Database Model Visualization in Virtual Reality: Exploring WebVR and Native VR Concepts

Roy Oberhauser
Computer Science Dept.
Aalen University
Aalen, Germany
e-mail: roy.oberhauser@hs-aalen.de

Abstract - **Databases are becoming an ubiquitous and integral part of most software as the data era and the Internet of Everything unfolds. Alternative database types such as NoSQL grow in popularity and allow data to be stored and accessed more simply or in new ways. Thus, software developers, not just database specialists, are more likely to encounter and need to deal with databases. Virtual Reality (VR) technology has grown in popularity, yet its integration in the software development tool chain has been limited. One potential application area for VR technology that has not been sufficiently explored is database-model visualization. This paper describes Virtual Reality Immersion in Data Models (VRiDaM), a generic database-model approach for visualizing, navigating, and conveying database-model information interactively. It describes and explores both native VR and WebVR solution concepts, with prototypes showing the viability of the approach.**

Keywords - virtual reality; database visualization; database tools; visual development environments; database modeling; software engineering; WebVR; Benediktine space.

I. INTRODUCTION

This article is an extended version of a former conference publication, see [1] for further details. In our time and the foreseeable future, data has become the most coveted "raw material", in some respects analogous to a gold rush. IDC estimates the current annual data creation rate at 16.1ZB (Zettabytes), and by 2025 163ZB, with embedded data by then constituting nearly 20% of all data created [2]. Cisco estimates there will be 27bn networked devices by 2021 [3]. The ongoing digitalization involving Industry 4.0 and the Internet of Everything will imply a large increase in databases to be able to store and retrieve this data, in particular embedded databases. As data explodes, software engineers are increasingly faced with the daunting task of structuring and analyzing such databases across various DataBase Management System (DBMS) types, including relational and NoSQL types such as document (semi-structured), key-value, wide column (extensible record), in memory, and graph [4].

Faced with developing and maintaining software that integrates some form of data store, software engineers must increasingly deal with analyzing and changing data models. While the original developers may have (had) a clear (and correct to a certain degree) mental model of their actual data model, the maintenance situation is exacerbated by proliferation of relational (mostly SQL) and NoSQL database types. Furthermore, the relatively high turnover rates common in the software industry can result in developers unfamiliar with the data models attempting to quickly comprehend the database structures involved with these software applications.

With regard to comprehension, humans tend to be visually-oriented and can detect and remember visual patterns well. Information visualization has the potential to support human understanding and insight while dealing with resource constraints on the human as well as computer side. While common ways for visually conveying database structures include 2D entity-relationship (E-R) modeling and diagrams [5], these are typically applied to relational databases (RDB), while NoSQL databases may or may not provide a tool that includes visual support. Usually a database product will provide a preferred product-specific tool offering that may be web-based or have a standard 2D graphical user interface (GUI), whereas certain tools can support multiple database products of one specific type (e.g., MySQL Workbench for SQL databases).

Contemporaneously, VR provides new options and capabilities for visual immersion and has made inroads in its accessibility, as prices have dropped and software and hardware capabilities have improved. The VR market has been rapidly expanding, with VR revenue reaching $2.7bn in 2016, with an expected 10-fold increase to $25bn by 2021 [6]. Broad VR usage is still relatively new and the developer market segment small in comparison to the general VR market segment. Thus, software engineers as yet do not have access to integrated VR capabilities in their development tool chains. In this respect, the application of VR to database structures has been insufficiently explored, and a general approach and specific solutions concepts for utilizing native VR and WebVR are not yet prevalent.

This paper contributes Virtual Reality Immersion in Data Models (VRiDaM, pronounced like freedom), a generalized approach to heterogeneous (relational and non-relational) database-model visualization in VR. In this paper we explore two specific VR solution concepts: 1) WebVR in a web browser using a Benediktine-space [7]-[10] visualization paradigm, and 2) a native VR visualization concept. Database models from different data store types can be visualized and navigated (locally or remotely) in VR using a VR headset and controller. We describe principles and features for visualizing, navigating, and conveying database information interactively to support exploratory, analytical, and descriptive cognitive processes [11]. Prototype implementations demonstrate the viability of the approach. This paper extends [1] by contributing a generalized approach and providing both native VR and WebVR solution concepts to address database visualization.

