SALT: Sales Autocompletion Linked Business Tables Dataset

Anonymous Author(s)
Affiliation
Address
email

Abstract

Foundation models, particularly those that incorporate Transformer architectures, have demonstrated exceptional performance in domains such as natural language processing and image processing. Adapting these models to structured data, like tables, however, introduces significant challenges. These difficulties are even more pronounced when addressing multi-table data linked via foreign key, which is prevalent in the enterprise realm and crucial for empowering business use-cases. Despite its substantial impact, the research focusing on such linked business tables within enterprise settings remains a significantly important yet underexplored domain. To address this, we introduce a curated dataset sourced from an Enterprise Resource Planning (ERP) system, featuring extensive linked tables. This dataset is specifically designed to support research endeavors in table representation learning. By providing access to authentic enterprise data, our goal is to potentially enhance the effectiveness and applicability of models for real-world business contexts.

1 Introduction

Deep learning has made substantial strides in areas like text understanding, language translation, image classification, and object detection. These advancements are largely driven by foundational models trained on diverse datasets and self-supervised training techniques, especially those that incorporate Transformer architectures. However, using these models on structured, tabular data, essential for enterprise business operations, poses unique challenges. These challenges become more pronounced with multi-table configurations consisting of large tables interconnected by foreign keys and comprising extensive business datasets, a setup to which we refer to as *linked business tables*. Such setups are common in real-world business scenarios.

The challenges in applying foundational models to linked business data are primarily twofold: algorithmic and data-related. Algorithmically, a significant challenge is adapting models that were originally designed for unstructured internet data to handle structured data effectively - see (Grinsztajn et al., 2022) for a comprehensive discussion. This process requires a sophisticated integration of structural knowledge and the unique characteristics of linked business data, which is inherently more complex and interconnected than straightforward internet-scraped table data.

One major limitation in the current landscape is the absence of realistic, enterprise-linked multi-table datasets at scale. Existing table datasets often originate from HTML pages and do not accurately represent the complexity and dynamics of expansive database tables used in active enterprise systems (Bodensohn et al., 2024). Moreover, obtaining large, clean, and high-quality datasets for structured tabular applications presents difficulties (Hulsebos et al., 2023; Van Breugel and Van Der Schaar, 2024), particularly in enterprise settings where data privacy, confidentiality, and commercial interests restrict data access. This lack of suitable public datasets leads to significant domain adaptation challenges and shifts in data distribution, which pose difficulties for many existing models (Fey et al., 2024).

Submitted to Table Representation Learning Workshop at NeurIPS 2024. Do not distribute.

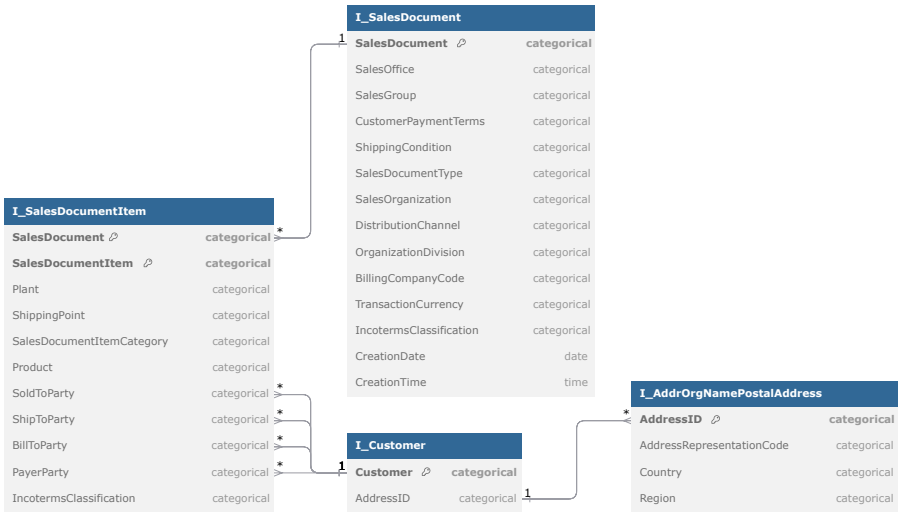

Figure 1: **Dataset Schemas**. Schemas for the four tables constituting the SALT dataset. Primary keys are highlighted in **bold** letters and with a key symbol. Foreign keys interconnecting tables.

