Received December 1, 2021, accepted December 20, 2021, date of publication December 28, 2021, date of current version January 11, 2022.

Digital Object Identifier 10.1109/ACCESS.2021.3139071

Comparative Analysis of Data Modeling Design Tools

GONÇALO CARVALHO[1], SERGII MYKOLYSHYN[1], BRUNO CABRAL[1], JORGE BERNARDINO[1,2], (Member, IEEE), AND VASCO PEREIRA[1]
[1]Centre for Informatics and Systems, Department of Informatics Engineering, University of Coimbra, 3004-531 Coimbra, Portugal
[2]Polytechnic of Coimbra, ISEC, 3030-199 Coimbra, Portugal

Corresponding author: Gonçalo Carvalho (gcarvalho@dei.uc.pt)

This work was supported by the European Regional Development Fund (FEDER), through the Competitiveness and Internationalization Operational Programme (COMPETE 2020) of the PORTUGAL 2020 Framework Project InfraCrit with Nr. 039555 under Grant POCI-01-0247-FEDER-039555.

ABSTRACT Conceptual modeling describes the physical or social aspects of the world abstractly, encompassing the interpretation of data production, gathering, visualization, and analysis. The quality of the data analysis system will limit the excellence of any decision-making process. Thus, accurately specifying the physical data model is essential. The primary goal of our work is to compare tools that can create this physical model. We recognize several types of data models, but we only include the relational data model. We evaluate free and commercial data modeling tools. But it is challenging to decide how to compare them and which elements are crucial. We propose a new approach for software tools' evaluation based on the Business Readiness Rating (BRR) model and the OSSpal evaluation methodology. In this work, we show that this new methodology can be tailored to the needs of each individual developer or team, thus providing proper and meaningful results. Also, by applying this hybrid approach to the evaluation of data modelling tools, we show it can robustly handle the bias from lesser relevant evaluation categories.

INDEX TERMS Data modeling, design tools, database management systems, data modeling tools.

I. INTRODUCTION

A *data model* is a set of concepts that can describe the data structure and operations on a database [1]. In this paper we address the term in the sense of conceptual, logical, and physical data model. These data structures include objects, relationships between these objects, and rules that define how data is organized. Determining the business needs will lead to the data model. The business stakeholders' feedback is crucial to define rules and requirements to be incorporated into the design of a new system or adapted in an iteration of an existing one [2].

Data modeling creates a visual representation of either a whole information system or parts of it to reveal connections between data points and structures. This is the first step in data design, and Simsion and Witt [3] defined it as "*a design activity which classifies information in an organized way and defines their relations.*" Therefore, the process of data modeling involves professional data modelers working closely with business stakeholders, as well as potential users of the information system [4].

The Entity Relationship (ER) model is one of the fundamental conceptual data models, which is usually associated with relational databases. This model is the focus of this work because it is the model most often adopted at this stage of conceptual design. An Entity Relationship Diagram (ERD) is a drawing that communicate the relationships between entities [5]. An entity is a "thing" or "object" in the real world that is distinguishable from other objects. Relationships have cardinalities, attributes, and constraints. The cardinality of a relationship indicates the number of occurrences between two entities [6].

Any database architect needs to work with a tool that allows an easy data model design. Such a choice will have a direct impact on the project quality. The design tool must be suitable to represent a data model, be easy to use, and support a different number of Database Management Systems (DBMS). In addition, the tool should allow defining constraints such as Primary Key (PK), or Not Null (NN), and produce a Structured Query Language (SQL) script from the representation of data objects created by the user.

The associate editor coordinating the review of this manuscript and approving it for publication was Vlad Diaconita.

To the best of our knowledge, this is the first work that evaluates data modeling tools using a formal evaluation methodology. However, this is not an exhaustive assessment of all available features and did not consider the business context where developers will use the tools. The major contributions of this work are the following:

- Provide a background on conceptual modeling and data modeling tools;
- Propose a new methodology for evaluating software modeling tools based on the BRR model and the OSSpal methodology;
- Demonstrate an evaluation of free and commercial data modeling tools. This evaluation reflects the usefulness of the tools and their level of productivity for users.

We organized the rest of this paper as follows. In Section II, we review other published approaches to software evaluation. In Section III, we provide the background on conceptual models. In Section IV, we describe and analyze each tool. In Section V, we explain the evaluation methodology used in the tools' assessment. In Section VI, we evaluate the selected tools. Finally, in Section VII, we describe the main conclusions.

II. RELATED WORK

In this section, we will review other approaches that evaluated software quality through different methodologies.

In 1988, Saaty [7] introduced the Analytic Hierarchy Process (AHP). Used in complex decisions, it structures and analyzes the choices by weighting the decision criteria. Different authors applied it in several areas, such as government, business, industry, healthcare, and education. AHP assists decision-makers in identifying a decision that adequately satisfies their goal and their understanding of the problem. AHP is a structured framework that comprehensively and rationally quantifies distinct elements. It steers the decision according to the primary goals while also evaluating alternative solutions. However, the authors did not develop this technique considering decision-making for software tools.

In 2005, SpikeSource, the Center for Open-Source Investigation at Carnegie Mellon West, and Intel Corporation created the BRR model [8], which lets IT managers promptly deliver informed and educated decisions about open-source software. They developed BRR to be a complete, simple, adaptable, and consistent model to help choose the right software. The authors evaluated open-source software according to 12 categories: functionality, usability, quality, security, performance, scalability, architecture, support, documentation, adoption, community, and professionalism. They divided the BRR into four phases: i) *A quick assessment*, to identify a list of components, measure each component, and remove any component that does not fit the user requirements. ii) *Target usage assessment*, to define the 12 category weights according to importance (from 1 to 12). We should choose the top seven (or less) and assign a percentage of importance to each one, totaling 100%. Set the metric weights within each category according to their importance, also summing 100%. iii) *Data collection and processing*, to collect data for each metric in each category, and calculate the applied weighting for each metric. iv) *Data translation*, to calculate the final BRR score.

In 2017, Wasserman *et al.* [9] proposed an extension to the BRR, which originated the OSSpal open-source software assessment methodology. The authors' motivation was to solve the shortcomings of the original approach, such as i) some bias in the BRR score, according to the evaluator knowledge of the project, available documentation and commercial support; ii) the lack of details provided by a single numeric score; iii) the reduced amount of adequate software to undergo this evaluation; and iv) the prime consideration of opinions of others, including both peers and experts. Thus, the authors introduced some changes, among other minor improvements, i) because the BRR only used the top seven ranked categories, which may leave out of the analysis important categories to other evaluators, Wasserman *et al.* condensed the Categories from 12 into seven, which the authors state to be the most important in open-source software: Functionality, Operational Software Characteristics, Support and Service, Documentation, Software Technology Attributes, Community and Adoption, and Development Process; ii) removed the final score calculation formula; iii) created a list of adequate software for evaluation and grouped them into categories based on the software taxonomy produced annually by the International Data Corporation (IDC) [10]; and iv) developed a website for users to use and rank the software tools to surpass the impossibility to assess which tool is better amongst two or three with the same feature score.

We resorted to several databases, DBLP - computer science bibliography, Google Scholar, and IEEE Xplore, to find other works regarding the use of these evaluation schemas.

From 2017 to 2019 several papers used the OSSpal methodology in different research areas, such as Business Intelligence Tools [11], [12], Data Mining Tools [13], E-commerce Tools [14], Project Management Tools [15]–[18], and NoSQL DBMS [19]. These works used the same implementation and analyzed three or four tools. The results when evaluating the same tool were different, for example, OpenProject 4.5 and 3.45 in [16] and [15] respectively, and ProjectLibre 3.82 and 3.6 in [16] and [17] respectively. This emphasizes the subjective approach of the evaluation, because the categories' weights and the number of analyzed characteristics encompassing the Functionality category were different while evaluating the same tools. Also, all these works had a final score, so they were an assessment through the BRR model assessment rather than an evaluation by the OSSpal methodology.

Nevertheless, the penalty for high scores in less critical measures is one of the OSSpal methodology shortcomings. Ultimately, it will depend on the person evaluating the software products, which can lead to some bias scoring.

In our approach, we did not consider AHP for our assessment due to its limitations and weaknesses on evaluating software. Karthikeyan *et al.* [20] listed several, such as it can settle just direct models, inconsistencies in positioning, and human emotions associated to past experiences. We will use the BRR model, with the OSSpal Categories, which are more adapted to the tools under evaluation, and we introduce some changes by providing a broader range of values for the evaluation of the features. We also evaluate a significantly higher amount of tools (17), both free and commercial, in a different research area, and through a survey to professionals, we aimed to remove the subjectiveness of our evaluation.

III. BACKGROUND IN CONCEPTUAL MODELING

In this section, we address the Conceptual Model (CM) topic, identifying key features and languages.