The paper is organized as follows: the following section discusses related work. Section III provides background information. Section IV presents our general solution approach and two solution concepts. Section V describes the implementation. In Section VI, the solution is evaluated and is followed by the conclusion.

II. Related Work

While we found no directly related work, we did find work related to the application of VR for visualizing data in specific areas such as astronomy, biology, geography, etc. Okada et al. use VR to visualize and explore tweet data [12][13]. Moran et al. [14] developed a Unity3D application for geospatial visual analytics of Twitter data. Sun and Wu [15] describe a VR data analytics platform supporting multidimensional data in a geographic based view, specifically presenting factory chemical readings and meteorological data. While VR Juggler [16] provides VR support for developing VR applications, it does not address database modeling and visualization. As to VR approaches for software structure visualization, ExplorViz [17] is a WebVR application that supports VR exploration of 3D software cities using Oculus Rift together with Microsoft Kinect for gesture recognition. As to non-VR visualization, [18] provides an overview and survey of 3D software visualization tools across various software engineering areas. Software Galaxies [19] gives a web-based visualization of dependencies among popular package managers and supports flying, with each star representing a package clustered by dependencies. CodeCity [20] is a 3D software visualization approach based on a city metaphor and implemented in SmallTalk on the Moose reengineering framework. X3D-UML [21] provides 3D support with UML grouping classes in planes. In contrast, VRiDaM focuses on specifically on visualizing database structures.

Database management (DBM) tools are typically DB type-specific and require some installation. Each professional RDB product usually offers its own tool, but since most RDBs support the Structured Query Language (SQL), certain RDB tools can access other RDBs using RDB-specific drivers. For example, MySQL Workbench works across multiple databases and supports reverse-engineering of 2D E-R diagrams. Schemaball [22] provides a 2D circular composite schema diagram for SQL databases. As to 3D RDB tools, DIVA (Database Immersive Visual Analysis) uses stacked rings with rectangular tables attached to them (forming a cylinder), with the tables with the most foreign keys sorted to the top of the stack. Alternatively, stacked square layers of tables can be displayed and 2D E-R diagrams shown. Actual data values are not visualized. For NoSQL databases, each database type differs and the associated DBM tools. One example of a popular wide-column database (WDB) is Apache Cassandra, for which DataStax Studio is a Java-based DBM tool with a web GUI (Graphical User Interface). As to document-oriented databases (DDBs), MongoDB is a popular example and MongoDB Compass, Robomongo, and Studio 3T being example DBM tools. For graph databases (GDB), Neo4j is popular and graph DBM tools include Neo4j admin, Structr, Gephi, Graffeine, etc. In contrast, the VRiDaM approach is more generalized to work across heterogeneous DB types, thus permitting users to only ramp up on one tool, it is cross-platform and provides an immersive web-based VR experience. Furthermore, in contrast to the 3D DIVA or 2D Schemaball, our approach avoids the visual connection 'yarnballs' as the number of connections and tables scale.

Work on big data visualization techniques in conjunction with VR is discussed by Olshannikova et al. in [23]. Herman, Melançon, and Marshall [24] survey work on graph visualization and navigation for information visualization. In contrast, our focus is displaying the database-model structure, not on displaying and analyzing large amounts of data per se.

III. Background Information

A. Benediktine Space

Benediktine space transforms or maps an information object’s attributes to extrinsic (e.g., Cartesian coordinates, time) and intrinsic (e.g., size, shape, color) information spatial dimensions. To keep objects from overlapping, mapping principles include exclusion, maximal exclusion, scale, and transit [7]-[10].

B. WebVR

WebVR is a Mozilla JavaScript API that enables web browsers to access VR hardware. A-Frame is an open source framework for displaying VR content within HTML. It is based on an entity component system architecture in which each object in a scene is an entity (a container) consisting of components that provide desired functionality or behavior for that entity. A-Frame uses the three.js library to provide 3D graphics display in the browser and simplify WebGL programming. Systems are data containers. In contrast to game or PC station VR solutions, the use of VR within web browsers is relatively new, thus in deciding on the visualization techniques to use we considered the limitations and available capabilities and performance offered with WebVR for standard hardware (such as notebooks) that developers might use. Visualization considerations included selecting what and how many objects are displayed at any given time. Furthermore, in contrast to games, there is no real upper limit on the number of simultaneous entities a database or database model may have, which may overtax the computing and rendering capabilities and have negative impacts on the frame rates in VR, making the experience unsatisfactory and possibly resulting in VR sickness.

