A Technical/Scientific Document Management Platform

Melissa Lemos[1][0000−0003−1723−9897], Grettel M. García[1][0000−0001−9713−300X],
Yenier T. Izquierdo[1][0000−0003−0971−8572], Cleber Oliveira[1],
Jefferson Alves de Sousa[1], Bernardo Florindo Mortari Rezende[1],
Bruno Novelli[1], and Marco A. Casanova[1,2][0000−0003−0765−9636]

[1] Instituto Tecgraf, PUC-Rio, Rio de Janeiro, 22451-900, RJ, Brazil
{melissa,ggarcia,ytorres,cleberoli,alvesjefferson,bernardofmr,bnovelli}
@tecgraf.puc-rio.br
[2] Departamento de Informática, PUC-Rio, Rio de Janeiro, 22451-900, RJ, Brazil
casanova@inf.puc-rio.br

Abstract. This paper first presents DANKE, a data and knowledge management platform that allows users to submit keyword queries to a centralized database. DANKE uses a knowledge graph to provide a semantic view of the centralized database in a vocabulary familiar to the users. The paper then describes DANKE-U, a specialized module that enables DANKE to handle unstructured data, including scientific and engineering documents. Lastly, the paper presents a real use case from the oil and gas industry, involving technical/scientific documents.

Keywords: Database Keyword Search · Knowledge Graphs · Data Integration · Engineering Data · Scientific Data · Unstructured Data.

1 Introduction

This paper first presents DANKE, a data and knowledge management platform that allows users, without technical training, to submit keyword queries to a centralized database. DANKE's primary motivation is to democratize access to data and documents, originally dispersed across different data sources, without requiring users to write scripts or depend on the development of applications that provide forms for querying this data.

DANKE helps construct a centralized database through a data integration process with the following major steps: (i) data extraction from the original data sources; (ii) transformation and enrichment of these data; (iii) loading the data into the centralized database; (iv) indexing the data. DANKE uses a knowledge graph to provide a semantic view of the centralized database in a vocabulary familiar to the users. The knowledge graph is the basis of the search engine, which matches the keywords users submit with the concepts of the graph, and uses the matchings to compile automatically an SQL (or RDF) query on the centralized database.

The paper then describes DANKE-U, which enables DANKE to handle unstructured data, including scientific and engineering documents. DANKE-U is a collection of components that process unstructured data, extracting metadata, thumbnails, texts, images, named entities, and tables. With the help of this extension, DANKE's users can seamlessly search unstructured documents and relate them to structured data.

Lastly, the paper presents a real use case from the oil and gas industry, where DANKE is applied to technical/scientific documents and structured data about subsea production systems, that play a critical role in offshore oil and gas extraction. In this context, technical/scientific documents are challenging to interpret: they may contain data organized in non-conventional tables that merge cells (between rows and columns); or they may take the form of engineering drawings, etc. Thus, off-the-shelf tools are of limited help to analyze such documents.

The paper is organized as follows. Section 2 contains a very brief discussion on related work. Section 3 introduces DANKE. Section 4 outlines DANKE-U. Section 5 covers a use case. Finally, Section 6 contains the conclusions.

2 Related Work

This section briefly covers related work in areas directly connected to data integration and database keyword search.

Data Integration. Classic data integration is usually divided into the major sub-problems of data retrieval, data fusion, schema alignment, and entity linkage [3]. The progress the data integration community has made in addressing these challenges in the context of big data integration was explored in [5]. The ties between machine learning and data integration were discussed in [4, 20].

In the context of engineering data, Nguyen et al. [17] described the development of a framework for data integration to optimize the remote operations of offshore wind farms. Espinola et al. [6] presented an approach that integrates data from mixed/augmented reality tools and embedded intelligent maintenance systems to support operators/technicians during maintenance tasks, providing easier access, understanding, and comprehension of information from different systems. Urbina Coronado et al. [22] described how to integrate data from machine tools with production data collected by a Manufacturing Execution System (MES) to monitor process output, consumable usage, and operator productivity.

Database Keyword Search Engines. Early relational keyword-based query processing tools [1, 18, 19] explored the foreign/primary keys declared in the relational schema to compile a keyword-based query into an SQL query with a minimal set of join clauses based on the notion of candidate networks.

State-of-the-art content-based image retrieval strategies [9, 13] assume that high-dimensional vectors represent images, created using Deep Learning techniques. Alternatively, the tool might transform both the text and the images (or, in fact, any other media as well) into a single high-dimensional vector, as in cross-modal retrieval techniques [2, 25].

Tautkute et al. [21] proposed a multimodal search engine in the fashion domain that retrieves items aesthetically similar to a query composed of image and textual inputs. Vo et al. [23] addressed a similar problem, where a query is specified in the form of an image and a text description of desired modifications to the input image. Yu et al. [24] introduced a multimodal model for understanding commerce topics, where the content can be an image, a text, or a combination of both.