CM describes the physical or social aspects of the world abstractly. The result of a proper and rigorous CM design is a functionally richer, less error-prone, adequately attuned, able to adapt to varying user requirements, and less expensive system [21]. Thus, designing the CM at the beginning of the development cycle should be mandatory. It will be easier to follow and adapt to user requirements and explore existing relationships between the concepts.

We adopt data models to manage and analyze data representing any information system. The data model is an essential element of the system development or database design processes. Although the data modeling phase embodies only a smaller dimension of the development effort, its influence on the eventual result is reasonably broader than any other phase. Moody *et al.* [22] mentioned the vast amount of alternative designs to address Conceptual Data Modeling (CDM). Several alternative models could provide accurate solutions, but may have quite distinct implications for database and system design. The process of data modeling is not simple, meaning it usually demands multiple iterations [23].

Thalheim [24] point different CM notations used to describe requirement specifications, such as the ER diagram [25], the Unified Modeling Language (UML) [26], which are the most regularly employed, and Business Process Modeling and Notation. Object-Oriented (OO) models are essentially expressive and more fitted to describe static and dynamic features of complex applications. The OO modeling field relates objects and attributes, whereas the real-world realm deals with things and properties [27]. Also, Model-driven Engineering methodology can use UML, SysML, or other modeling languages.

The ERD has existed for over 35 years, and is appropriate for data modeling because it is abstract and is easy to discuss and explain. It is easy to translate ER models to tables. The base of this type of modeling are entities, which hold information and relationships, defined as the associations between entities [28].

The primary advantages of CM for general systems and specifically for DBMS are:

- Provide a high-level perception of how the system will operate in the real world;
- Connect different mental models into a single CM design;
- Ensure that the data representation is accurate because missing fields in the database cause unreliable results;
- Get a clear understanding of the data that developers can manage when building the actual database;
- Identify any redundant or missing data;
- Make maintenance and upgrades faster, easier, and more affordable.

Next, we perform a qualitative analysis of data modeling tools.

IV. MODELING TOOLS

In this section, we will analyze different data modeling tools. The following list is not exhaustive, but representative of widely-used tools. We will include the conceptual model and the script generation (forward engineering) in the analysis's scope. However, we will not cover the physical deployment, access, and configuration of the database.

To create an ER model, which describes interrelated things of interest in a specific domain, it is necessary to specify **Entities**, which classify the *things of interest*, and **Relationships** that can exist between entities. In this model, the relationships amongst entities have a cardinality setting that illustrates the following options: one-to-one, one-to-many, zero-to-one, zero-to-many, and many-to-many. Despite this graphical representation, the physical data model produces a better comprehension of the relationships. The physical data model contains tables and the constraints define the cardinality of the relationship, such as Primary Key (PK), Foreign Key (FK), Unique (UQ), Not Null (NN), Check, Default values, as well as Indexes. Converting the design of the CM into a physical data model offers a straightforward interpretation of the model. In this paper, we address the representation of the relational model. However, we acknowledge other design descriptions, such as classes in an OO database or description of JSON schema for a document database.

After these steps, and with a proper definition of the business logic, the next step is *Forward Engineering*, which is the auto-generation of a SQL script from the created representation. This last step is crucial for any database engineer to minimize errors and the time spent creating a database.

From the list of *79 Data Modeling Tools Compared* [29] and the *20 Best Data Modeling Tools* [30], we selected those that allow the user to perform *Forward Engineering*, and only considered tools that continued receiving updates after 2018. We sorted them into four major product types: Online free tools, Online commercial tools, Desktop free tools, and Desktop commercial tools. In this context, the free type is understood as being free of charge, not for the possibility of accessing the original code. Despite other types could be considered, the four product types are able to broadly

TABLE 1. Modeling Tools.

Modeling Tools	
Product Type	**Name**
Online free tools (I)	Dbdesigner.id https://dbdesigner.id
	Onda http://onda.dei.uc.pt/v3
	WWW SQL Designer https://github.com/ondras/wwwsqldesigner
Online commercial tools (II)	Dbdesigner.net https://app.dbdesigner.net
	dbDiffo https://dbdiffo.com
	GenMyModel https://www.genmymodel.com
	Lucidchart https://lucid.app
	sqlDBM https://sqldbm.com
Desktop free tools (III)	MySQL Workbench - Community Version https://github.com/mysql/mysql-workbench
	pgModeler https://github.com/pgmodeler/pgmodeler
	Umbrello UML https://github.com/KDE/umbrello
Desktop commercial tools (IV)	dbSchema https://dbschema.com
	dbWrench http://www.dbwrench.com
	Erwin Data Modeler https://erwin.com/products/erwin-data-modeler
	Navicat https://www.navicat.com
	Oracle SQL Developer Data Modeler https://www.oracle.com/database/technologies/appdev/sqldeveloper-landing.html
	PowerDesigner https://www.sap.com/products/powerdesigner-data-modeling-tools.html

represent all the products available today and are well-aligned with the different users' and enterprises' needs.

Another relevant characteristic is that some tools have a visual representation of the conceptual data model, but others only show the physical data model. However, some have both types of visualization. On the one hand, the conceptual data model only describes the data and its relations. On the other, the physical data model displays' table structures, including column name, data type, and the constraints, such as PK, FK, and UQ, represent the relationships between tables.

In the following subsections, we perform a qualitative analysis of the selected tools (see Table 1). We provide a brief description of the key features, such as the release year; real-time collaboration options; generation of the physical model; the presence of *Reverse Engineering* (auto-generation of ER from SQL) and *Forward Engineering*; supported DBMS and data types; different constraints; the incorporation of CM; finally, the pros and cons are analyzed.

A. ONLINE FREE TOOLS

Online free tools work on every platform, receive constant updates from the community, and do not require installing the software. Also, every developer can contribute to any of these projects.

1) DBDESIGNER.ID

Dbdesigner.id (2019 version) is a database design tool for web developers and beginners, which started in 2019 under MIT License as a hobby project and is continuously under development. The authors' goal was to make database management accessible to everyone. They developed it in JavaScript, HTML, and CSS. To use Dbdesigner, it is necessary to create a new user account. With this tool, it is possible to share a link with a project contributor to work simultaneously. It does not allow a CM design. The tool does not provide reverse engineering, and only supports script generation for MySQL, which is a shortcoming considering that other tools support multiple DBMSs, thus causing a significant penalty in the assessment. For each column, the user can choose among different data types (tinyint, smallint, bigint, int, bigint, float, double, datetime, date, timestamp, char, varchar, binary, blob, text, JSON). It enables an option to specify constraints such as PK, FK, NN, and UQ. It is also possible to set a default value for each entry in a table. The user can select the referencing table and column name to create a relationship.

Pros: Link sharing for collaboration.

Cons: Needs registration and only uses MySQL.

2) ONDA

ONDA (Online Database Architect v3), is a database modeling tool developed by the Department of Informatics Engineering (DEI) at University of Coimbra (Portugal) community, with the first version released in 2014, developed with JavaScript, HTML, and CSS. It has fast loading time, but there is no real-time collaboration possibility. With Onda, it is possible to draw conceptual databases and visualize the physical data model. There is no reverse engineering option, but it is possible to perform forward engineering for the most popular DBMS like PostgreSQL, MySQL, Oracle, MariaDB, and SQLite. Each column can have different data types, such as boolean, integer, float, date, character, varchar, text, or BLOB. It is possible to add constraints for each column, such as PK, NN, Check Constraint (CH), and UQ. The FK are automatically added in the physical model through the designed relationships.

Pros: Zoom in/out, possibility to export the CM into different DBMS.

Cons: Some bugs on the physical model and script generation.

3) WWW SQL DESIGNER

Released in 2005, WWW SQL Designer (version 2.7), allows users to create and export data models to SQL scripts.

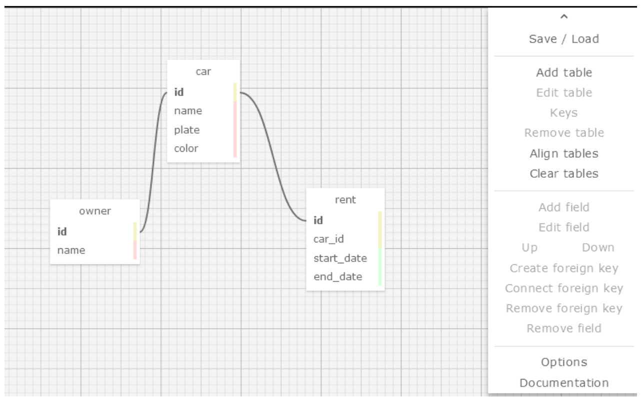


FIGURE 1. **Example of a database model at WWW SQL Designer.**

The interface has a mini-map for fast navigation (Fig. 1). It does not have real-time collaboration or representation of the CM. With this tool, it is possible to perform forward engineering for the MySQL DBMS, but it is impossible to perform reverse engineering. It supports many database constraints, such as PK, FK, UQ, and NN. There are different data types such as int, decimal, char, binary, BLOB, date, and time.