To characterize WebVR performance, we experimented with the implementation, some measurements of which are shown in Table I. They are averaged across 10 measurements for 10 tables with 50 columns each on a notebook with Intel Core i5-3320M 2.6Ghz, 8GB RAM, Win7 x64, Intel HD Graphics 4000, Chrome 60.0.3112.113 and A-Frame 0.6.1.

TABLE I. AVERAGE A-FRAME PERFORMANCE (FRAMES PER SECOND)

Variants	Loading (fps)	Running (fps)
spheres, no edges	25	61
spheres, with edges	21	53
labeled spheres, no edges	11	19
circles, no edges	25	57
spheres, no edges (10x nodes)	3	12

Based on our experience and measurements with the A-Frame implementation, the following performance findings were made and affected our solution: 1) the number of rendered objects has a major impact on performance, 2) edges have a negative effect on performance, 3) text labels have a severe impact, 4) circles and spheres are equivalent.

For Finding 1, that implies that only the minimum number of objects should be displayed. Thus, rows (data values) will only be shown for selected column, not for all columns. Due to Finding 2, objects will be displayed without edges and connectors between objects will be avoided (force-graph). Due to Finding 3, text will be limited and the data value only shown when a circle (tuple) is selected.

IV. SOLUTION APPROACH

Visualization, especially VR with its wide viewing angles, can leverage peripheral vision for information, whereby visual data is both consciously and unconsciously seen and processed. If leveraged well, visualization can be cognitively easier in providing insights than an equivalent textual analysis would require. Traditional text-centric tabular formats or boxes in E-R diagrams do not explicitly take advantage of such visual capabilities. Also, if the contents of each database table were visualized as a rectangular 2D object, as it scales both in number of tables and the size of various tables, lucidity issues occur that nullify the advantage of VR visualization.

To provide some background context for our VRiDaM approach, we describe several perspectives that were considered. Information can be grouped and large amounts of information provided in a relatively small amount of graphical space. Yet humans are limited in their sensory perception and focus, and thus visualization considerations include: determining the proper balance for what to place into visual focus in which context, what to place into the periphery, what to hide or show, and to what extent and at what point should what be visualized. To develop actionable insights from visualization, the knowledge crystallization cognitive process involves acquiring information, making sense of it, creating something new, and acting on it [26]. Regarding visual perception, gestalt psychology [27] is based on the four principles of emergence, reification, multistable perception, and invariance. Formulated gestalt grouping laws regarding visual perception include proximity, similarity, closure, symmetry, common fate, continuity, good gestalt, past experience, common region, and element connectedness. For visual representation, we considered Don Norman's design principles, in particular affordance, consistency, and mapping [28]. Other concepts considered were [10] information space, cognitive space, spatialization, ordination, and pre-attentive processing, which refers to the accumulation of information from the environment subconsciously [29]. Visualization techniques explicitly analyzed with regard to their appropriateness for displaying data models in VR included books, cone trees, fisheye views, information cubes, perspective walls, spheres, surface plots/cityscapes/3D bar graphs, viewpoints, workspace/information space/3D rooms, worlds in worlds, and Benediktine space.

Our VRiDaM solution approach is shown in Fig. 1 and involves transforming the raw data and its metadata to internal structures (the first two being purely data forms), and then mapping these to visual element structures, and transforming these to the views seen be the user (the last two being visual forms). To adjust the views, the user provides interaction input affecting one of the aforementioned transformation steps.

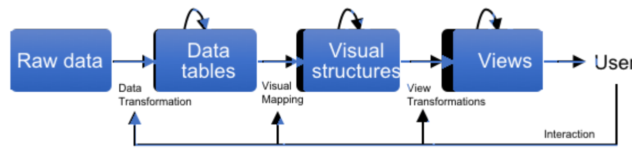

Figure 1. VRiDaM architecture based on the information visualization reference model by Card et al. [30].

Our VR approach works across different databases products and database types (SQL and NoSQL), thus familiarity with a single VR app could be leveraged across the various database types. Alternatively, currently developing unique VR app tools for each database and/or database type would be exorbitant relative to the number of software engineers that have VR capabilities and have database-model interests, and inefficient from a learning/training perspective.