DANKE [12] is an evolution of earlier tools [8, 10, 11]. DANKE-U enables DANKE to handle unstructured data, including scientific and engineering documents, as mentioned in the introduction.

3 A Brief Overview of DANKE

A *keyword query* in DANKE is just a list of terms, or *keywords*, that the user wants to search the database for, and may include reserved terms, such as "<", ">", "*between*". In relational jargon, an *answer* to a keyword query is formatted as a table whose columns (or column names) contain the keyword matches. The answer may be the result of joining several database tables, that is, an answer to a keyword query does not need to be constructed out of a single table.

DANKE has three main components (see Figure 1): *Storage Module*; *Preparation Module*; and *Knowledge Extraction Module.*

The *Storage Module* houses a relational (or RDF) database, constructed from various data sources. The database is described by a knowledge graph (KG), which is independent of the data model of the underlying database. The *Storage Module* also holds data indices required to support keyword search and other services.

The *Preparation Module* provides tools for creating the knowledge graph and for constructing and updating the centralized database through a pipeline responsible for a typical data integration process. This includes indexing data, collecting data from data sources, transforming and enriching data, and ingesting data into the database. This module enables customization of the pipeline in the most convenient manner for each case. The knowledge graph KG is defined by de-normalizing the schemas of the underlying databases, enriching KG with metadata that help interpret the data, and indicating which properties will have their values indexed.

The *Knowledge Extraction Module* uses the technology described in [7, 8, 10] and explores the knowledge graph and the data indices to compile a keyword query into an SQL (or SPARQL) query that returns data that best match the keywords. It features an algorithm [8] that accepts as input a keyword query Q_K, the knowledge graph KG, and the indices, and: (1) finds matches with the keywords in Q_K; (2) creates a conceptual query Q_C by exploring the keyword matches found and KG; (3) compiles Q_C into an SQL (or SPARQL) query Q_S, which is then executed.

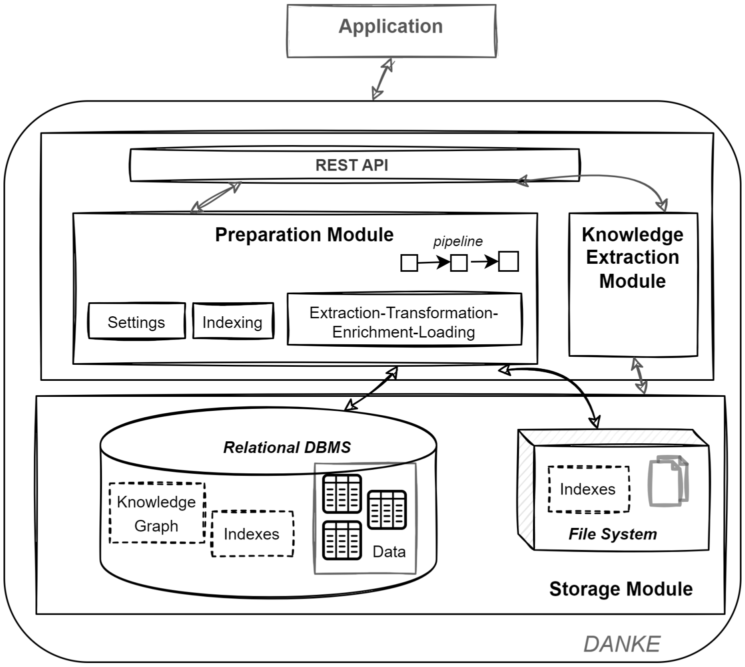


Fig. 1. Architectural Overview of the DANKE Platform

4 DANKE-U

To handle unstructured data, such as technical/scientific documents, DANKE creates pipelines using DANKE-U, as illustrated in Figure 2. DANKE-U comprises a set of components that process unstructured data, extracting metadata, thumbnails, texts, images, named entities and tables, for example.

A pipeline encompasses several forms of text document processing, automatically extracting information and enriching the database with new data and relationships. If it is necessary to search through the textual content of the file, the pipeline must include text indexing, enabling the search engine to match keywords with the textual content of the file.

In relational mode, documents and the associated data are recorded in a table, including their identification, file path, thumbnail, textual content, and metadata such as title, author, date, and type. Furthermore, other tables that relate documents to additional information may also be used.