Pros: Many data types, and drag-and-drop features.

Cons: No real-time collaboration, no representation of the CM, and only exports MySQL scripts.

B. ONLINE COMMERCIAL TOOLS

This subsection will introduce the online commercial tools. Because they work online, it is possible to use them on every platform. It is necessary to get a subscription or activation key. Otherwise, it is limited, where the features are only available for a short period or with limited options.

1) Dbdesigner.net

Since 2006, Dbdesigner.net (2018 version), is a database schema designer for data modeling (Fig. 2). The table representation is clean and has different colors for each table entry. It is possible to share the database design with other users to work simultaneously on it. This tool does not provide the visualization of the databases' CM model. Dbdesigner also offers a reverse engineering option. Besides it, forward engineering enables code export for DBMS like MySQL, Microsoft SQL Server, PostgreSQL, Oracle, and SQLite. This tool has distinct features that make it unique, such as a mini-map for fast navigation, keyboard shortcuts, instant save with history, copy and paste, undo and redo, and notes and comments. It supports data types such as binary, boolean, date, decimal, float, integer, and varchar. The database constraints supported are: PK, FK, NN, and UQ.

Pros: Developers can work simultaneously, has several database design templates, features as instant save, undo and redo, easy-to-use interface. Also, it enables reverse engineering.

Cons: No representation of the CM.

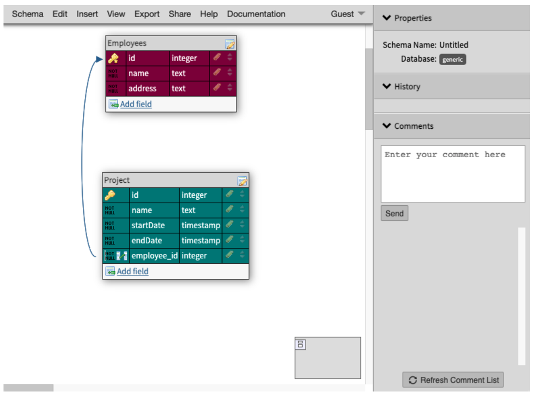


FIGURE 2. **Example of a database model at Dbdesigner.net.**

2) dbDiffo

Released in 2014, dbDiffo (version 3.2.72), is a database modeling tool similar to Onda. Before using the tool, it is necessary to specify the model name and choose the DBMS for script generation. With dbDiffo, it is impossible to do a real-time collaboration, and it does not provide the CM design. This tool does not allow reverse engineering, but regarding forward engineering, it is possible to export scripts for DBMS like IBM DB2, Microsoft SQL Server, MySQL, Oracle, and PostgreSQL. Also, the tool performs model checking. The tables' columns data types may be bigint, binary, bit, blob, char, date, datetime, decimal, double, float, integer, longblob, longtext, mediumint, mediumtext, numeric, smallint, text, time, timestamp, varchar, or year. Regarding the database constraints, it is possible to define PK, FK, and NN.

Pros: History toolbar, and unlimited undo.

Cons: No real-time collaboration and no reverse engineering.

3) GenMyModel

Released in 2012, GenMyModel (2013 version), speeds up the design of software architecture and business processes. It is easy to add new entities and create relationships between them because of its interface and the provided documentation. It is necessary to log into the application via GitHub or Google account, create a new diagram, and select a relational database. It is possible to create a database from scratch or select an existing project from the cloud. GenMyModel has real-time collaboration with a chat and also allows the creation of the CM, since it has its base in the UML. The supported DBMS are Apache Hive, Oracle, MySQL, and PostgreSQL. This tool supports different data types, such as boolean, binary, character, date, float, integer, time, or varchar. The tool can auto-generate PDF and MS Word documents based on custom templates and export diagrams to GitHub. The developers can use the open API and integration functions to build integrations for testing or code proofing. Regarding the constraints, it is possible to define PK, FK, NN, CH, and UQ.

Pros: Real-time team collaboration features, it auto-generates the documentation of the data models.

Cons: If using the free version, it only provides basic features and limits the design to 20 objects, including not only the tables, but also each column and relationship.

4) LucidChart

Released in 2008, Lucidchart (2018 version) is a powerful tool, it has a free version, and also offers a trial to explore its full potential. Lucidchart has team collaboration, it is also based in UML, and allows CM design. It is not possible to perform reverse engineering. Forward engineering is possible to several DBMS such as MySQL, PostgreSQL, Microsoft SQL Server, and Oracle. Each column may have any data type introduced by the user, later converted into the specific DBMS data types. As in other tools, it is possible to choose from the list of existing data types. The constraints are PK, FK, and NN.

Pros: Real-time collaboration feature, and different templates.

Cons: The user must know the specificities of the DBMS data types because it is possible to insert any string in the data type field.

5) sqlDBM

Released in 2017, sqlDBM (version 2.11.650.0), has a free version that comes with limited features, and it is possible to try the full version for 14 days. This design tool only requires configuring the database type, and it does not offer CM design. There is also a team collaboration tool. It is possible to perform reverse engineering, and forward engineering allows exporting the script to Microsoft SQL Server, MySQL, Snowflake, Amazon Redshift, PostgreSQL, and Azure Synapse Analytics. Each column data type can be bigint, bigint unsigned, binary, bit, blob, char, date, datetime, decimal, double, double unsigned, float, integer, numeric, text, time, timestamp, varchar, or year. Also, it is possible to specify the constraints PK, FK, NN, and UQ.

Pros: This tool has forward engineering and team collaboration possibilities. Minimal tutorial, in the beginning, explaining functionalities, no need to sign-up. "Undo" and "redo" options also present.

Cons: No CM design.

C. DESKTOP FREE TOOLS

In this subsection, we analyze four desktop free tools. These require installation, and as make part of the open-source community are free to use, and every developer can contribute to their development.

1) MySQL WORKBENCH - COMMUNITY VERSION

Created in 2002, MySQL Workbench (version 8.0.23) is an application used to manage and design a database schema. The open-source version has a GPL license with a GitHub repository. The significant differences between Community and Enterprise versions are the non-presence of schema and

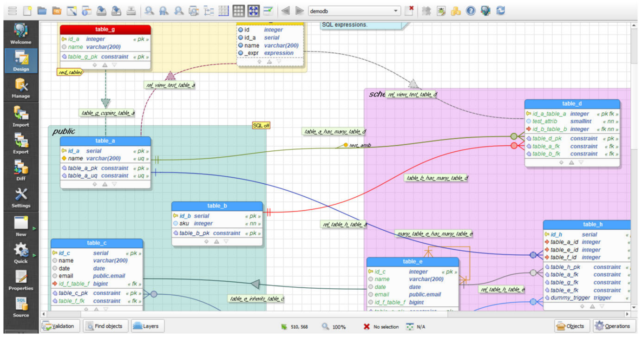


FIGURE 3. Example of a database model at pgModeler.

model validation, automated documentation of databases, and non-existence of firewall specification rules. Before working with the tool, it is necessary to set up the connection to the existing database. Otherwise, it is impossible to export the MySQL script resulting from the creation of tables and relationships. MySQL Workbench does not have a real-time collaboration feature, but has the capacity to design the CM. There is a possibility to reverse and forward engineer for MySQL Server database. Different categories organize the data types like numeric, characters, time, geometry, and others, such as bits or boolean. The number of constraints is also considerable, it enables an option to insert PK, FK, NN, UQ, Binary, and Unsigned (U).

Pros: Unlimited "Undo" and "Redo" options.

Cons: Only available for Windows and no real-time collaboration. Also, it is mandatory to set up the connection to the existing database.

2) pgModeler

Created in 2006, pgModeler (version 0.9.2) is a database modeling tool designed for PostgreSQL databases. Despite the need to pay for the compiled version, it is possible to get the open-source version and compile it manually. The tool has different colors to help visualization (Fig. 3). If there are missing functionalities, it is possible to create new extensions and contribute to open-source code development. The tool does not have an option for real-time collaboration, and it has the feature to design the CM. It also provides reverse and forward engineering for PostgreSQL databases. The tool performs model and database design checks. pgModeler has a database management module where it is possible to run SQL commands, explore the objects, and handle data. It supports distinct data types like bigint, bit, bool, char, date, decimal, float, int, JSON, money, text, time, and varchar. Also, different constraints such as PK, FK, UQ, Exclude (E), CH, and NN are present.

Pros: It is possible to collaborate on the tool.

Cons: Only supports PostgreSQL DBMS.

3) UMBRELLO UML

Released in 2006, Umbrello UML (version 2.32.0) is a UML diagram program developed by an international free software community. It is not possible to perform real-time collaboration. It has a feature to design conceptual data models. It is impossible to reverse engineering, but the forward engineering option allows generating SQL scripts for MySQL and PostgreSQL. Each table column can have different data types, such as boolean, char, double, float, int, or string. And it includes different database constraints: PK, FK, UQ, and NN.