V. SOLUTION CONCEPTS

Native and browser-based VR provide different capabilities or degrees thereof, and thus have a different potential in what can be done and what constraints and issues arise. Typically, a native VR setting will provide a more mature platform with greater VR capabilities, better hardware integration, and better performance optimization. The platform is optimized for VR, whereas a WebVR solution tries to add in VR into a browser-focused experience that was not originally designed to support VR. Thus, each of our solution concepts explores and applies different solution principles to make the most of the potential within the constraints of the environment.

In both concepts, cubes are used to represent database tables (collections for document stores, or labels for graphs), analogous to cube furniture that can be used as a table.

A. VRiDaM-N Concept

For the VRiDaM-N solution concept, spatial proximity between cubes was not used to indicate closer relations, rather an initial circular placement was used and lines/pipes used to indicate which are related. The following principles (P:) were applied:

Every table is rendered as a generic cube with a rotating label on top. This permits the use of the cube sides for graphical icons or pictures to support a more visual experience from various angles, yet the name can usually be well read from the various angles.

Customizable icons placed on the cube sides can be used to indicate the kind of data held in a table. These can provide a custom graphical indicator as to the kind of data the table holds, without needing to read the table label. Thus, users can more rapidly identify tables. From the rear, the icons are not seen, rather attribute (column) names, but the table labels can be read. A default icon (empty table) can be used when no other is specified.

Attributes (columns, or keys for document or graph stores) are indicated as names are placed spatially behind the table cubes as an attached cube. In case the attribute names are of interest, the user can make these visible and navigate to the sides or rear of the table cube to read the attribute names, with a longer list resulting in a deeper cube, providing a visual cue that the attribute set is more extensive. When viewed from the front, the attribute names are not visible, thus simplicity is provided based on perspective when desired. Because the native rendering of text is better and more readable from various angles, we chose to stack the attribute labels on the sides and rear and keep them visible, rather than spatially separating them as a network as we did in the WebVR concept. An indicator (red colored attribute name text) can be used as a differentiator to indicate foreign keys, and similarly indexes could be shown with a different color.

Relations are visualized as a network. Pipes (equivalent to 3 dimensional lines) are used to indicate foreign key relations.

Custom grouping. Logically-related tables can be grouped by the user as desired within colored label frames.

Virtual tablet for details and accessing data tuples (rows). To view table details or raw data, the data content (tuples or rows) can be viewed via a horizontally expandable virtual tablet that can show all attributes, while supporting vertical scrolling to view any data the table holds. We decided against showing all data on a plane, since deeper text cannot be read anyway and the scrollbar can indicate how much data exists.

B. VRiDaM-WebVR Concept

The WebVR solution concept utilizes a browser-based WebVR implementation to enable cross-platform access to VR content. It assumes the user has a VR headset, but not necessarily VR controllers. Software engineers often work across different operating systems (Windows, Linux, etc.), and this permits them to utilize the app from any platform with appropriate WebVR browser support. With the WebVR solution, we struggled with readability and adequate performance when much text was rendered, and navigation and flythrough was not as smooth, and thus chose utilizing other visual components to convey information and to reduce the amount of visible text.

For the WebVR solution concept, we applied the following principles (P:):

Leverage spatial visualization in VR using a Benediktine spatial object placement approach. Our approach leverages the additional dimensional visualization and navigational capabilities VR provides (within current limitations of WebVR). Specifically, we utilize a Benediktine space visualization technique [7] with visual object spatial placement based on extrinsic spatial dimensions, whereas other entity properties are represented by intrinsic dimensions (form, size, color, etc.). The principle of exclusion ensures no two objects occupy the same spatial location, and the principle of maximal exclusion ensures that different data items are separated as much as possible [8].

Leverage dynamic self-organizing force-directed graph visualization to indicate the strength of relationship between objects via proximity. For visualizing relations, dynamic self-organizing force-directed graph placement [25] can be used in place of connectors to indicate via proximity more strongly related entities from those that are less- or unrelated. This is combined with visual highlighting of related objects when selecting an object. In this way we intend to avoid the "ball of yarn" issue with visual connectors as database models scale, or that a circular placement of many tables is no longer perceivable or meaningful to the user.