An example of text document processing is extracting attribute values from specific entities using advanced table processing techniques facilitated by the "Table Extraction" component of DANKE-U. This component can be tailored to extract data from various types of tables, as illustrated on the right-hand side of Figure 2, and executes the processing in three stages: (i) Collect and Settings: involves reading a folder containing PDF documents and a configuration file defining the metadata and tables to be investigated in the documents. The configuration determines domain-specific structures (e.g., applying regular expressions to identify entity and attribute names) used to select tables and attributes for extraction, along with formatting options for each attribute in the output. (ii) Extractor: extracts the attributes and tables defined in the previous stage. The process utilizes the Camelot library [3], capable of extracting tables with column and line separators (lattices, illustrated in Figures 4 and 5 of the appendix) as well as those without clear separators (stream, illustrated in Figure 6 of the appendix). From the Camelot output, the component identifies and extracts the attributes based on the configuration. (iii) Cleaner and Formatter: transforms the extracted data into tabular format, which can be stored using spreadsheets, CSV, or database tables. The user can personally and configure the output format through the configuration file.

Another relevant example of text processing is Named Entity Recognition. Assuming the closed-world hypothesis, where there is a known set of values representing an entity, this technique can identify which documents mention a specific entity in their text. Hence, it is possible to relate documents to entities, storing this relationship in the database.

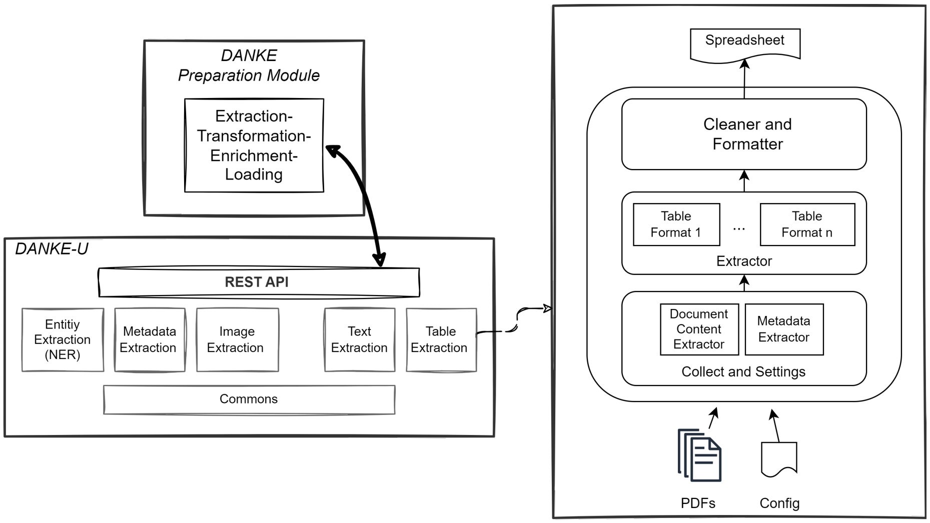


Fig. 2. Architectural Overview of the DANKE-U Module

[3] https://camelot-py.readthedocs.io/en/master/index.html

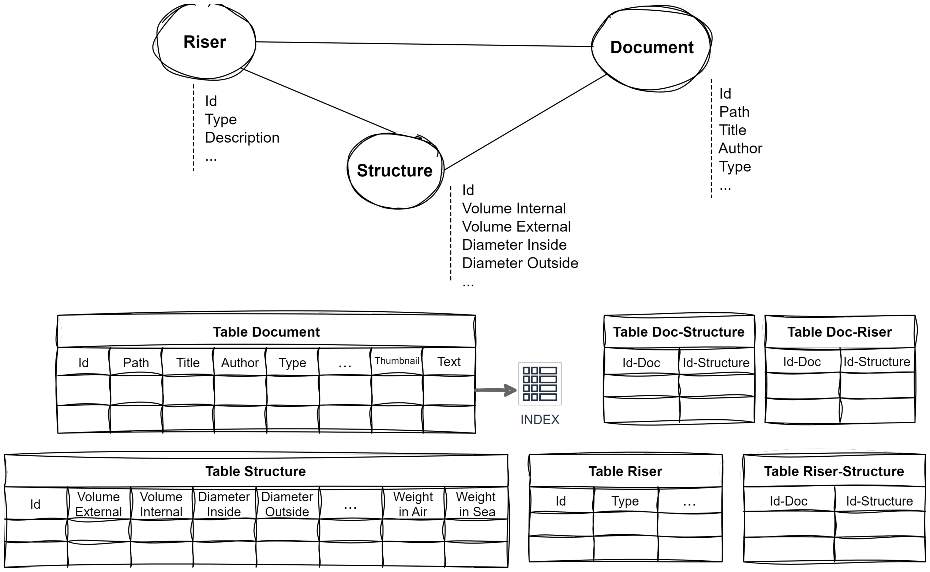


Table Document

Id	Path	Title	Author	Type	...	Thumbnail	Text

Table Doc-Structure

Id-Doc	Id-Structure

Table Doc-Riser

Id-Doc	Id-Structure

Table Structure

Id	Volume External	Volume Internal	Diameter Inside	Diameter Outside	...	Weight in Air	Weight in Sea

Table Riser

Id	Type	...