Pros: Feature for CM.

Cons: Limited number of DBMS to export scripts, and no real-time collaboration.

D. DESKTOP COMMERCIAL TOOLS

In this subsection, we analyze desktop commercial tools. We chose these according to Google Trends, since the number of tools in this product type is high. For us to consider a tool, it had to be googled at least ten times per week, from 2004 until 2021, worldwide, and we found six desktop proprietary tools matching this criterion. To use either tool, first, it is necessary to install it on the machine and then either try a free trial or buy the commercial version.

1) dbSchema

Released in 2016, dbSchema (version 8.3.2) cannot provide a conceptual data model, neither exists real-time collaboration. It has both reverse and forward engineering that works with all relational DBMSs, including SqlServer, SAP Adaptive Server, Oracle, MySql, Ingres, Informix, Db2, Derby, Firebird, Frontbase, Cache, Pervasive, PostgreSQL, and SQLite. Also, the tool performs model checking. Like all the previous tools, it allows for each column to have one of the following data types: blob, boolean, char, date, double, float, int, json, real, text, and varchar. The constraints available are PK, FK, and NN.

Pros: The tool has both reverse and forward engineering for many DBMS.

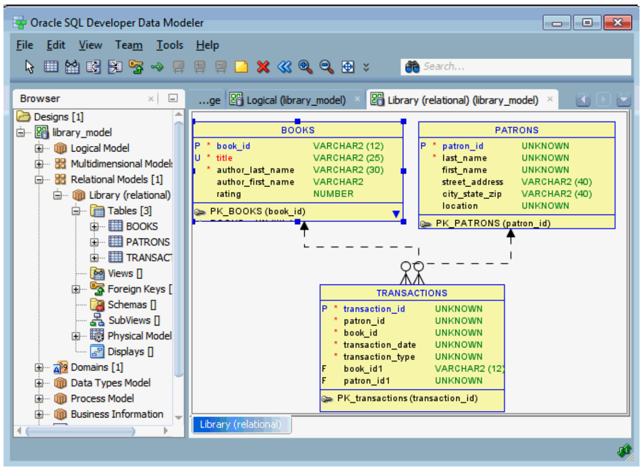
Cons: No real-time collaboration.

2) dbWrench

dbWrench (version 4.2.5) is a multi-platform database design and synchronization software, released in 2004. dbWrench supports the design of the CM, but has no real-time collaboration. It allows reverse engineering, and the forward engineering tool generates SQL scripts for Microsoft SQL Server, Oracle, PostgreSQL, and MySQL. It is possible to connect to a database and run the code for table creation. The tool has model checking and if subscribed database design checking. It offers some default column templates that save time creating tables. When adding a column, there is a possibility to specify data types like binary, blob, bit, boolean, char, date, decimal, double, float, JSON, number, real, time, and varchar. The constraints are PK, FK, and NN.

Pros: Reverse and forward engineering for several DBMS.

Cons: No real-time collaboration.



BOOKS			
P	*	book_id	VARCHAR2 (12)
U	*	title	VARCHAR2 (25)
	*	author_last_name	VARCHAR2 (30)
		author_first_name	VARCHAR2
		rating	NUMBER
PK_BOOKS (book_id)			

PATRONS			
P	*	patron_id	UNKNOWN
	*	last_name	UNKNOWN
		first_name	UNKNOWN
		street_address	VARCHAR2 (40)
		city_state_zip	VARCHAR2 (40)
		location	UNKNOWN
PK_PATRONS (patron_id)			

TRANSACTIONS			
P	*	transaction_id	UNKNOWN
	*	patron_id	UNKNOWN
	*	book_id	UNKNOWN
	*	transaction_date	UNKNOWN
	*	transaction_type	UNKNOWN
F		book_id1	VARCHAR2 (12)
F		patron_id1	UNKNOWN
PK_transactions (transaction_id)			

FIGURE 4. Example of a database model in Oracle SQL Developer Data Modeler.

3) ERWIN DATA MODELER

Founded in 1988, Erwin Data Modeler (2020 version) allows the creation and maintenance of databases and provides several tutorials to help understand how to do data modeling. Financial services, healthcare, critical infrastructure, and technology companies use Erwin Data Modeler. The tool does not have an option of real-time collaboration. Erwin provides a possibility to design the conceptual data model. It has reverse and forward engineering that supports DBMS such as Oracle, MySQL, IBM DB2, SAP IQ, and Teradata. Also, the tool provides model checking and in the paid version database design checking. The different data types for columns in this tool are, amongst others, binary, byte, interval, datetime, image, audio, varchar, char, integer, date, boolean, real, and float. The tool has constraints such as PK, FK, NN, and UQ.

Pros: Supports several DBMS for forward engineering.

Cons: It is necessary to fill in a form to try the trial version, but there is no guarantee that the application will be accepted. The local version of the tool doesn't work on macOS, and it is necessary to use the cloud version. No real-time collaboration.

4) NAVICAT

Navicat (version 15) is proprietary software created in 2002 and provides a mini-map for fast navigation. It allows adding colors to tables, thus making them more visually appealing. This tool has real-time collaboration. It is a powerful and cost-effective database design tool that allows designing CM. It allows performing reverse and forward engineering processes. This tool allows creating data models for MySQL, Microsoft SQL Server, Oracle, PostgreSQL, SQLite, and MariaDB DBMS. The tool also performs model checking. It has several data types, such as blob, boolean, integer, varchar, date, and timestamp. For the data constraints, there are PK, FK, NN, and UQ.

Pros: Performs reverse and forward engineering.

Cons: No real-time collaboration.

TABLE 2. Comparison of different tools.

Product Type	Tool name	Characteristics							
		Open source	Online	Supported DBMS	Supported OS	Constraints	CM design	Free	Licence
I	Dbdesigner.id	✓	✓	MySQL	Windows, Linux, macOS	PK, FK, UQ, NN	✗	✓	MIT
	Onda	✓	✓	PostgreSQL, MySQL, Oracle, MariaDB, and SQLite	Windows, Linux, macOS	PK, FK, NN, CH, UQ	✓	✓	CCANCSAIL*
	WWW SQL Designer	✓	✓	MySQL, SQLite, Oracle, PostgreSQL, MySQL, web2py	Windows, Linux, macOS	PK, FK, UQ, NN	✗	✓	BSD-3-Clause
II	Dbdesigner.net	✗	✓	MySQL, Microsoft SQL, PostgreSQL, Oracle, SQLite	Windows, Linux, macOS	PK, FK, UQ, NN	✗	✗	Proprietary
	dbDiffo	✗	✓	IBM DB2, MS SQL Server, MySQL, Oracle, PostgreSQL	Windows, Linux, macOS	PK, FK, NN	✓	✗	Proprietary
	GenMyModel	✗	✓	Apache Hive, Oracle, MySQL, PostgreSQL	Windows, Linux, macOS	PK, FK	✓	✗	Proprietary
	Lucidchart	✗	✓	MS SQL Server, MySQL, Oracle	Windows, Linux, macOS	PK, FK, NN	✓	✗	Proprietary
	sqlDBM	✗	✓	MS SQL Server, MySQL, Snowflake, Amazon Redshift, PostgreSQL, Azure Synapse Analytics	Windows, Linux, macOS	PK, FK, NN	✗	✗	Proprietary
III	MySQLWorkbench Community Version	✓	✗	MySQL	Windows	PK, FK, NN, UQ, Binary, Unsigned	✓	✓	GPL
	pgModeler	✓	✗	PostgreSQL	Windows, Linux, macOS	PK, FK, UQ, E, CH	✓	✓	GPL
	Umbrello UML	✓	✗	MySQL, PostgreSQL	Windows, macOS	PK, FK, UQ	✓	✓	LGPL
IV	dbSchema	✗	✗	SqlServer, SAP Adaptive Server, Oracle, MySQL, Ingres, Informix, Db2, Derby, Firebird, Frontbase, Cache, Pervasive, PostgreSQL, SQLite	Windows, Linux, macOS	PK, FK, NN	✗	✗	Proprietary
	dbWrench	✗	✗	Microsoft SQL Server, Oracle, PostgreSQL, MySQL	Windows, Linux, macOS	PK, FK, NN	✓	✗	Proprietary
	Erwin Data Modeler	✗	✗	Oracle, MySQL, IBM, DB2, SAP IQ, Teradata	Windows, Linux, macOS	PK, FK, NN	✓	✗	Proprietary
	Navicat	✗	✗	MySQL, Microsoft SQL Server, Oracle, PostgreSQL, SQLite, MariaDB	Windows, Linux, macOS	PK, FK, UQ, NN	✓	✗	Proprietary/ Shareware
	Oracle SQL Developer Data Modeler	✗	✗	Oracle, DB2, MySQL, and Microsoft SQL Server MySQL	Windows, Linux, macOS	PK, FK, UQ, NN	✓	✗	Proprietary
	PowerDesigner	✗	✗	Oracle, PostgreSQL, IBM DB2, SAP, Sybase, Microsoft SQL, Teradata, Hadoop Hive, MySQL	Windows	PK, FK, UQ, NN	✓	✗	Proprietary

Constraints meaning: PK - Primary Key; FK - Foreign Key; UQ - Unique; NN - Not Null; E - Exclude; CH - Check; U - Unsigned
*Creative Commons Attribution-NonCommercial-ShareAlike 4.0 International License

5) ORACLE SQL DEVELOPER DATA MODELER

Oracle SQL Developer Data Modeler (version 20.2 tested) was released in 2006, and it is an integrated development environment that simplifies the development and management of Oracle databases (Fig. 4). This application permits conceptual data modeling. Although, it does not have real-time collaboration. The application allows performing reverse and forward engineering for the Oracle DBMS. It supports managing the Oracle Database performance, security, storage, and settings. Also, the tool performs model checking and in the subscribed version database design checking. The tool has different data types such as blob, char, decimal, float, date, and timestamp. There is the possibility to define PK, FK, and UQ constraints.

