page 6639–6649, Red Hook, NY, USA, 2018. Curran Associates Inc.

Jingyi Qiu, Aibo Song, Jiahui Jin, Jiaoyan Chen, Xinyu Zhang, Xiaolin Fang, and Tianbo Zhang. Matching tabular data to knowledge graph with effective core column set discovery. In *ACM SIGMOD*, 2024.

Jingang Qu, David Holzmüller, Gaël Varoquaux, and Marine Le Morvan. Tabicl: A tabular foundation model for in-context learning on large data. *arXiv preprint arXiv:2502.05564*, 2025.

Joshua Robinson, Rishabh Ranjan, Weihua Hu, Kexin Huang, Jiaqi Han, Alejandro Dobles, Matthias Fey, Jan E. Lenssen, Yiwen Yuan, Zecheng Zhang, Xinwei He, and Jure Leskovec. Relbench: A benchmark for deep learning on relational databases, 2024.

Ravid Shwartz-Ziv and Amitai Armon. Tabular data: Deep learning is not all you need. *Inf. Fusion*, 2022.

Marco Spinaci, Marek Polewczyk, Johannes Hoffart, Markus C. Kohler, Sam Thelin, and Tassilo Klein. PORTAL: Scalable tabular foundation models via content-specific tokenization. In *NeurIPS 3rd TRL Workshop*, 2024.

Marco Spinaci, Marek Polewczyk, Maximilian Schambach, and Sam Thelin. Contexttab: A semantics-aware tabular in-context learner. In *1st ICML Workshop on Foundation Models for Structured Data*, 2025.

Nathan Vandemoortele, Bram Steenwinckel, Sofie Van Hoecke, and Femke Ongenae. Scalable table-to-knowledge graph matching from metadata using llms. In *SemTab'24: Semantic Web Challenge on Tabular Data to Knowledge Graph Matching 2024,ISWC, CEUR Workshop Proceedings*, 2024.

A Model Parameters

Table 2: Hyperparameters Used by Each Model

Model	Parameter	Value
AutoGluon	*NN_TORCH Hyperparameters*	
	num_epochs	100
	batch_size	8192
	learning_rate	0.001
	dropout_prob	0.1
	num_gpus	1
	FASTAI Hyperparameters	
	epochs	50
	batch_size	4096
	layers	[800, 400]
	num_gpus	1
	General Training Parameters	
	time_limit	7200s (2 hrs)
	holdout_frac	0.1
	num_bag_folds	3
	presets	best_quality
GraphSAGE	epochs	1000
	patience	5
	num_layers	2
	channels	128
	aggregation	sum
	normalization	batch_norm
	learning_rate	0.0001
	optimizer_eps	1e-8
	tune_metric	mrr
CARTE	num_model	1
	random_state	42
	n_jobs	1
	loss	categorical_ cross-entropy
	val_size	0.10
	batch_size	4
	early_stopping _patience	5
	learning_rate	0.0006
	num_layers	3
	dropout	0.1
	max_epoch	200
	chunk_size	2048

B Dataset Statistics

This hybrid relational–graph structure supports new learning paradigms that jointly reason over transactional evidence and semantic context, combining tabular embeddings with graph-based reasoning. Table 3 summarizes the composition and semantic augmentation introduced in SALT-KG.

Table 3: Summary of CDS Views infused in SALT-KG.

CDS View	Fields	Obj Node Types
I_SalesDocument	286	1
I_SalesDocumentItem	412	1
I_Customer	132	1
I_Address_2	94	1,954
I_AddrOrgPostalAddress	66	1,954