Table Riser-Structure

Id-Doc	Id-Structure

Fig. 3. Extracting data from tables

5 Use Case

This section introduces the domain in which DANKE is employed to handle scientific and technical documents, illustrating the approach with real-life cases in subsea production systems of offshore oil and gas extraction.

Subsea production systems are composed of increasingly complex structures to meet the growing demands of large deep-water oil fields. Moreover, maintaining the physical integrity of several subsea components faces an additional challenge from their remote and hard-to-access locations. Nonetheless, tracking their operational health is crucial to guarantee environmental safety, reduce the high costs of an unplanned operation shutdown, and meet regulatory demands.

To accomplish these tasks, subsea experts need efficient access to consistent and up-to-date information about equipment designs, layouts and locations, inspection reports, maintenance activities, and other daily-generated data. However, throughout the life cycle of the subsea system, a large variety of data are generated and stored in various places, creating a complex environment. In this context, accessing, searching, navigating, and integrating this data becomes critical for decision-making. However, these actions are often costly and require substantial time from experts, technicians, and engineers.

Figure 3 illustrates a fragment of the knowledge graph that represents the subsea domain, encompassing: (i) a riser, a conductor pipe connected to offshore production platforms and the subsea flowlines, manifolds, and wellhead; (ii) a structure detailing the specific attributes for which a riser is designed, such as volume, diameter, and weight; and (iii) documents containing engineering and scientific data related to the risers and structures. Figure 3 also displays the resulting relational database, featuring tables to store data about risers, structures, documents, and the relationships among them. The appendix contains three illustrations of tables with a layout commonly found in PDFs in a real-world scenario; however, for confidentiality reasons, the values of the structures are fictitious.

The following use cases illustrate the approaches used in this domain for searching the documents.

Use Case 1 - A keyword query over document metadata. Suppose the user enters the keyword query "document type technical drawing". DANKE compiles this keyword query into an SQL query involving the table "Document" and its attribute "type", filtering for the attribute value "technical drawing", and responds with documents of the type technical drawing. Document metadata may have been automatically extracted from the PDF file using the "Metadata Extraction" component of DANKE-U, or it could have been manually entered by a specialist using an Electronic Document Management (EDM) system.

Use Case 2 - A keyword query over document text. Suppose the user enters the keyword query "document diagram yyyy.xxxx". DANKE compiles this keyword query into an SQL query to find documents of type "diagram" that contain the text "yyyy.xxxx". To locate the text "yyyy.xxxx" within diagrams, represented as images in PDFs, it is necessary to extract the text from the images using OCR, a task performed by the "Text Extraction" component of DANKE-U. Additionally, the pipeline created in DANKE's Preparation Module must store and index the text extracted from the PDF.

Use Case 3 - A keyword query relating data and documents. Suppose the user enters the keyword query "structure document diagram yyyy.xxxx". DANKE compiles this keyword query into an SQL query similar to the previous example but includes the relationships between structures and documents. To establish this relationship, document texts must be processed to identify those that mention structure identifiers. This task is accomplished by the "Entity Extraction" component of DANKE-U. Using the structure identifiers, previously known and stored in the table "Structure", the component "Entity Extraction" processes the document texts and returns a table associating the structure identifiers with the document identifiers.

Use Case 4 - A keyword query involving attribute values defined in documents. Suppose the user enters the keyword query "riser xxxx-yyyy structure diameter." DANKE compiles this keyword query into an SQL query that retrieves the diameter of the structures for the riser "xxxx-yyyy". Some attributes of the structure are not originally present in a structured form (e.g., in a relational database) and had to be extracted through a sophisticated process that interprets various table formats found in documents. This extraction is performed by the "Table Extraction"component of DANKE-U, customized to process complex tables. Figures 4, 5 and 6 in appendix illustrates a real-life scenario. However, for confidentiality reasons, the values presented in the tables are fictitious. The classical format of a table places each attribute in a separate cell, as in Figure 4. In this example, the figure shows a table with diameter *4* for the structure identified as *001.00024*. However, in some cases, when the attribute value is the same for different structures, the attribute value is presented only once by merging cells. Figure 5 presents a slightly more complex table to process, as it contains the same diameter value (*4*) in a merged cell, for three different structures: *001.00021, 001.00022, and 001.00023*. Figure 6 depicts a table with a different layout but also containing the diameter value 4 of structure *001.00024*. Furthermore, a page break may occur within the same table.

6 Conclusions and Directions for Future Work

The paper first presented the overall architecture of DANKE, a data and knowledge extraction platform, based on a search engine. Then, it described DANKE-U, that enables DANKE to handle unstructured data, including scientific and engineering documents. Finally, the paper presented a use case from the oil and gas industry, involving technical/scientific documents, processed using DANKE.