Pros: Quick loading time, and the existence of a tutorial that explains how the tool works.

Cons: Only supports Oracle database.

6) PowerDesigner

PowerDesigner (version 16.7) is a collaborative enterprise modeling tool, released in 1989 with the name of "AMC*Designer" and is currently owned by SAP. It is a modeling tool for easy visualization, understanding, and management of the data in a project. PowerDesigner has real-time collaboration, and it is possible to implement a CM. Also, it has multiple database connections to model the data. The tool also offers reverse engineering. Moreover, forward engineering allows working with the most popular DBMS, such as Oracle, PostgreSQL, IBM DB2, Microsoft SQL Server, SAP, Sybase, Hadoop Hive, MySQL, and Teradata. This tool has model checking and in the paid version database design checking. This tool only works on Windows operating system. PowerDesigner allows creating multiple entities at once, which saves a lot of time. The data types are integer, decimal, money, boolean, characters, text, date, and timestamp. For the data constraints, there are PK, FK, NN, and UQ.

Pros: Real-time collaboration, and several DBMS for forward engineering.

Cons: Only available for Windows.

In this section, we did a qualitative evaluation of each tool by presenting concise descriptions and pros and cons. Table 2 displays a summary of the fundamental characteristics of each tool. In the following section, we will explain the evaluation method and the scoring assessment with the objective of producing an objective analysis.

V. THE EVALUATION METHODOLOGY

Since we focus on evaluating data modeling tools for relational databases, this section will explain how to perform this assessment.

We propose a hybrid evaluation methodology that will suppress the previously mentioned shortcomings, such as bias and lack of details in the BRR score, and penalty for high scores in less critical measures in the OSSpal methodology. We developed an approach based on the BRR calculations, but instead of the 12 categories, we used the seven defined in the OSSpal methodology, which encompass all significant categories by combining some of the original 12 into a single category. Also, we introduced some changes in the evaluation of the Functionality features of BRR. First, we removed the binary answers by providing a broader range of values. Second, we discarded adding extra features, no bonus score for tools, and only scored a tool with -1 when a feature is absent from the proposed set. These changes provide better assessment of the characteristics and features of the tools. They widen the scope of values to clarify the differences amongst tools and eliminate subjectivity issues by not adding features that are not crucial to a fair assessment.

The OSSpal methodology combines quantitative and qualitative measures to evaluate and compare software tools in several categories. The examiner assigns a quantitative score to the tools instead of only analyzing the pros and cons. OSSpal proposes the following seven categories:

- **Functionality**: analyses how well the software meets the user's requirements.
- **Operational Software Characteristics**: evaluates how secure the software is, how well does it perform, how good is the User Interface (UI), and how easy is the software to install, configure, deploy, and maintain.
- **Support and Service**: examines how well is the software component supported and if there is commercial or community support or both.
- **Documentation**: assesses if there is a suitable tutorial and reference documentation for the software.
- **Software Technology Attributes**: analyses how good the software architecture is and how portable, extensible, open, and easy to integrate it is.
- **Community and Adoption**: examines the adoption of the component by community, market, and industry. Also, how active is the community for the software.
- **Development Process**: evaluates the level of professionalism of the development process and the project organization.

After assessing the previous OSSpal categories, four steps were followed according to the BRR model:

1) Identify all the software components to be analyzed and measure them considering the evaluation criteria.
2) Select an appropriate weighting factor for each category/metric (sub-categories):
   a) Assign a percentage of importance for each category that will be used as a weighting factor. The sum of all the weigh factors should add up to 100%.
   b) Similarly, if a specific category is evaluated using multiple metrics, assign a percentage of importance for each metric, totaling 100% within the category.
3) Score each category/metric and assign it a value from 1 to 5 (1 - Unacceptable, 2 - Poor, 3 - Acceptable, 4 - Very Good, 5 - Excellent).
4) The category evaluation and the weighting factors (Eq. 1) should be used to calculate the final score (Eq. 2).

$$\text{Category Score} = \frac{\sum \textit{Metric score} \times \textit{Metric weight}}{\sum \textit{Metric weights}} \tag{1}$$

$$\text{Final Score} = \frac{\sum \textit{Category score} \times \textit{Category weight}}{\sum \textit{Category weight}} \tag{2}$$

As defined in the BRR model [8], the Functionality category will have a different approach because *"each type of software application has a unique set of features that needs to be fulfilled by the software package"*. The original model evaluates the presence of features, not being a typical qualitative or quantitative measurement. However, in our hybrid approach, we introduce a quantitative measure of these features to highlight the differences between the tools. The method to assess this category ends with Equation 3, and is as follows:

- Specify the features to analyze, weighting them from 1 to 3 (less important to very important);
- Compare the feature list of the component being evaluated with the standard feature list. For each feature:
  - If met, classify the implementation of the feature using a scale from 1 to 3 (poor implementation to full implementation) and multiply by the feature weight;
  - If not met, subtract importance weight from the sum (classify as -1).
- Standardize the result to a scale from 1 to 5 (Table 3):
  - The result is the cumulative sum of all the feature results, which punishes the missing features.

$$\text{Functionality Score} = \frac{\sum \textit{Feature score} \times \textit{Feature weight}}{\sum \textit{Feature weights}} \tag{3}$$

Equation 3 scores each of the functionality items based on a weighted average. The aim of this evaluation is to provide a fair evaluation based on known relevant characteristics of the tools in analysis. Although it does not consider specific business contexts, if needed, context can influence a feature's weight.

TABLE 3. Quality ranking.

Values	Score	Evaluation
>96%	5	Excellent
[90% - 96% [	4	Good
[80% - 90% [	3	Acceptable
[65% - 80% [	2	Poor
<65%	1	Unacceptable

VI. MODELING TOOLS EVALUATION

In this section, we present the application of the methodology proposed in the previous section to evaluate the data modeling tools.

To define the weights to be given to OSSPal categories, we carried out a survey. In this survey, we consulted 13 professionals in the area of databases and software engineering. First, we asked them to give a score (in percentage) of the seven OSSpal categories summing 100%, according to their experience in database modeling and general software engineering. Then, we asked them to rank, with values ranging from one to three, the chosen features of the Functionality category. We could detail other significant features, such as script generation, supported notations, secure connection to the database, or transformation rules. However, to reduce the number of parameters to consider, they were merged in broader categories. This approach uses fewer characteristics and was necessary to keep the time to answer the survey under 5 minutes to avoid people from abandoning the survey, something that is reported to occur at about 7 minutes into answering.[1] Tables 4 and 5 display the results of our survey. Moreover, the survey was individual. Each professional provided their answers, and there was no group decision-making activity to conclude the percentage of each category or the weight each feature should have. This adaptability is also an advantage of our approach. It allows readjusting the values according to the goal of any user using it.

Regarding the *Category* (Table 4), we asked to fill each Category with a percentage totaling 100%. We analyzed the average, standard deviation (STDDEV), and median. The maximum and minimum values of each Category were disparate. The lower value was a 15% difference between the values in the *Support and Service* and *Development Process* categories, up to 40% in the *Functionality* category. Thus, we evaluated the average without the maximum and minimum values and presented the difference between these values. The only positive difference are in the *Operational Software Characteristics* (0.38) and *Software Technology Attributes* (0.02) categories. We considered the average values (AVG column) because of three reasons: i) the sum of the average values is 100%; ii) the sum of average without maximum and minimum values was only 98.09%, and the differences are not significant between the two approaches; iii) the sum of median values was only 97.50%, and the differences are not significant between average and median.