VI. Implementation

For the following prototype implementation screenshots, the Northwind Trading sample database consisting of 13 tables and 6635 records was used for testing and visualizing data. Additional features that can be added to our prototype in future work includes support for visualizing constraints and indexes.

A. VRiDaM-N Concept

To implement the native VR solution concept, the Unity game engine with the SteamVR plugin and runtime was chosen due to its multi-platform support, direct VR integration, popularity, and cost. For testing with VR hardware, we used the HTC Vive, a room scale VR set with a head-mounted display and two wireless handheld controllers tracked by two base stations.

Fig. 3 shows the default circular placement after loading with the front view. Cubes in purple are labeled on top with a rotating label that aligns to the camera position. The cubes can have a custom icon (such as us_states which shows the outline of the continental USA), or a default empty table icon such as categories. Table relations are shown via the connecting black pipes. The black cubes showing attribute names can be turned off if desired. Because we wanted to emphasize the icons and increase contrast, we chose a white space as the background rather than black for instance.

Fig. 4 shows a partial side perspective. The black cubes behind the purple table cubes represent attributes and contain the attribute names, with the higher black cubes representing a larger set of attributes (e.g., see employees, orders, or customers vs. the smaller shippers table). Thus, one has a visual indication of which tables have fewer attributes.

Fig. 5 shows the view from the rear, showing attribute names and any foreign key relations via red text and pipes drawn to the related key.

In Fig. 6, a virtual tablet is shown, that shows the data contents and details of the selected table. This tablet can be placed where spatially desired, and can expand when needed to show all attributes of a given table.

Fig. 7 shows the grouping capability, where the color and label of a frame can be chosen, and the cubes representing tables can be moved and logically organized as desired by the user.

Fig. 8 shows a side view where attributes can be seen within the grouping frames.

The solution concept is generic; SQL support in the native VR prototype was implemented, but due to resource and schedule constraints support for NoSQL databases will be implemented in future work.

B. VRiDaM-WebVR Concept

To implement the WebVR-based prototype, A-Frame and D3.js were utilized, which produces dynamic, interactive data visualizations in web browsers. For a self-organizing force-directed graph, our implementation uses the d3-force-3d physics engine from D3. Firefox and Chrome were used as web browsers. For database connectivity, the Spring Framework 4.3.1 was used and tested with PostgreSQL, MSSQL, MongoDB, Cassandra, and Neo4j. Content for the force-directed graph component was transformed to JSON format. Fig. 9 shows the class diagram regarding database integration.

The following visual object forms were selected:

Cubes are used to represent database tables (collections for document stores, or labels for graphs), analogous to cube furniture that can be used as a table (Fig. 10).

Cylinders are used to represent database attributes (columns) (set of similarly typed data, known as keys for document stores or graphs), analogous to columns in a building (Fig. 11).

Planes are used for projecting the database data records (rows, tuples, or entries - the actual data values) since these can be very large in both columns and records and would thus occupy a large amount of VR space as seen in Fig. 12). A plane supports maximum readability and permits VR navigation around it.

As to navigation and interaction, users can move objects as desired using standard VR controllers (we used an HTC Vive) or can use a mouse and keyboard. As seen in Fig. 13, besides using spatial proximity to indicate closer associations, if a user selects an object, that object and all its directly referenced objects are highlighted. A key image is provided as an affordance and, if selected, a popup shows the primary and foreign keys names. If desired, lines can optionally be used to emphasize relations as shown in Fig. 14.

The configuration menu is overlaid and can be used to connect to a database and query (e.g., SQL, Cypher, etc.) by typing on the keyboard and executed via enter. In order to quickly find a table, they are listed on the menu for selection and navigation to the visual object.

Fig. 15 shows the VRiDaM visualization for MongoDB document store with dbkoda example data [26] (a Northwind port was no longer available), while Fig. 16 shows Neo4j graph database with Northwind example data.

VII. EVALUATION

To evaluate VRiDaM, we compared its usage with a typical 2D database tool, DbVisualizer 10.0.13 (Free). The Northwind database was loaded to provide an impression of the sample model's complexity – the figures are not meant to be readable. Fig. 17 shows the hierarchic view, Fig. 18 the circular view, Fig. 19 the circular view with table names only, Fig. 20 the organic view, and Fig. 21 the orthogonal view.