DANKE is currently being equipped with a Natural Language (NL) interface, constructed with the help of a Large Language Model (LLM), to help users locate data using NL questions, covering language constructs not supported by keyword queries [14–16]. The final goal is to explore the combination of controllable database keyword search with LLM freedom to create a *trustworthy application* that generates *reliable answers.*

As for future work, the application area involves a large variety of documents, including complete engineering drawings. This demand calls for the development of other specialized set of components to be included in DANKE-U.

ACKNOWLEDGEMENTS

This work was partly funded by FAPERJ under grant E-26/202.818/2017; by CAPES under grants 88881.310592-2018/01, 88881.134081/2016-01, and 88882.164913/2010-01; by CNPq under grant 302303/2017-0; and by Libra Consortium (Petrobras, Shell Brasil, Total Energies, CNOOC, CNPC, and PPSA) within the ANP R&D levy as a commitment to research and development investments.

References

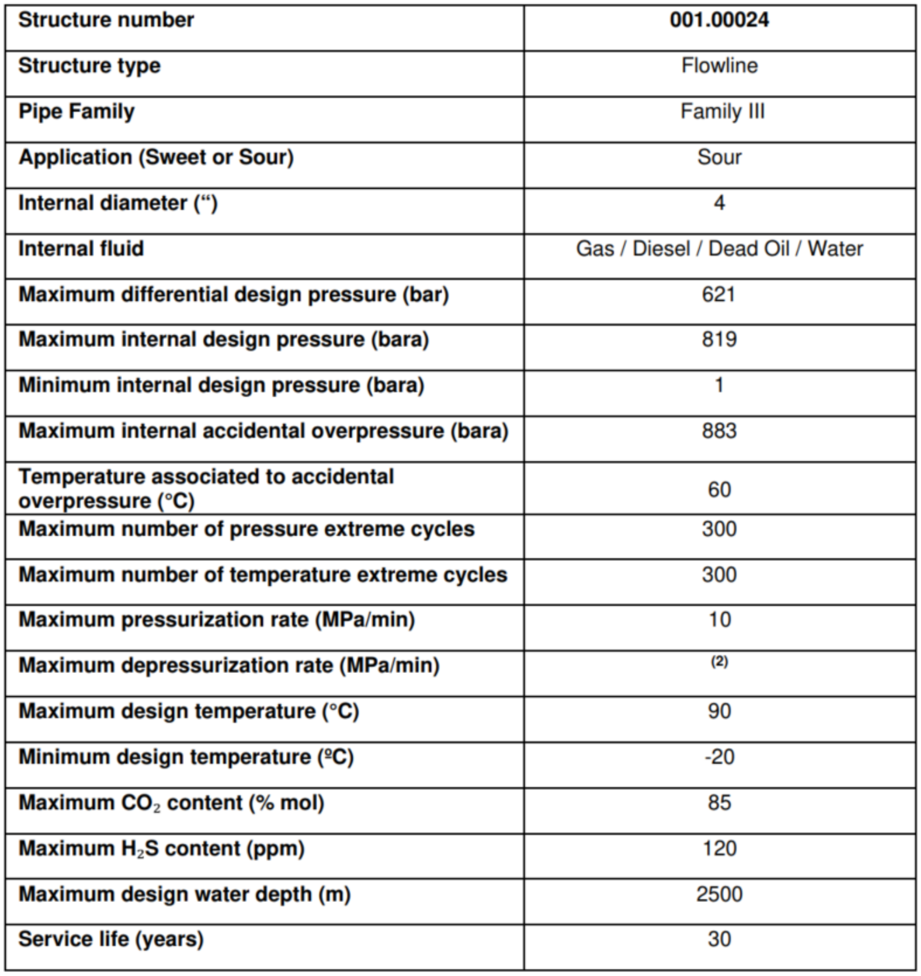
1. Bergamaschi, S., Guerra, F., Interlandi, M., Trillo-Lado, R., Velegrakis, Y.: Combining user and database perspective for solving keyword queries over relational databases. Information Systems **55**, 1–19 (2016). https://doi.org/10.1016/j.is.2015.07.005