[1]https://www.surveymonkey.com/curiosity/survey_completion_times/

The *Functionality* Category had the highest weight, and we further decomposed it into features or sub-categories. We asked the same community to rank and weight each measure, that we previously found crucial to the tool assessment, with a value between 1 (less important) and 3 (very important). We also analyzed the average, standard deviation (STDDEV), and median. The maximum and minimum values of each Functionality were equivalent amongst the functionalities. Because the variation of the values is minor, the differences between averages (with and without maximum and minimum values) are not significant (0.01), we used the average values (AVG column). Despite this selection of weights, a different context may require different values. In the proposed methodology, this adaptation is possible by assigning a different set of weights without altering the evaluation values of each criterion.

The primary category is *Functionality* as it encompasses, among others, the number of supported DBMS, the restrictions it has, and constraints. Therefore, this category got a weight of **37.69%**. In the second place, *Operational Software Characteristics* has **19.62%** as well and includes areas such as security, performance, usability, reliability, and scalability, which are crucial to evaluate each tool. The *Documentation* category comes in third with **10.46%** once good information helps with installation, configuration, and extension of the software easily. *Support and Service*, *Community and Adoption*, and *Software Technology Attributes* had **9.73%**, **9.38%**, and **8.62%** respectively because the software needs to be supported, modular, and easy to extend. The latter category also measures if the project is extensible and how fast problem resolution is. With less importance, *Development Process* categories received a weight of **4.50%**. Table 4 represents these weights, ordered from most to less important, based on the average (in bold).

The next step was to decompose the Functionality category into the most relevant characteristics (features or sub-categories). Table 5 displays the results of our survey to the research community survey, previously explained, the average (in bold) ordered these, leading to the following layout: Reverse engineering (2.62), Model checking (2.46), Supported DBMS (2.38), Supported constraints (2.31), Database design checking (2.31), CM design (2.08), Supported OS (1.92), and Real-time collaboration (1.69). The sum of the weights was **17.77**, this value will be used in Equation 1. To model large and complex systems, we consider CM an important feature for these tools to have. However, the surveyed professionals scored it as one of the less important. This shows that the surveyed professionals are more acquainted with small to medium projects that do not fully benefit from conceptual modelling as large and more complex projects do. Moreover, our approach allows for any user to only change the features weight to have a completely new perspective on the analyzed tools.

TABLE 4. OSSpal Category weights (%).

Category	AVG	STDDEV	MEDIAN	MAX	MIN	AVG (w/o MAX-MIN)	AVG (w/o MAX-MIN)-AVG
Functionality	**37.69**	12.50	35.00	60	20	37.27	-0.42
Operational Software Characteristics	**19.62**	6.64	20.00	30	5	20.00	0.38
Documentation	**10.46**	5.37	10.00	20	3	10.27	-0.19
Support and Service	**9.73**	5.40	8.00	20	5	9.23	-0.50
Community and Adoption	**9.38**	7.18	7.00	30	2.5	8.14	-1.25
Software Technology Attributes	**8.62**	4.29	10.00	15	2	8.64	0.02
Development Process	**4.50**	3.79	5.00	15	0	3.95	-0.55
Sum	**100**	-	97.50	-	-	98.09	-2.5

TABLE 5. Functionality Features Weights (1 to 3).

Measures	AVG	STDDEV	MEDIAN	MAX	MIN
Reverse engineering	**2.62**	0.49	3.00	3.00	2.00
Model checking	**2.46**	0.75	3.00	3.00	1.00
Supported DBMS	**2.38**	0.74	3.00	3.00	1.00
Supported constraints	**2.31**	0.61	2.00	3.00	1.00
Database design checking	**2.31**	0.72	2.00	3.00	1.00
CM design	**2.08**	0.83	2.00	3.00	1.00
Supported OS	**1.92**	0.83	2.00	3.00	1.00
Real-time collaboration	**1.69**	0.91	1.00	3.00	1.00
Sum	**17.77**	-	-	-	-

TABLE 6. Measures of the Functionality Category for each tool.

Product Type	Tool name	Measures and weights								Weighted total	W.Total / Sum Weights (Eq. 3)	Percentage (%)	Normalization Tab 3
		Rev. eng.	Model check	Supp. DBs	Supp. constr.	DB design checking	CM design	Supp. OS	Real-time collab.				
		2.62	2.46	2.38	2.31	2.31	2.08	1.92	1.69				
I	Dbdesigner.id	3	-1	1	3	-1	-1	3	2	19.46	1.10	36.51	1
	Onda	-1	-1	3	3	-1	3	3	-1	17.00	0.96	31.89	1
	WWW SQL Designer	-1	-1	1	3	-1	-1	3	-1	3.92	0.22	7.36	1
II	Dbdesigner.net	3	-1	3	3	-1	-1	3	3	25.92	1.46	48.63	1
	dbDiffo	-1	3	3	2	-1	2	3	-1	22.46	1.26	42.14	1
	GenMyModel	-1	-1	3	3	-1	2	3	1	18.31	1.03	34.34	1
	Lucidchart	-1	-1	3	2	-1	2	3	1	16.00	0.90	30.01	1
	sqlDBM	3	-1	3	3	-1	-1	3	1	22.54	1.27	42.28	1
III	MySQL Workbench	3	-1	1	3	-1	3	1	-1	18.85	1.06	35.35	1
	pgModeler	3	3	1	3	3	3	3	-1	41.77	2.35	78.35	2
	Umbrello UML	-1	-1	2	3	-1	2	2	-1	10.62	0.60	19.91	1
IV	dbSchema	3	3	3	2	-1	-1	3	-1	26.69	1.50	50.07	1
	dbWrench	3	3	3	2	1	3	3	-1	39.62	2.23	74.31	2
	Erwin Data Modeler	3	3	3	3	1	2	3	-1	39.85	2.24	74.75	2
	Navicat	3	3	3	3	-1	3	3	2	42.38	2.39	79.51	2
	Oracle SQL Developer Data Modeler	3	3	3	3	1	3	3	-1	41.92	2.36	78.64	2
	PowerDesigner	3	3	3	3	1	3	1	1	41.46	2.33	77.78	2

Table 6 displays the result of the assessment of each Feature of the Functionality Category, according to the methodology criteria. Between brackets are the average weight values that resulted from our survey to professionals. As previously mentioned, any software tool that is not equipped with one of the measures under analyses will have its weight (importance) subtracted in the calculation. Also, to better differentiate the tools, we added a wider range for the values depending on each measure:

1) Supported DBMS: this category accounts for the number of DBMS that the tool supports to generate SQL scripts from the physical data model. This characteristic has the highest weight value because it is important for a tool to have a wide range of choices, not constraining the user. The tools that support one database got the score of 1, the tools that support two, a score of 2, and three or more DBMS receive a score of 3;
2) The number of different constraints that the tool has is defined in the feature: Supported constraints. If it has at least two, such as PK and NN, the tool receives a score of 2, and those that have these or more, such as UQ or E have a score of 3;
3) In the feature Supported OS, we considered three OS: Windows, Linux, and macOS. Each tool will score one point for each supported OS. Thus, a tool

TABLE 7. Assessment of the OSSpal Categories' score for each tool.

Product Type	Tool name	Categories						
		Functionality (37.69)	Operational Software Characteristics (19.62)	Documentation (10.46)	Support and Service (9.73)	Community and Adoption (9.38)	Software Technology Attributes (8.62)	Development Process (4.50)
I	Dbdesigner.id	1	5	2	2	3	4	3
	Onda	1	4	1	3	2	4	4
	WWW SQL Designer	1	3	4	5	3	4	5
II	Dbdesigner.net	1	5	5	2	3	3	3
	dbDiffo	1	3	4	3	2	3	3
	GenMyModel	1	3	3	2	3	3	3
	Lucidchart	1	4	3	4	4	3	5
	sqlDBM	1	5	4	2	3	3	5
III	MySQL Workbench	1	4	2	2	3	4	3
	pgModeler	2	4	5	5	4	4	4
	Umbrello UML	1	3	5	4	5	4	5
IV	dbSchema	1	4	5	3	3	3	5
	dbWrench	2	3	3	3	3	3	3
	Erwin Data Modeler	2	4	4	2	3	3	3
	Navicat	2	3	4	2	3	3	3
	Oracle SQL Developer Data Modeler	2	5	3	2	3	3	3
	PowerDesigner	2	3	4	2	3	3	5

like PowerDesigner that only supports Windows will score 1, Umbrello UML supports Windows and macOS will score 2, and the rest 3.