To tackle these issues, we have curated the Sales Autocompletion Linked Business Tables (SALT) dataset, sourced from an Enterprise Resource Planning (ERP) system. ERP systems are comprehensive, multifunctional platforms essential for managing all core business operations including finance, human resources, production, and supply chains. As the backbone of organizational data management, ERP systems provide an excellent foundation for developing and evaluating data models that accurately reflect complex, real-world enterprise environments. The SALT dataset, which includes interconnected relational tables with a focus on sales, encompasses several million entries across various enterprise sales operations.

By sharing the SALT dataset with the research community, we aim to stimulate advancements in table representation learning and refine algorithm development to enhance applicability and performance in real-world settings. This initiative is crucial for evolving deep learning models that not only understand but also effectively function within the complexities of large-scale enterprise data landscapes, thereby promoting the development of enterprise-specific machine learning applications.

Related Work: The majority of existing table datasets originate from scraping the Web, notably extracted from HTML pages or CSV files from GitHub, which inadequately capture the complexity and dynamics typical of large database tables that are employed in operational enterprise systems. WebTables (Cafarella et al., 2008) corpus includes a massive collection of 233 million tables, sourced from HTML pages via the Common Crawl project. While WebTables offers an extensive quantity of tables, its diversity is constrained because it solely comprises HTML tables from web pages. TURL Deng et al. (2020) provides a cleaner corpus of 580 thousand tables extracted from Wikipedia. In contrast, GitTables (Hulsebos et al., 2023) contains over 10 million tables extracted from "comma-separated value" files (CSVs) found on GitHub. Tables from GitTables generally exhibit structural differences compared to those from WebTables, making GitTables an essential corpus despite its focus on a single file type. TabLib (Eggert et al., 2023) comprises 627 million tables across various file formats and totaling 69 TiB, sourced from GitHub and Common Crawl. Notably, it comprises exceptionally large tables of several million rows and columns. LakeBench (Deng et al., 2024) is a collection of benchmarks to resemble enterprise data lakes, containing tables from a variety of sources such as open government data for the purpose of unionability, joinability, and subset tasks.

2 SALT Dataset

Background: The SALT dataset is specifically curated to mirror customer interactions within an Enterprise Resource Planning (ERP) system and is designed to train models that assist users by predicting fields typically missing in sales orders. This dataset is crucial to the sales and distribution process, especially for creating the "Sales Order Document." Each of these documents records a single transaction that includes various items intended for specific customers, marking a distinct

Table 1: **Breakdown of atomic data composition of datasets:** Datasets SALT, *GitTables* (Hulsebos et al., 2023), *WebTables* (Lehmberg et al., 2016) and *TabLib* (Eggert et al., 2023)

Atomatic data type	SALT	GitTables	WebTables	TabLib
Numeric	38.7%	57.9%	51.4%	33.6%
String	58.1%	41.6%	47.4%	61.8%
Other	3.2%	0.5%	1.2%	4.6%

phase in the sales cycle.
Structured around four principal tables—sales documents, sales document items, customers, and addresses—the dataset consolidates data from a single enterprise that underwent anonymization. The sales documents table logs vital details such as sales office, sales group, payment conditions, and shipping arrangements, limiting its entries to those specifically categorized as sales orders. The sales document items table captures detailed information for each line item in these documents, including the product sold, shipping point, and the parties involved in the transaction. Concurrently, the customer table holds comprehensive master data about customers, further elaborated in the addresses table with specifics like country and region. The input variables in the dataset include a mix of fields typically populated by users during the creation of a sales order, augmented by master data fields like material number and customer details. The target variables are not always maintained; they are optional and may not be filled out for certain transactions depending on particular scenarios or requirements. This intricate structure of SALT not only enhances model training for missing field predictions but also effectively replicates complex ERP interactions within sales order automation.