A convenience sample of eleven computer science students was selected. All indicated they had some familiarity with SQL and they lacked NoSQL experience, so we chose to compare VRiDaM-WebVR with the common SQL tool DbVisualizer. Three had not experienced VR before. One experienced VR sickness symptoms and thus only the remaining ten were included in the results. The subjects were randomly selected to start in either VR mode (6) or the common tool (4). PostgreSQL Version 10 with the Northwind sample database was used. Java 8 update 151, Apache Tomcat v8.0, AFRAME 0.8.2, Firefox 61, and SteamVR Version 2018-05-24 (1527117754).

These database tasks were given verbally and equivalent but not the exact same five questions asked in the other mode:

1) Which tables have a relation to table X?
2) To which table(s) does the table X have a relation?
3) What columns does table X have?
4) What are the foreign or primary keys of table X?
5) What are the keys in table X?

TABLE II. VRIDAM VS. DBVISUALIZER TASKS (AVERAGED)

	VRiDaM-WebVR	**DbVisualizer**
Task duration (mm:ss)	4:48	1:46
Cumulative answers given (total/incorrect/missing)	130/6/4	140/1/6
Task correctness	92%	95%

The tasks results are shown in Table II. VRiDaM-WebVR task duration was 2.7 times longer, and this can be expected since VR requires navigation time through space that 2D tools do not incur. The number of correct answers across the five tasks were 13 in VR and 14 in DbVisualizer, with ten subjects resulting in 130 or 140 cumulatively correct answers respectively. These longer VR task durations may be acceptable for certain user scenarios, and gives insight into what liabilities can be expected. A correctness difference of 3% across ten subjects is not necessarily significant and shows that the users were able to immerse themselves within minutes into a Benediktine space paradigm and perform tasks effectively. The task correctness differences could be attributed to personality, human alertness, distraction, or other factors beyond the paradigms or VR influence, as only 4 subjects in VR and only 3 subjects in non-VR were responsible for all errors, the rest had no mistakes.

Subjective impressions are shown in Fig. 2 for VRiDaM-WebVR intuitiveness and suitability of the controller interface and visualization as well as overall enjoyment. We note no significant difference between the interaction and the visualization intuitiveness or suitability. Only one subject preferred VRiDaM-WebVR. This may indicate that more training and experience would be needed for them to feel more comfortable with a WebVR tool than with a 2D tool. Various debriefing comments indicated that the Benediktine spatial arrangement was either liked or not an issue for the subjects. When debriefed about what they liked about VRiDaM-WebVR, comments included that it was a better database-model visualization, that tables were real objects instead of text boxes, how tables where displayed in space, and the highlighting effect.

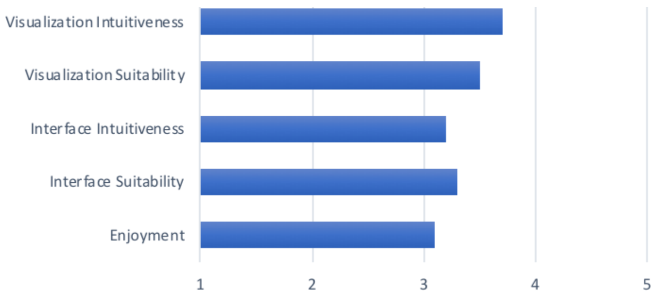

Figure 2. Average responses for VRiDaM-WebVR (scale of 1 to 5 with 5 best).

The evaluation shows some of the challenges in utilizing VR for database-model visualization and interaction. VR object interaction is not standardized, nor do users have familiarity and experience with it as they do with 2D mouse-based user interfaces. While VR enables new immersive paradigms and metaphors, these are not necessarily immediately intuitive. VR movement (moving the camera perspective) is more time consuming than scrolling or zooming a 2D perspective. For simpler tasks, VR tends to require more interaction time to accomplish the same task and thus entails efficiency costs. A 3D space permits objects to hide other objects, and for opaque objects requires movement to determine that no other objects are hidden. Given that the subjects were already familiar with E-R diagrams and SQL tools, yet had no prior training with VRiDaM-WebVR and Benediktine space, we are satisfied with the ratings on suitability and intuitiveness.