4) Only some tools have reverse engineering, and we set the values according to the limitations. If the tool is limited to 5 entities the score will be 1, until 10 entities the tool scores 2, and if there are no limitations scores 3. Otherwise, the set importance will be subtracted from the cumulative sum, so the score will be set as -1;
5) Concerning the feature Real-time collaboration, we set the values regarding the number of allowed users or the need to pay. If a tool has this feature only in the paid version scores 1, if it is free but is limited to a certain number of users scores 2, and if there is no limitation to users, and it's free scores 3. Otherwise, the set importance will also be subtracted from the cumulative sum, so the score will be set as -1;
6) The evaluation of the feature CM design is according to the tools' capacity to provide a CM design. If a tool has a limitation in the number of entities and relationships scores 1, if the limitation is the number of entities or relationships scores 2, no limitations score 3. Otherwise, the set importance will be subtracted from the cumulative sum, so the score will be set as -1.
7) Model-checking: represents any alert message about errors present in the model, such as *entities errors* (names or redundant entities), *relationship errors* (names, degree, cardinality, existence, type, or redundant relationships), *completeness* (ensure no entities have been omitted), and *schema* (check attribute names (e.g. TableName_AttribName)). If a tool has this feature only in the paid version scores 1, if it is free but is limited to a certain number of errors scores 2, and if there is no limitation to alerts and it's free scores 3. Otherwise, the set importance will also be subtracted from the cumulative sum, so the score will be set as -1;
8) Database design checking: feature to test the schema, tables, triggers, etc., and data integrity and consistency of the database. If a tool has this feature only in the paid version scores 1, if it is free but limited it should score 2, but if there is no limitation it scores 3. Otherwise, the set importance will also be subtracted from the cumulative sum, so the score will be set as -1.

Following the proposed methodology, we calculated the weighted total, representing the cumulative multiplication of the score by the feature's weight (importance). For example, Dbdesigner.id Functionality score (Equation 3) is calculated as: $[(3\times 2.62)+(-1\times 2.46)+(1\times 2.38)+(3\times 2.31)+(-1\times 2.31)+(-1\times 2.08)+(3\times 1.92)+(2\times 1.69)]/17.77 = 1.10$.

Using this value, we assessed the percentage of each tool score. Finally, we converted the values according to Table 3, represented in the last column. None of the tools achieved the highest score (5). pgModeler, dbWrench, Erwin Data Modeler, Navicat, Oracle SQL Developer Data Modeler, and PowerDesigner all with a score of 2, were the best-rated tools in this Category.

The last step is to calculate each tool's score, using Equation 2, taking into account the values of different categories from the Table 7, multiplying it by its weights, where we converted the percentage into unit values, and adding up these scores. The result of these calculations is shown in Table 8.

After applying the proposed methodology to the tools (Eq. 2), in the product type of Online free tools, **WWW SQL Designer** received the highest score, **2.71**. This tool has some shortcomings, such as supporting only one database, not allowing real-time collaboration, and not having reverse

TABLE 8. Final score.

Product Type	Tool name	Score
I	WWW SQL Designer	**2.72**
	Dbdesigner.id	2.52
	Onda	2.27
II	Dbdesigner.net	**2.75**
	sqlDBM	2.74
	Lucidchart	2.72
	dbDiffo	2.26
	GenMyModel	2.15
III	pgModeler	**3.45**
	Umbrello UML	2.92
	MySQL Workbench	2.33
IV	Oracle SQL Developer Data Modeler	**2.92**
	Erwin Data Modeler	2.83
	dbSchema	2.74
	PowerDesigner	2.72
	Navicat	2.63
	dbWrench	2.62

engineering. These contributed to a low value in the Functionality category, the one that has the highest weight. However, in the remaining categories, the tool performs well, giving it a prominent place.

In the product type of Online commercial tools, **Dbdesigner.net** ended the assessment with **2.75** . This tool has a high number of supported DBMS, restrictions, and operating systems. It supports reverse engineering and real-time collaboration. Although it is not extensible, the tool is easy to work, with good documentation, qualified support, and broadly adopted by the community.

In the product type of Desktop free software, **pgModeler** received the highest score, **3.45** out of 5, the overall highest value. Despite only supporting PostgreSQL, what translated to a low score in an important feature, and not having real-time collaboration, this tool achieved a good Functionality score. In addition, pgModeler is extensible, has clear documentation, and is a globally used software by the community.

In the product type of Desktop commercial software, **Oracle SQL Developer Data Modeler** got the highest score, **2.92**. This tool supports several DBMS, has real-time collaboration, model checking and database design checking (in the paid version), allows performing reverse engineering, and it has good documentation.

All the previously mentioned and analyzed tools offer unique resources to its users, and the differences in the final score display their heterogeneity. This classification results mainly from the fundamental differences in the Functionality category. From the eight features in our approach, all tools have three (with the highest importance according to our survey). However, the BRR model imposes a penalization when a tool has a missing feature.

In addition, it allows us to generate results that accurately reflect the usefulness of the tools and their level of productivity for users, based on the chosen functionality characteristics. Although we have reached this result with certain weights, the advantage of this methodology is that it allows changing the weights to more context-specific values. If someone wants a more contextualized analysis of a unique area, they must change these values. Changing weights allows adapting the method to a more specific view without altering the tools' assessment values.

Ultimately, the choice must also encompass the purpose of the project and the developer's skills. Nevertheless, we only considered eight features. Despite our survey to classify their importance, we recognize that, for other evaluators, these specific features may not be the best, the most adequate, or the most representative for their specific needs of data modeling tools. We tried to remove that subjectivity by gathering input from several database and software engineering professionals. However, the methodology enables the user to easily adapt the weights as they see fit to suit their specific context, thus achieving more adequate results for their requirements and context.

VII. CONCLUSION

This paper covers the evaluation of six open-source and eleven proprietary database modeling tools using a new and tailored approach. We gathered information for this analysis from the documentation available on each tool's website, by installing, testing, and using the tools. Also, we selected the proprietary desktop software, taking into account their popularity in Google Trends. Notwithstanding, we recognize that the context in which the tools will be used may represent an important factor for choice.

We used a hybrid methodology based on BRR and OSSpal and made a survey with professional researchers to assess category and functionalities weights. WWW SQL Designer (2.72), Dbdesigner.net (2.75), pgModeler (3.45), and Oracle SQL Developer Data Modeler (2.92) are the most valuable tools in Online free, Online commercial, Desktop free, and Desktop commercial, respectively. Overall, pgModeler, an open-source and free desktop tool, achieved the highest value amongst the 17 evaluated tools according to the criteria chosen.

To provide a meaningful evaluation of database modelling tools, it was necessary to develop a hybrid methodology, since neither BRR nor OSSPal were able to fully satisfy the requirements of this analysis per se. BRR is prone to ignore important information, since it considers subjectively the seven highest ranked categories out of the 12 possible. And the OSSpal methodology does not provide a referential final score. In addition, by combining both methodologies and the further improvement to the functional evaluation, we have shown that our approach provides more meaningful and tailored results for the evaluators, that can be also adapted to support decision in specific contexts, just by changing the weights. Changing weights allows adapting our approach to a more specific view without altering the evaluation values of each criterion.

As future work, we intend to develop a framework that can easily implement the evaluation methodology.

REFERENCES

[1] S. B. Navathe, "Evolution of data modeling for databases," *Commun. ACM*, vol. 35, no. 9, pp. 112–123, Sep. 1992.

[2] *IBM Definition of Data Modeling*. Accessed: Apr. 27, 2021. [Online]. Available: https://www.ibm.com/cloud/learn/data-modeling

[3] G. Simsion and G. Witt, *Data Modeling Essentials*. Amsterdam, The Netherlands: Elsevier, 2004.

[4] *Wikipedia Definition of Data Modeling*. Accessed: Apr. 27, 2021. [Online]. Available: https://www.ibm.com/cloud/learn/data-modeling

[5] R. Kimball and M. Ross, *The DataWarehouse Toolkit*. Hoboken, NJ, USA: Wiley, 2004.

[6] A. Silberschatz, H. F. Korth, and S. Sudarshan, *Database System Concepts*, 7th ed. New York, NY, USA: McGraw-Hill, 1991.

[7] T. L. Saaty, "What is the analytic hierarchy process?" in *Mathematical Models for Decision Support*, G. Mitra, H. J. Greenberg, F. A. Lootsma, M. J. Rijkaert, and H. J. Zimmermann, Eds. Berlin, Germany: Springer, 1988, pp. 109–121.

[8] OpenBRR: SpikeSource and Intel Corporation, "Business readiness rating for open source," USA, Tech. Rep. BRR_whitepaper_2005RFC1, 2005.

[9] A. I. Wasserman, X. Guo, B. McMillian, K. Qian, M.-Y. Wei, and Q. Xu, "OSSpal: Finding and evaluating open source software," in *Proc. IFIP Int. Conf. Open Source Syst.*, 2017, pp. 193–203.

[10] *International Data Corporation Software Taxonomy*, Int. Data Corp., Needham, MA, USA, 2016.

[11] T. Ferreira, I. Pedrosa, and J. Bernardino, "Evaluating open source business intelligence tools using OSSpal methodology," in *Proc. 9th Int. Joint Conf. Knowl. Discovery, Knowl. Eng. Knowl. Manage.*, A. L. N. Fred and J. Filipe, Eds., 2017, pp. 283–288.