Task: In the dataset, 21 fields are categorized as input variables, serving as features for predictive modeling applications, while 8 fields are designated as target variables, intended for prediction based on the input data analysis. The predictive model, which will be trained using this dataset, is specifically tasked with performing multiclass classification on seven critical variables. These variables are essential for ensuring the seamless execution of sales orders:

- `I_SalesDocument.SalesOffice` - Sales activities for specific products and regions
- `I_SalesDocument.SalesGroup` - Sales representatives managing responsibilities
- `I_SalesDocument.CustomerPaymentTerms` - Payment conditions, i.e., deadlines and early payment discounts
- `I_SalesDocument.ShippingCondition` - Logistics terms
- `I_SalesDocumentItem.ShippingPoint` - Dispatch location
- `I_SalesDocumentItem.Plant` - Production/ storage facility, critical for inventory control
- `I_SalesDocument.IncotermsClassification` and `I_SalesDocumentItem.IncotermsClassification` - International commercial terms, outline transaction responsibilities like shipping and insurance[1]

Structure: The dataset is structured into four primary tables encompassing a total of 573,810 sales orders (`I_SalesDocument`), which include 2,706,491 sales order items (`I_SalesDocumentItem`) associated with 136,317 unique business partners (`I_Customer`) and 1,650,641 (`I_AddrOrgNamePostalAddress`) addresses - see Fig. 1 for the table schemas. The table fields are filtered to include only the data relevant to the specific use case described above. After filtering, the tables are merged to form a single flat dataset containing 2,706,491 rows, such that each row in the dataset represents a single sales order item (for more details see Appendix Sec. A.3). The entries cover transactions conducted between January 1, 2018, and December 31, 2020. To assess the dataset's predictive modeling utility, data was divided with temporal splits, with validation segments starting from February 1, 2020, and test segments from July 1, 2020. For on analysis on the distribution values see Tab. 1.

Data Insights: The dataset employed in this study is derived from authentic industry data captured by an Enterprise Resource Planning (ERP) system, documenting sales orders. This dataset has undergone minimal pre-processing primarily aimed at addressing privacy concerns. Several challenges arise from the nature and quality of the dataset, which need careful consideration:

- **Diversity:** There is a substantial diversity in certain data fields due to the wide range of unique values they contain. For instance, the field `I_SalesDocumentItem.ShipToParty`

[1]This field can be defined independently on item and header level, which is why both are included.

includes 21,997 distinct customer IDs, while `I_SalesDocumentItem.Product` comprises 209,823 unique product identifiers.

- **Class imbalance:** The dataset demonstrates a pronounced class imbalance. The distribution of sales offices across sales orders is highly skewed; the most frequently occurring sales office is associated with 75% of the orders, and the two most common sales offices collectively account for 98% of the data. Despite this, there are 33 distinct sales offices represented in more than one order, suggesting a long-tail distribution.
- **Noise:** A considerable amount of input noise is evident within the dataset. Since data entry is frequently manual, discrepancies may arise as different employees might handle identical business scenarios differently or make inadvertent errors. Moreover, certain fields may be occasionally left blank, potentially leading to gaps in the data.
- **Data drift:** Technically, the dataset is prone to data drift, a phenomenon where the categorizations such as sales groups within the ERP system evolve over time. This drift may particularly impact analyses involving temporal splits of the data, as category definitions may shift across the time periods considered. Notably, the target categories are not subject to such data drift.

Table 2: **Classification performance of baseline models:** Evaluation of baseline models on the eight different tasks on SALT. **Top:** Simple baselines **Middle:** Gradient-boosted decision tree models **Bottom:** Deep learning methods. **Performance metric:** Mean Reciprocal Rank.