2. Costa Pereira, J., Coviello, E., Doyle, G., Rasiwasia, N., Lanckriet, G., Levy, R., Vasconcelos, N.: On the role of correlation and abstraction in cross-modal multimedia retrieval. Transactions of Pattern Analysis and Machine Intelligence **36**(3), 521–535 (March 2014). https://doi.org/10.1109/TPAMI.2013.142
3. Doan, A., Halevy, A.Y., Ives, Z.G.: Principles of Data Integration. Morgan Kaufmann, San Francisco, CA, USA, 1st edn. (2012). https://doi.org/10.1016/C2011-0-06130-6
4. Dong, X.L., Rekatsinas, T.: Data integration and machine learning: A natural synergy. In: Proceedings of the 2018 International Conference on Management of Data. p. 1645–1650. SIGMOD '18, Association for Computing Machinery, New York, NY, USA (2018). https://doi.org/10.1145/3183713.3197387
5. Dong, X.L., Srivastava, D.: Big data integration. In: 2013 IEEE 29th International Conference on Data Engineering (ICDE). pp. 1245–1248 (2013). https://doi.org/10.1109/ICDE.2013.6544914
6. Espíndola, D.B., Fumagalli, L., Garetti, M., Pereira, C.E., Botelho, S.S., Ventura Henriques, R.: A model-based approach for data integration to improve maintenance management by mixed reality. Computers in Industry **64**(4), 376–391 (2013). https://doi.org/10.1016/j.compind.2013.01.002
7. García, G.M.: A Keyword-based Query Processing Method for Datasets with Schemas. Ph.D. thesis, Thesis presented to the Graduate Program in Informatics, PUC-Rio (2020)
8. García, G.M., Izquierdo, Y.T., Menendez, E., Dartayre, F., Casanova, M.A.: Rdf keyword-based query technology meets a real-world dataset. In: Proceedings of the International Conference on Extending Database Technology. pp. 656–667. OpenProceedings.org (2017). https://doi.org/10.5441/002/edbt.2017.86
9. Hameed, I.M., Abdulhussain, S.H., Mahmmod, B.M.: Content-based image retrieval: A review of recent trends. Cogent Engineering **8**(1), 1927469 (2021). https://doi.org/10.1080/23311916.2021.1927469
10. Izquierdo, Y.T., García, G.M., Menendez, E.S., Casanova, M.A., Dartayre, F., Levy, C.H.: Quiow: a keyword-based query processing tool for rdf datasets and relational databases. In: International Conference on Database and Expert Systems Applications (DEXA). pp. 259–269. Springer (2018). https://doi.org/10.1007/978-3-319-98812-2_22
11. Izquierdo, Y.T., García, G.M., Menendez, E., Leme, L.A.P., Neves, A., Lemos, M., Finamore, A.C., Oliveira, C., Casanova, M.A.: Keyword search over schema-less rdf datasets by sparql query compilation. Information Systems **102**, 101814 (2021). https://doi.org/10.1016/j.is.2021.101814
12. Izquierdo, Y.T., García, G.M., Lemos, M., Novello, A., Novelli, B., Damasceno, C., Leme, L.A.P.P., Casanova, M.A.: A platform for keyword search and its application for covid-19 pandemic data. Journal of Information and Data Management **12**(5) (Nov 2021). https://doi.org/10.5753/jidm.2021.1904, https://sol.sbc.org.br/journals/index.php/jidm/article/view/1904
13. Li, X., Yang, J., Ma, J.: Recent developments of content-based image retrieval (cbir). Neurocomputing **452**, 675–689 (2021). https://doi.org/10.1016/j.neucom.2020.07.139
14. Nascimento, E.R., Garcia, G.M., Feijó, L., Victorio, W.Z., Lemos, M., Izquierdo, Y.T., Garcia, R.L., Leme, L.A.P., Casanova, M.A.: Text-to-sql meets the real-world. In: (Accepted to the 26th Int. Conf. on Enterprise Info. Sys.) (2024)
15. Nascimento, E.R., Garcia, G.M., Victorio, W.Z., Lemos, M., Izquierdo, Y.T., Garcia, R.L., Leme, L.A.P., Casanova, M.A.: A family of natural language inter-

faces for databases based on chatgpt and langchain (short paper). In: Companion Proceedings of the 42nd International Conference on Conceptual Modeling: Posters and Demos co-located with ER 2023, Lisbon, Portugal, November 06-09, 2023. CEUR Workshop Proceedings, vol. 3618 (2023), https://ceur-ws.org/Vol-3618/pd_paper_1.pdf