VIII. Conclusion

This paper contributes VRiDaM, an immersive VR database visualization approach. Since native VR and WebVR have different capabilities and maturity levels at this time, two separate solution concepts were described and their principles elucidated. The VRiDaM-N, since it has better textual rendering capability and performance, can render comprehensive database information equivalent to common 2D graphical database tools, while permitting an immersive fly-through experience. Furthermore, customizable table icons provide a more visual annotation of what a table contains. For performance, the VRiDaM-WebVR solution concept seeks to reduce the required rendering overhead of objects and text, and avoids the risk of connection "yarn-balls" associated with other techniques that visualize many relations by leveraging the spatial locality of Benediktine space. The prototypes showed the feasibility of the native VR and WebVR solution concepts. The empirical evaluation of VRiDaM-WebVR showed it to be less efficient for equivalent analysis tasks while the correctness was slightly worse. Intuitiveness, suitability, and enjoyment were given a better than neutral rating on average. One case of VR sickness occurred and we hope to address it in future work.

One ongoing challenge for a generic tool approach is the plethora of non-standardized interfaces to NoSQL and other databases. However, by providing driver plugins we believe that the adaptation overhead is small in relation to the advantages of a VR visualization that VRiDaM brings. Future work includes a more comprehensive empirical study and will investigate various optimizations to improve usability, performance, and scalability.

APPENDIX

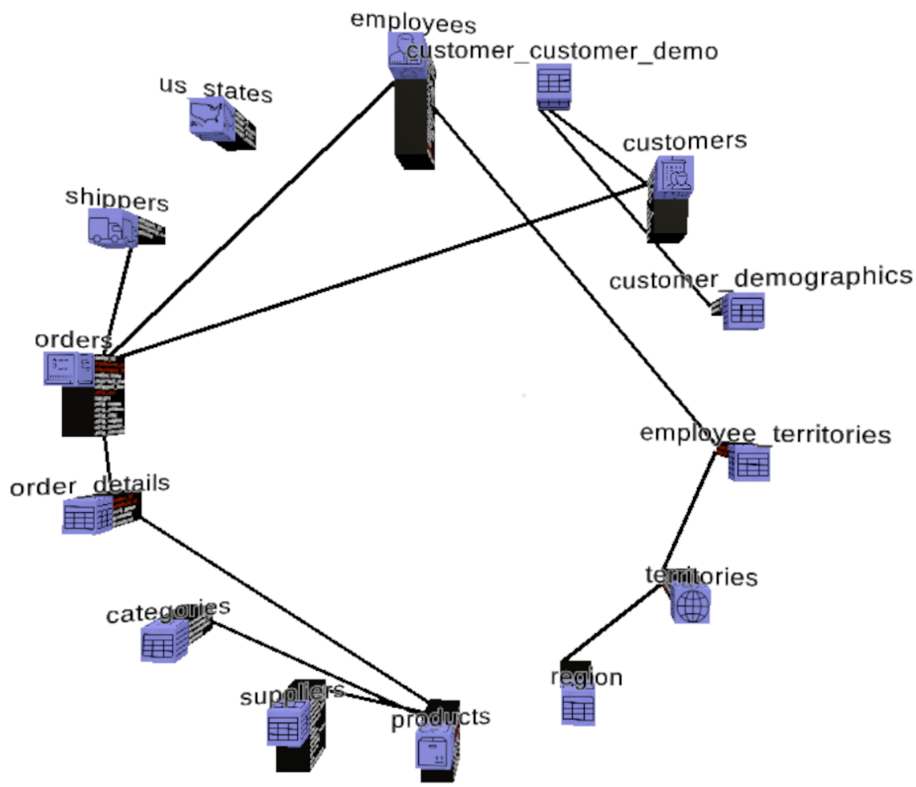

Figure 3. VRiDaM-N front view showing Northwind tables and icons.