Performance Baseline - MRR (↑)					
Method \ Target Variable	**Plant**	**Shipping Point**	**Item Incoterm Cls.**	**Header Incoterm Cls.**	
Random Classifier	0.32	0.22	0.33	0.33	
Majority Class Baseline	0.51	0.41	0.49	0.49	
XGBoost (Chen and Guestrin, 2016)	**0.99**	0.96	0.73	0.73	
LightGBM (Ke et al., 2017)	**0.99**	0.86	**0.82**	**0.82**	
CatBoost (Prokhorenkova et al., 2018)	**0.99**	**0.99**	**0.82**	**0.82**	
CARTE (Kim et al., 2024)	0.97	0.96	0.77	0.77	
AutoGluon (Erickson et al., 2020)	**0.99**	0.98	0.79	0.79	

(continued)	**Sales Office**	**Sales Group**	**Pay. Terms**	**Ship. Condition**	**Avg.**
Random Classifier	0.97	0.01	0.13	0.16	0.31
Majority Class Baseline	**0.99**	0.03	0.23	0.29	0.43
XGBoost (Chen and Guestrin, 2016)	0.98	0.62	0.75	0.71	0.81
LightGBM (Ke et al., 2017)	**0.99**	0.21	0.83	0.80	0.79
CatBoost (Prokhorenkova et al., 2018)	**0.99**	0.17	0.56	0.74	0.76
CARTE (Kim et al., 2024)	**0.99**	0.40	0.70	0.77	0.79
AutoGluon (Erickson et al., 2020)	**0.99**	**0.66**	**0.84**	**0.82**	**0.86**

3 Experiments & Results

We evaluate the SALT dataset using several baselines for tabular data on the joined table. The only preprocessing applied is filling in missing values with either a constant value (for the categorical features) or the mean value (for numerical features). The fields related to creation date and time were only used to split the data and then discarded. The validation set was used for early stopping and no hyperparameter tuning was performed. See Tab. 2 for the detailed breakdown of performance evaluation tasks of each task. As can be seen the AutoGluon Erickson et al. (2020) shows best performance on SALT with a significant margin of $(+0.05\ p.)$ w.r.t. the next best approach XGBoost (Chen and Guestrin, 2016). The analysis reveals several noteworthy insights: **i)** Certain target variables demonstrate substantial predictability, achieving prediction scores near 0.99, indicating a high degree of accuracy. **ii)** The dataset exhibits significant class imbalance, which is particularly evident from the performance of the majority class baseline. This imbalance is most pronounced when predicting variables such as the Sales Office. **iii)** The predictive performance of the model is adversely affected when tasked with predicting fields like the Sales Group, which suffers from high cardinality issues, complicating accurate classification.

4 Conclusion

We introduce a novel dataset focused on linked business data, demonstrating the characteristics of data within actual enterprise systems. We further assessed the performance of current tabular models against tree-based and cutting-edge models. The empirical data reveal that most tabular models effectively manage the prediction tasks in SALT. To augment the dataset's complexity and utility in future work, we plan to include additional tables from a broader range of scenarios, data from multiple companies, and enhance the semantic richness of the dataset to present greater challenges.

References

Jan-Micha Bodensohn, Ulf Brackmann, Liane Vogel, Matthias Urban, Anupam Sanghi, and Carsten Binnig. 2024. Llms for data engineering on enterprise data. In *Joint proceedings of workshops at the 50th International Conference on Very Large Data Bases (VLDB 2024), Guangzhou, China, August 26 - August 30, 2023, VLDBW 2024*, Tabular Data Analysis Workshop Proceedings.

Michael J. Cafarella, Alon Halevy, Daisy Zhe Wang, Eugene Wu, and Yang Zhang. 2008. Webtables: exploring the power of tables on the web. *Proc. VLDB Endow.*, 1(1):538–549.

Tianqi Chen and Carlos Guestrin. 2016. XGBoost: A scalable tree boosting system. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '16, pages 785–794, New York, NY, USA. ACM.

Xiang Deng, Huan Sun, Alyssa Lees, You Wu, and Cong Yu. 2020. Turl: table understanding through representation learning. *Proc. VLDB Endow.*, 14(3):307–319.

Yuhao Deng, Chengliang Chai, Lei Cao, Qin Yuan, Siyuan Chen, Yanrui Yu, Zhaoze Sun, Junyi Wang, Jiajun Li, Ziqi Cao, Kaisen Jin, Chi Zhang, Yuqing Jiang, Yuanfang Zhang, Yuping Wang, Ye Yuan, Guoren Wang, and Nan Tang. 2024. Lakebench: A benchmark for discovering joinable and unionable tables in data lakes. *Proc. VLDB Endow.*, 17(8):1925–1938.