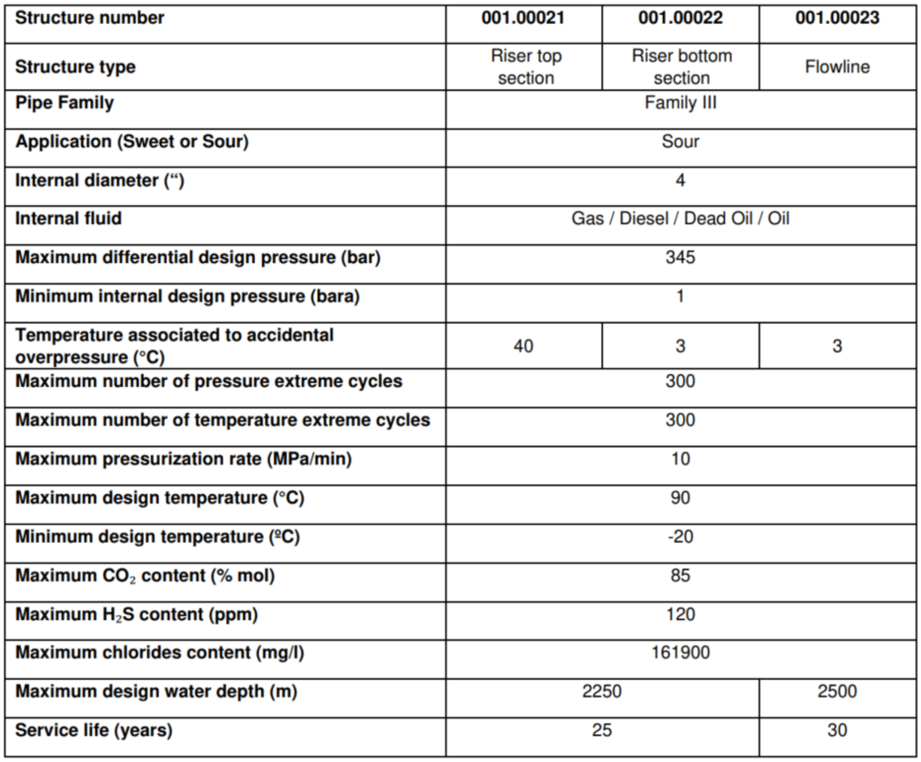
16. Nascimento, E.R., Izquierdo, Y.T., Garcia, G.M., Coelho, G., Feijó, L., Lemos, M., Leme, L.A.P., Casanova, M.A.: My database user is a large language model. In: (Accepted to the 26th Int. Conf. on Enterprise Info. Sys.) (2024)
17. Nguyen, T.H., Prinz, A., Friisø, T., Nossum, R., Tyapin, I.: A framework for data integration of offshore wind farms. Renewable Energy **60**, 150–161 (2013). https://doi.org/10.1016/j.renene.2013.05.002
18. de Oliveira, P., da Silva, A., de Moura, E.: Ranking candidate networks of relations to improve keyword search over relational databases. In: 2015 IEEE 31st International Conference on Data Engineering. pp. 399–410 (2015). https://doi.org/10.1109/ICDE.2015.7113301
19. Ramada, M.S., Silva, J.C., Leitão-Júnior, P.S.: From keywords to relational database content: A semantic mapping method. Information Systems **88**, 101460 (2020). https://doi.org/10.1016/j.is.2019.101460
20. Stonebraker, M., Ilyas, I.F.: Data integration: The current status and the way forward. IEEE Data Eng. Bull. **41**, 3–9 (2018), https://api.semanticscholar.org/CorpusID:49407081
21. Tautkute, I., Trzciński, T., Skorupa, A.P., Brocki, Ł., Marasek, K.: Deepstyle: Multimodal search engine for fashion and interior design. IEEE Access **7**, 84613–84628 (2019). https://doi.org/10.1109/ACCESS.2019.2923552
22. Urbina Coronado, P.D., Lynn, R., Louhichi, W., Parto, M., Wescoat, E., Kurfess, T.: Part data integration in the shop floor digital twin: Mobile and cloud technologies to enable a manufacturing execution system. Journal of Manufacturing Systems **48**, 25–33 (2018). https://doi.org/10.1016/j.jmsy.2018.02.002, special Issue on Smart Manufacturing
23. Vo, N., Jiang, L., Sun, C., Murphy, K., Li, L.J., Fei-Fei, L., Hays, J.: Composing text and image for image retrieval-an empirical odyssey. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 6439–6448 (2019). https://doi.org/10.1109/CVPR.2019.00660
24. Yu, L., Chen, J., Sinha, A., Wang, M., Chen, Y., Berg, T.L., Zhang, N.: Commercemm: Large-scale commerce multimodal representation learning with omni retrieval. In: Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining. pp. 4433–4442 (2022). https://doi.org/10.1145/3534678.3539151
25. Zeng, D., Yu, Y., Oyama, K.: Deep triplet neural networks with cluster-cca for audio-visual cross-modal retrieval. ACM Trans. Multimedia Comput. Commun. Appl. **16**(3) (2020). https://doi.org/10.1145/3387164