[12] N. Leite, I. Pedrosa, and J. Bernardino, "Open source business intelligence platforms' assessment using OSSpal methodology," in *Proc. 15th Int. Joint Conf. e-Bus. Telecommun. (ICETE)*, Porto, Portugal, vol. 1, C. Callegari, M. van Sinderen, P. Novais, P. G. Sarigiannidis, S. Battiato, Á. S. S. de León, P. Lorenz, and M. S. Obaidat, Eds., Jul. 2018, pp. 356–362.

[13] A. K. Pereira, A. P. Sousa, J. R. Santos, and J. Bernardino, "Open source data mining tools evaluation using OSSpal methodology," in *Proc. 13th Int. Conf. Softw. Technol. (ICSOFT)*, Porto, Portugal, L. A. Maciaszek and M. van Sinderen, Eds., Jul. 2018, pp. 706–712.

[14] T. Ferreira, I. Pedrosa, and J. Bernardino, "Evaluating open source E-commerce tools using OSSpal methodology," in *Proc. 20th Int. Conf. Enterprise Inf. Syst. (ICEIS)*, Funchal, Portugal, vol. 1, S. Hammoudi, M. Smialek, O. Camp, and J. Filipe, Eds., Mar. 2018, pp. 213–220.

[15] H. C. de Paula and J. Bernardino, "An application of OSSpal for the assessment of open source project management tools," in *Proc. 15th Int. Conf. Web Inf. Syst. Technol. (WEBIST)*, Vienna, Austria, A. Bozzon, F. J. D. Mayo, and J. Filipe, Eds., Sep. 2019, pp. 411–417.

[16] A. Oliveira and J. Bernardino, "Evaluating open source project management tools using OSSPal methodology," in *Proc. 15th Int. Conf. Web Inf. Syst. Technol. (WEBIST)*, Vienna, Austria, A. Bozzon, F. J. D. Mayo, and J. Filipe, Eds., Sep. 2019, pp. 343–350.

[17] J. Marques and J. Bernardino, "Evaluation of Asana, Odoo, and ProjectLibre project management tools using the OSSpal methodology," in *Proc. 11th Int. Joint Conf. Knowl. Discovery, Knowl. Eng. Knowl. Manage. (ICK)*, Vienna, Austria, vol. 2, J. L. G. Dietz, D. Aveiro, and J. Filipe, Eds., Sep. 2019, pp. 397–403.

[18] S. Cruz and J. Bernardino, "Project management tools assessment with OSSpal," in *Proc. 11th Int. Joint Conf. Knowl. Discovery, Knowl. Eng. Knowl. Manage. (ICK)*, Vienna, Austria, vol. 2, J. L. G. Dietz, D. Aveiro, and J. Filipe, Eds., Sep. 2019, pp. 390–396.

[19] A. Calçada and J. Bernardino, "Evaluation of Couchbase, CouchDB and MongoDB using OSSpal," in *Proc. 11th Int. Joint Conf. Knowl. Discovery, Knowl. Eng. Knowl. Manage. (ICK)*, Vienna, Austria, vol. 1, A. L. N. Fred and J. Filipe, Eds., Sep. 2019, pp. 427–433.

[20] R. Karthikeyan, K. Venkatesan, and A. Chandrasekar, "A comparison of strengths and weaknesses for analytical hierarchy process," *J. Chem. Pharmaceutical Sci.*, vol. 9, no. 3, pp. 12–15, 2016.

[21] D. Batra and G. M. Marakas, "Conceptual data modelling in theory and practice," *Eur. J. Inf. Syst.*, vol. 4, no. 3, pp. 185–193, Aug. 1995.

[22] D. L. Moody and G. G. Shanks, "What makes a good data model? Evaluating the quality of entity relationship models," in *Proc. Int. Conf. Conceptual Modeling*, in Lecture Notes in Computer Science: Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics, vol. 881, 1994, pp. 94–111.

[23] D. Brdjanin, G. Banjac, D. Banjac, and S. Maric, "An experiment in model-driven conceptual database design," *Softw. Syst. Model.*, vol. 18, no. 3, pp. 1859–1883, Jun. 2019.

[24] B. Thalheim, *Entity-Relationship Modeling: Foundations of Database Technology*. New York, NY, USA: Springer-Verlag, 2000.

[25] P. P.-S. Chen, "The entity-relationship model—Toward a unified view of data," *ACM Trans. Database Syst.*, vol. 1, no. 1, pp. 9–36, 1976.

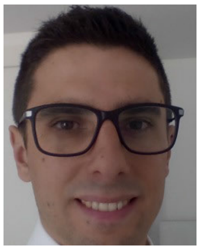
[26] "OMG unified modeling language (OMG UML)," USA, Tech. Rep. formal/2017-12-05, Dec. 2017.

[27] S. Al-Fedaghi and H. Alahmad, "Orientation in conceptual modeling frameworks," in *Proc. IEEE 15th Int. Conf. Dependable, Auton. Secure Comput., 15th Int. Conf. Pervasive Intell. Comput., 3rd Int. Conf. Big Data Intell. Comput. Cyber Sci. Technol. Congr. (DASC/PiCom/DataCom/CyberSciTech)*, Nov. 2017, pp. 1298–1303.

[28] N. E. A. Watt, *Database Design*, 2nd ed. BC, Canada: BCcampus, 2014.

[29] *79 Data Modeling Tools Compared*. Accessed: Oct. 16, 2020. [Online]. Available: https://www.databasestar.com/data-modeling-tools/

[30] *20 Best Data Modeling Tools*. Accessed: Oct. 16, 2020. [Online]. Available: https://www.guru99.com/data-modeling-tools-design-database.html



GONÇALO CARVALHO received the B.Sc. degree in geography from the University of Coimbra (UC), in 2005, the M.Sc. degree in geographical information systems from the University of Trás-os-Montes e Alto Douro (UTAD), in 2009, and the B.Sc. degree in informatics engineering from the Polytechnic of Coimbra (IPC), institutions in Portugal, in 2016. After the experience as a Web and Software Developer between 2015 and 2018, he currently has a studentship for his Ph.D. research from UC. His research interests include databases, distributed systems, edge computing, cyber-physical systems, and machine learning.

SERGII MYKOLYSHYN received the bachelor's degree from the University of Coimbra (UC), in 2019, where he is currently pursuing the master's degree in informatics engineering, specializing in software engineering. His research interests include databases and cloud/edge computing.

BRUNO CABRAL received the Ph.D. degree (Hons.) in the area of informatics engineering from the University of Coimbra (UC), in 2009. He holds a tenured professor position with the Informatics Engineering Department, UC. He has been an Adjunct Associate Teaching Professor with Carnegie Mellon University (CMU), USA, and a Faculty Member of the dual-degree master's in software engineering (MSE). He is the Coordinator of the Master Program in software engineering with UC and a Senior Researcher with the Centre of Informatics and Systems of the University of Coimbra (CISUC), Systems and Software Engineering Group. His research interests include distributed systems, parallel programming languages, machine learning, and dependable computing. He was either the PI or a Staff Member on multiple EU, Government and privately funded research projects, and frequently works as a Consultant to the software industry.

JORGE BERNARDINO (Member, IEEE) received the Ph.D. degree from the University of Coimbra, in 2002. From 2005 to 2010, he was the President of ISEC (Coimbra Engineering Institute). From 2017 to 2019, he was also the President of ISEC Scientific Council. In 2014, he was a Visiting Professor at CMU. He was the Director of the Applied Research Institute (i2A) of Polytechnic of Coimbra, from 2019 to 2021. He is currently a Coordinator Professor with the Polytechnic of Coimbra, ISEC, Portugal. He has authored more than 200 publications in refereed conferences and journals and participated in several national and international projects. His research interests include big data, NoSQL, data warehousing, dependability, and software engineering.

VASCO PEREIRA received the Ph.D. degree in informatics engineering from the Faculty of Sciences and Technology, University of Coimbra, Portugal, in 2016. He is currently an Assistant Professor with the Department of Informatics Engineering, University of Coimbra, and the Vice-Coordinator of the bachelor's degree in informatics engineering. He is also a Researcher with the Laboratory of Communications and Telematics, Centre of Informatics and Systems of the University of Coimbra (CISUC), where he has been involved in national and European research projects. He is currently involved in projects, such as 5G—Components and Services for 5G Networks (POCI-01-0247-FEDER-024539), MobiWise from mobile sensing to mobility advising (P2020 SAICTPAC/0011/2015) and SOCIALITE—Social-Oriented Internet of Things Architecture, Solutions and Environment (POCI-01-0145-FEDER-016655). His research interests include QoS, performance in wireless sensor networks and the IoT. See his homepage (http://faculty.uc.pt/uc26416) for more details.