Gus Eggert, Kevin Huo, Mike Biven, and Justin Waugh. 2023. Tablib: A dataset of 627m tables with context.

Nick Erickson, Jonas Mueller, Alexander Shirkov, Hang Zhang, Pedro Larroy, Mu Li, and Alexander Smola. 2020. Autogluon-tabular: Robust and accurate automl for structured data. *arXiv preprint arXiv:2003.06505*.

Matthias Fey, Weihua Hu, Kexin Huang, Jan Eric Lenssen, Rishabh Ranjan, Joshua Robinson, Rex Ying, Jiaxuan You, and Jure Leskovec. 2024. Position: Relational deep learning - graph representation learning on relational databases. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pages 13592–13607. PMLR.

Leo Grinsztajn, Edouard Oyallon, and Gael Varoquaux. 2022. 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*.

Madelon Hulsebos, Çagatay Demiralp, and Paul Groth. 2023. Gittables: A large-scale corpus of relational tables. *Proceedings of the ACM on Management of Data*, 1(1):1–17.

Guolin Ke, Qi Meng, Thomas Finley, Taifeng Wang, Wei Chen, Weidong Ma, Qiwei Ye, and Tie-Yan Liu. 2017. Lightgbm: a highly efficient gradient boosting decision tree. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, NIPS'17, page 3149–3157, Red Hook, NY, USA. Curran Associates Inc.

Myung Jun Kim, Leo Grinsztajn, and Gael Varoquaux. 2024. Carte: Pretraining and transfer for tabular learning. In *Forty-first International Conference on Machine Learning*.

Oliver Lehmberg, Dominique Ritze, Robert Meusel, and Christian Bizer. 2016. A large public corpus of web tables containing time and context metadata. In *Proceedings of the 25th International Conference on World Wide Web, WWW 2016, Montreal, Canada, April 11-15, 2016, Companion Volume*, pages 75–76. ACM.

Liudmila Ostroumova Prokhorenkova, Gleb Gusev, Aleksandr Vorobev, Anna Veronika Dorogush, and Andrey Gulin. 2018. Catboost: unbiased boosting with categorical features. In *NeurIPS*, pages 6639–6649.

Boris Van Breugel and Mihaela Van Der Schaar. 2024. Position: Why tabular foundation models should be a research priority. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pages 48976–48993. PMLR.

A Appendix

A.1 Table Detailed Information

This section describes the schemas of the four tables that constitute the SALT dataset. Figure 1 shows the schema of the tables. See Tab. 5 for an detailed overview of the *SalesDocumentItem*, Tab. 3 for *SalesOrder* details, Tab. 6 for customer data detail and Tab. 4 for the customer address details. The column *Is Target Field* indicates the target fields which should be predicted based on the other fields in the tables, since they are populated in a later stage of the sales order creation.

A.2 Additional Data Statistics

This section provides additional statistics on data composition. Table 8 provides statistics on the target fields. Table 8 provides a detailed statistics of the table fields.

Table 3: **SalesDocument in Detail**

Field Name	Data Type	Description	Is Target Field
SalesDocument	Categorical (integer)	ID of the sales document	
SalesDocumentType	Categorical (string)	Type of sales document	
SalesOrganization	Categorical (string)	ID of the sales organization	
DistributionChannel	Categorical (string)	ID of the distribution channel	
OrganizationDivision	Categorical (string)	ID of the organization division	
BillingCompanyCode	Categorical (string)	Company code to be billed	
TransactionCurrency	Categorical (string)	Currency code (EUR, USD, ...)	
CreationDate	Date (string)	Date of sales document creation	
CreationTime	Time (string)	Time of day of sales document creation	
SalesOffice	Categorical (string)	ID of the sales office	✓
SalesGroup	Categorical (string)	ID of the sales group	✓
CustomerPaymentTerms	Categorical (string)	ID of the payment terms	✓
ShippingCondition	Categorical (string)	ID of the shipping condition	✓
IncotermsClassification	Categorical (string)	ID of the incoterms	✓