7 Appendix



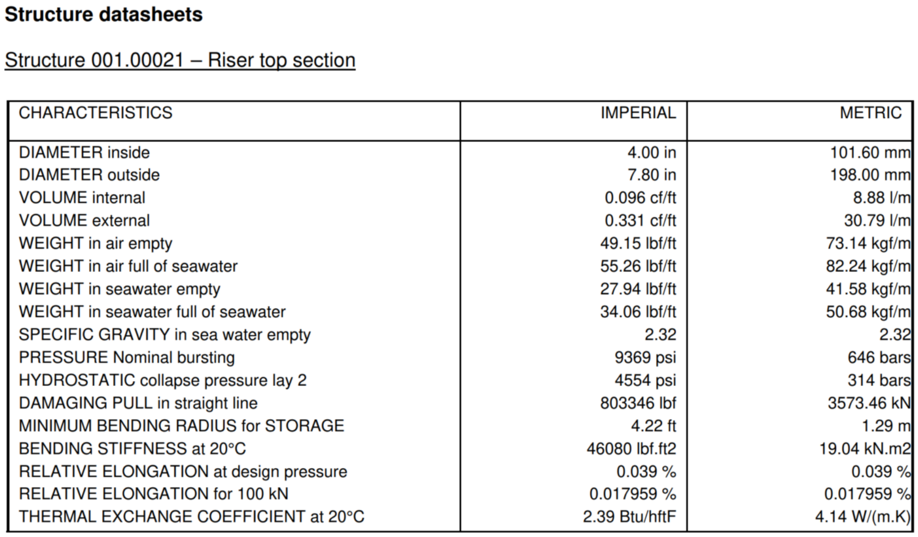
Structure number	**001.00024**
Structure type	Flowline
Pipe Family	Family III
Application (Sweet or Sour)	Sour
Internal diameter (“)	4
Internal fluid	Gas / Diesel / Dead Oil / Water
Maximum differential design pressure (bar)	621
Maximum internal design pressure (bara)	819
Minimum internal design pressure (bara)	1
Maximum internal accidental overpressure (bara)	883
Temperature associated to accidental overpressure (°C)	60
Maximum number of pressure extreme cycles	300
Maximum number of temperature extreme cycles	300
Maximum pressurization rate (MPa/min)	10
Maximum depressurization rate (MPa/min)	(2)
Maximum design temperature (°C)	90
Minimum design temperature (ºC)	-20
Maximum CO_2 content (% mol)	85
Maximum H_2S content (ppm)	120
Maximum design water depth (m)	2500
Service life (years)	30

Fig. 4. Example 1 of table format (with fictitious data)



Structure number	001.00021	001.00022	001.00023
Structure type	Riser top section	Riser bottom section	Flowline
Pipe Family	Family III		
Application (Sweet or Sour)	Sour		
Internal diameter (“)	4		
Internal fluid	Gas / Diesel / Dead Oil / Oil		
Maximum differential design pressure (bar)	345		
Minimum internal design pressure (bara)	1		
Temperature associated to accidental overpressure (°C)	40	3	3
Maximum number of pressure extreme cycles	300		
Maximum number of temperature extreme cycles	300		
Maximum pressurization rate (MPa/min)	10		
Maximum design temperature (°C)	90		
Minimum design temperature (ºC)	-20		
Maximum CO_2 content (% mol)	85		
Maximum H_2S content (ppm)	120		
Maximum chlorides content (mg/l)	161900		
Maximum design water depth (m)	2250		2500
Service life (years)	25		30

Fig. 5. Example 2 of table format (with fictitious data)



Structure datasheets

Structure 001.00021 – Riser top section

CHARACTERISTICS	IMPERIAL	METRIC
DIAMETER inside	4.00 in	101.60 mm
DIAMETER outside	7.80 in	198.00 mm
VOLUME internal	0.096 cf/ft	8.88 l/m
VOLUME external	0.331 cf/ft	30.79 l/m
WEIGHT in air empty	49.15 lbf/ft	73.14 kgf/m
WEIGHT in air full of seawater	55.26 lbf/ft	82.24 kgf/m
WEIGHT in seawater empty	27.94 lbf/ft	41.58 kgf/m
WEIGHT in seawater full of seawater	34.06 lbf/ft	50.68 kgf/m
SPECIFIC GRAVITY in sea water empty	2.32	2.32
PRESSURE Nominal bursting	9369 psi	646 bars
HYDROSTATIC collapse pressure lay 2	4554 psi	314 bars
DAMAGING PULL in straight line	803346 lbf	3573.46 kN
MINIMUM BENDING RADIUS for STORAGE	4.22 ft	1.29 m
BENDING STIFFNESS at 20°C	46080 lbf.ft2	19.04 kN.m2
RELATIVE ELONGATION at design pressure	0.039 %	0.039 %
RELATIVE ELONGATION for 100 kN	0.017959 %	0.017959 %
THERMAL EXCHANGE COEFFICIENT at 20°C	2.39 Btu/hftF	4.14 W/(m.K)

Fig. 6. Example 3 of table format (with fictitious data)