C Sample KG Context: I_SALESDOCUMENT


```
{
    "I_SALESDOCUMENT": {
        "name": "I_SALESDOCUMENT",
        "description": "Sales Document",
        "shortDescription": "This CDS view provides the prerequisites
            for answering questions about all relevant aspects of
            sales documents. Example business questions could include:
             What is the net value of a given sales document? Based on
             which document is a given sales order created? Which
            sales area does a given sales document belong to? Who is
            the sold-to party of a given sales document? When is a
            given sales order requested to be delivered? What is the
            overall processing status of a given sales document?",
        "details": "",
        "fields": [
            {
                "fieldName": "SALESDOCUMENT",
                "fieldDescription": "Sales Document",
                "fieldType": "abap.char",
                "fieldDetails": "",
                "dataElementDescription": "The number that uniquely
                    identifies the sales document.",
                "targetColumn" : "SALESDOCUMENT"
            },
            {
                "fieldName": "SALESDOCUMENTTYPE",
                "fieldDescription": "Sales Document Type",
                "fieldType": "abap.char",
                "fieldDetails": "",
                "dataElementDescription": "A classification that
                    distinguishes between different types of sales
                    documents.",
                "targetColumn" : "SALESDOCUMENTTYPE"
            },
            {
                "fieldName": "SALESORGANIZATION",
                "fieldDescription": "Sales Organization",
                "fieldType": "abap.char",
                "fieldDetails": "",
                "dataElementDescription": "An organizational unit
                    responsible for the sale of certain products or
                    services.",
                "targetColumn" : "SALESORGANIZATION"
            },
            {
                "fieldName": "DISTRIBUTIONCHANNEL",
                "fieldDescription": "Distribution Channel",
                "fieldType": "abap.char",
                "fieldDetails": "",
```

```
            "dataElementDescription": "The way in which products
                or services reach the customer.",
            "targetColumn" : "DISTRIBUTIONCHANNEL"
        },
        {
            "fieldName": "ORGANIZATIONDIVISION",
            "fieldDescription": "Division",
            "fieldType": "abap.char",
            "fieldDetails": "",
            "dataElementDescription": "A way of grouping materials
                , products, or services.",
            "targetColumn" : "ORGANIZATIONDIVISION"
        },
        {
            "fieldName": "CREATIONDATE",
            "fieldDescription": "Record Creation Date",
            "fieldType": "abap.dats",
            "fieldDetails": "",
            "dataElementDescription": "",
            "targetColumn" : "CREATIONDATE"
        },
        {
            "fieldName": "CREATIONTIME",
            "fieldDescription": "Time at Which Record Was Created"
                ,
            "fieldType": "abap.tims",
            "fieldDetails": "",
            "dataElementDescription": "The time of day at which
                the document was posted and stored in the system."
                ,
            "targetColumn" : "CREATIONTIME"
        },
        {
            "fieldName": "TRANSACTIONCURRENCY",
            "fieldDescription": "SD Document Currency",
            "fieldType": "abap.cuky",
            "fieldDetails": "",
            "dataElementDescription": "The currency that applies
                to the document.",
            "targetColumn" : "TRANSACTIONCURRENCY"
        },
        {
            "fieldName": "CUSTOMERPAYMENTTERMS",
            "fieldDescription": "Terms of Payment Key",
            "fieldType": "abap.char",
            "fieldDetails": "",
            "dataElementDescription": "Key for defining payment
                terms composed of cash discount percentages and
                payment periods.",
            "targetColumn" : "CUSTOMERPAYMENTTERMS"
        },
        {
            "fieldName": "SHIPPINGCONDITION",
            "fieldDescription": "Shipping Conditions",
            "fieldType": "abap.char",
            "fieldDetails": "",
            "dataElementDescription": "General shipping strategy
                for the delivery of goods from the vendor to the
                customer.",
            "targetColumn" : "SHIPPINGCONDITION"
        },
        {
            "fieldName": "INCOTERMSCLASSIFICATION",
            "fieldDescription": "Incoterms (Part 1)",
            "fieldType": "abap.char",
```

```
            "fieldDetails": "",
            "dataElementDescription": "Commonly used trading terms
                that comply with the standards established by the
                International Chamber of Commerce (ICC).",
            "targetColumn" : "INCOTERMSCLASSIFICATION"
        },
        {
            "fieldName": "BILLINGCOMPANYCODE",
            "fieldDescription": "Company Code to Be Billed",
            "fieldType": "abap.char",
            "fieldDetails": "",
            "dataElementDescription": "The company code represents
                an independent accounting unit.",
            "targetColumn" : "BILLINGCOMPANYCODE"
        },
        {
            "fieldName": "SALESGROUP",
            "fieldDescription": "Sales Group",
            "fieldType": "abap.char",
            "fieldDetails": "",
            "dataElementDescription": "A group of sales people who
                are responsible for processing sales of certain
               products or services.",
            "targetColumn" : "SALESGROUP"
        },
        {
            "fieldName": "SALESOFFICE",
            "fieldDescription": "Sales Office",
            "fieldType": "abap.char",
            "fieldDetails": "",
            "dataElementDescription": "A physical location that
               has responsibility for the sale of certain
               products or services within a given geographical
               area.",
            "targetColumn" : "SALESOFFICE"
        },
        "... [Status Fields - Collapsed for brevity]",
        "OVERALLSDPROCESSSTATUS, TOTALBLOCKSTATUS,
            OVERALLDELIVERYSTATUS",
        "OVERALLDELIVERYBLOCKSTATUS, OVERALLBILLINGBLOCKSTATUS",
        "OVERALLORDRELTDBILLGSTATUS, TOTALCREDITCHECKSTATUS",
        "CENTRALCREDITCHECKSTATUS, SALESDOCAPPROVALSTATUS",
        "... [Customer Fields - Collapsed for brevity]",
        "SOLDTOPARTY, CUSTOMERGROUP, CUSTOMERPRICEGROUP",
        "ADDITIONALCUSTOMERGROUP1-5, RETAILADDITIONALCUSTOMERGRP6
            -10",
        "CUSTOMERTAXCLASSIFICATION1-9",
        "... [Contract Fields - Collapsed for brevity]",
        "AGRMTVALDTYSTARTDATE, AGRMTVALDTYENDDATE",
        "SALESCONTRACTCANCLNREASON, SALESCONTRACTCANCLNPARTY",
        "SALESCONTRACTFOLLOWUPACTION, CONTRACTMANUALCOMPLETION",
        "... [Financial Fields - Collapsed for brevity]",
        "TOTALNETAMOUNT, CONTROLLINGAREA, COSTCENTER",
        "PRICEDETNEXCHANGERATE, ACCOUNTINGEXCHANGERATE",
        "SDPRICINGPROCEDURE, PRICINGDATE, PRICELISTTYPE",
        "... [Reference Fields - Collapsed for brevity]",
        "REFERENCESDDOCUMENT, REFERENCESDDOCUMENTCATEGORY",
        "PURCHASEORDERBYCUSTOMER, CUSTOMERPURCHASEORDERDATE",
        "EXTERNALDOCUMENTID, ACCOUNTINGDOCEXTERNALREFERENCE"
    ],
    "objNodeTypes": [
        {
            "objNodeTypeName": "SalesDocument",
            "objNodeTypeLabe": "Sales Document",
        }
```

```
            ]
        }
}
```