Table 4: **AddrOrgNamePostalAddress in Detail**

Field Name	Data Type	Description	Is Target Field
AddressID	Categorical (integer)	ID of the address	
AddressRepresentationCode	Categorical (integer)	System internal address code	
Country	Categorical (string)	Country name	
Region	Categorical (string)	Region name	

Table 5: **SalesDocumentItem in Detail**

Field Name	Data Type	Description	Is Target Field
SalesDocument	Categorical (integer)	ID of the sales document	
SalesDocumentItem	Categorical (integer)	ID of the sales document item	
SalesDocumentItemCategory	Categorical (string)	ID of the item category	
Product	Categorical (integer)	ID of the product sold	
SoldToParty	Categorical (integer)	ID of the customer sold to	
ShipToParty	Categorical (integer)	ID of the customer shipped to	
BillToParty	Categorical (integer)	ID of the customer billed to	
PayerParty	Categorical (integer)	ID of the payer	
Plant	Categorical (string)	ID of the plant	✓
ShippingPoint	Categorical (string)	ID of the shipping point	✓
IncotermsClassification	Categorical (string)	ID of the incoterms	✓

A.3 Data Extraction and Processing

The extracted tables are filtered to contain only the data relevant to this use case.

Table 6: **Customer in Detail**

Field Name	Data Type	Description	Is Target Field
Customer	Categorical (integer)	ID of the customer	
AddressID	Categorical (integer)	ID of customer's address	

Table 7: **Target field statistics**

Target field	Unique values (#)	Missing values (%)	Normalized entropy
I_SalesDocument.CustomerPaymentTerms	157	0.96	0.52
I_SalesDocument.IncotermsClassification	13	0.58	0.56
I_SalesDocument.SalesGroup	636	0.77	0.85
I_SalesDocument.SalesOffice	37	0.01	0.18
I_SalesDocument.ShippingCondition	55	0.01	0.55
I_SalesDocumentItem.IncotermsClassification	13	0.21	0.54
I_SalesDocumentItem.Plant	41	0.02	0.49
I_SalesDocumentItem.ShippingPoint	99	0.03	0.51

Table 8: **Field statistics:** Number of unique values, percentage of missing values, normalized entropy.

Table	Field	Unique values (#)	Missing values (%)	Normalized entropy
I_SalesDocument	BillingCompanyCode	33	0.0	0.61
I_SalesDocument	CreationDate	1081	0.0	0.96
I_SalesDocument	CreationTime	63247	0.0	0.97
I_SalesDocument	CustomerPaymentTerms	158	0.96	0.52
I_SalesDocument	DistributionChannel	4	0.0	0.08
I_SalesDocument	IncotermsClassification	14	0.58	0.56
I_SalesDocument	OrganizationDivision	1	0.0	0
I_SalesDocument	SalesDocument	573810	0.0	1.0
I_SalesDocument	SalesDocumentType	16	0.0	0.37
I_SalesDocument	SalesGroup	637	0.77	0.85
I_SalesDocument	SalesOffice	38	0.01	0.18
I_SalesDocument	SalesOrganization	36	0.0	0.6
I_SalesDocument	ShippingCondition	56	0.01	0.55
I_SalesDocument	TransactionCurrency	28	0.0	0.29
I_SalesDocumentItem	BillToParty	19483	0.0	0.78
I_SalesDocumentItem	IncotermsClassification	14	0.21	0.54
I_SalesDocumentItem	PayerParty	19441	0.0	0.78
I_SalesDocumentItem	Plant	42	0.02	0.49
I_SalesDocumentItem	Product	209823	0.0	0.75
I_SalesDocumentItem	SalesDocument	573699	0.0	0.92
I_SalesDocumentItem	SalesDocumentItem	948	0.0	0.57
I_SalesDocumentItem	SalesDocumentItemCategory	24	0.1	0.4
I_SalesDocumentItem	ShipToParty	21998	0.0	0.76
I_SalesDocumentItem	ShippingPoint	100	0.03	0.51
I_SalesDocumentItem	SoldToParty	19454	0.0	0.78
I_Customer	AddressID	136287	0.0	1.0
I_Customer	Customer	136317	0.0	1.0
I_Address	AddressID	1625958	0.0	1.0
I_Address	AddressRepresentationCode	1	100.0	0
I_Address	Country	244	13.17	0.57
I_Address	Region	615	80.54	0.74

- `I_SalesDocument` and `I_SalesDocumentItem` are filtered to contain only documents of category sales order
- Only orders which have been fully processed are included, that is, sales orders which have gone through the entire business process. The goal is to ensure that we are working with the data in its most complete sense and no further changes would be expected.
- `I_Customer` and `I_AddrOrgNamePostalAddress` are filtered to contain only the business partners which appear in the aforementioned sales orders and their respective addresses
- Additionally, the table fields are also filtered to include only those relevant to the use case, which were listed in the previous section

After extraction, the tables were joined together to create a flat structure, as described in List. 1, such that each row in the final table represents one sales order item. Note that, due to this flat structure, the

prediction targets which are defined on the sales order level will be repeated across multiple items / rows. This choice was made to allow for the possibility of training a single model to predict all target fields.

Listing 1: **Table join SQL Query:** SQL query used to join the 4 tables together

```
SELECT
        SalesDocumentItem.SalesDocument,
        SalesDocumentItem.SalesDocumentItem,
        SalesDocument.SalesOffice,
        SalesDocument.SalesGroup,
        SalesDocument.CustomerPaymentTerms,
        SalesDocument.ShippingCondition,
        SalesDocumentItem.Plant,
        SalesDocumentItem.ShippingPoint,
        SalesDocument.SalesDocumentType,
        SalesDocument.SalesOrganization,
        SalesDocument.DistributionChannel,
        SalesDocument.OrganizationDivision,
        SalesDocument.BillingCompanyCode,
        SalesDocument.TransactionCurrency,
        SalesDocumentItem.SalesDocumentItemCategory,
        SalesDocumentItem.Product,
        SalesDocumentItem.SoldToParty,
        SoldToPartyAddress.Country as SoldToPartyCountry,
        SoldToPartyAddress.Region as SoldToPartyRegion,
        SalesDocumentItem.ShipToParty,
        ShipToPartyAddress.Country as ShipToCountry,
        ShipToPartyAddress.Region as ShipToPartyRegion,
        SalesDocumentItem.BillToParty,
        BillToPartyAddress.Country as BillToPartyCountry,
        BillToPartyAddress.Region as BillToPartyRegion,
        SalesDocumentItem.PayerParty,
        PayerPartyAddress.Country as PayerCountry,
        PayerPartyAddress.Region as PayerRegion,
        SalesDocument.CreationDate,
        SalesDocument.CreationTime
FROM    I_SalesDocumentItem AS SalesDocumentItem
        INNER JOIN I_SalesDocument AS SalesDocument
                ON SalesDocument.SalesDocument = SalesDocumentItem.SalesDocument
        LEFT JOIN I_Customer AS SoldToPartyTable
                ON SalesDocumentItem.SoldToParty = SoldToPartyTable.Customer
        LEFT JOIN I_Customer AS ShipToPartyTable
                ON SalesDocumentItem.ShipToParty = ShipToPartyTable.Customer
        LEFT JOIN I_Customer AS BillToPartyTable
                ON SalesDocumentItem.BillToParty = BillToPartyTable.Customer
        LEFT JOIN I_Customer AS PayerPartyTable
                ON SalesDocumentItem.PayerParty = PayerPartyTable.Customer
        LEFT JOIN I_AddrOrgNamePostalAddress AS SoldToPartyAddress
                ON SoldToPartyAddress.AddressID = SoldToPartyTable.AddressID
        LEFT JOIN I_AddrOrgNamePostalAddress AS ShipToPartyAddress
                ON ShipToPartyAddress.AddressID = ShipToPartyTable.AddressID
        LEFT JOIN I_AddrOrgNamePostalAddress AS BillToPartyAddress
                ON BillToPartyAddress.AddressID = BillToPartyTable.AddressID
        LEFT JOIN I_AddrOrgNamePostalAddress AS PayerPartyAddress
                ON PayerPartyAddress.AddressID = PayerPartyTable.AddressID
```