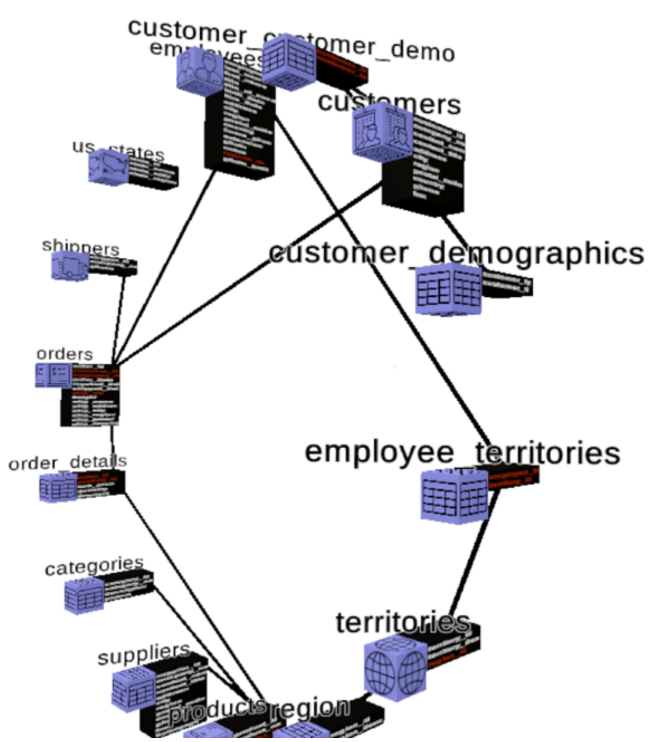

Figure 4. VRiDaM-N side view showing Northwind tables with attributes.

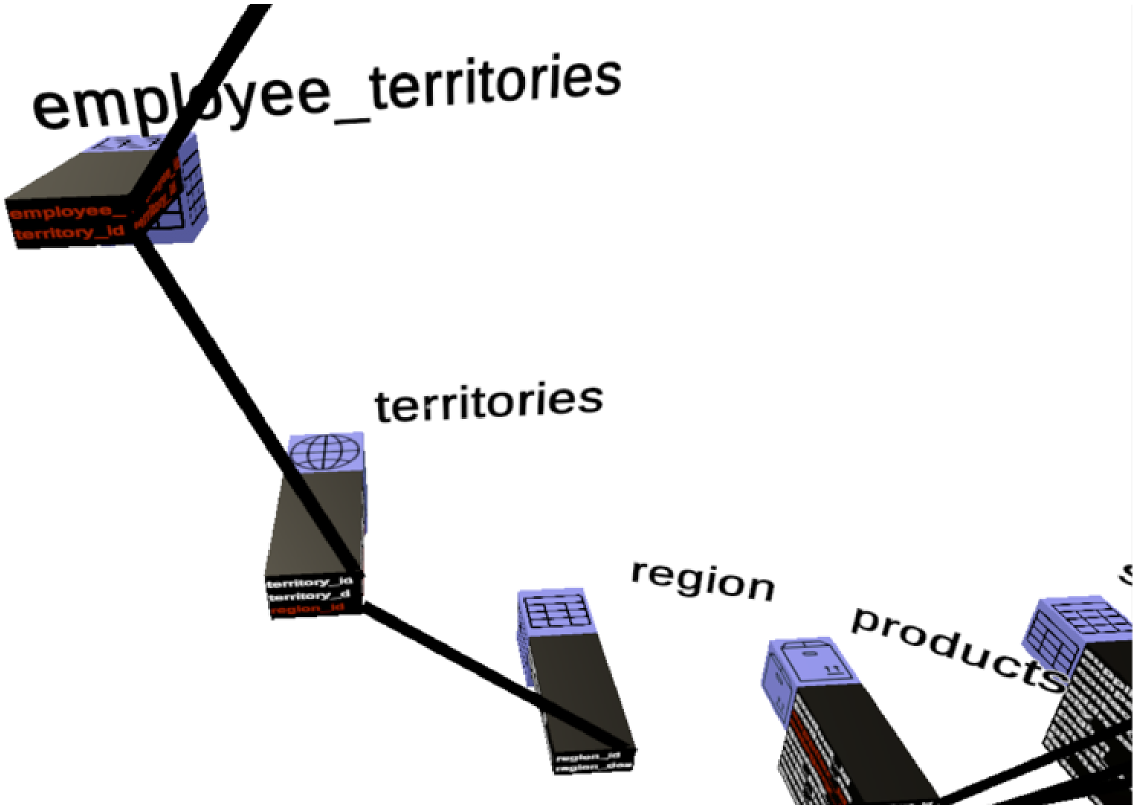

Figure 5. VRiDaM-N rear view showing key relations.

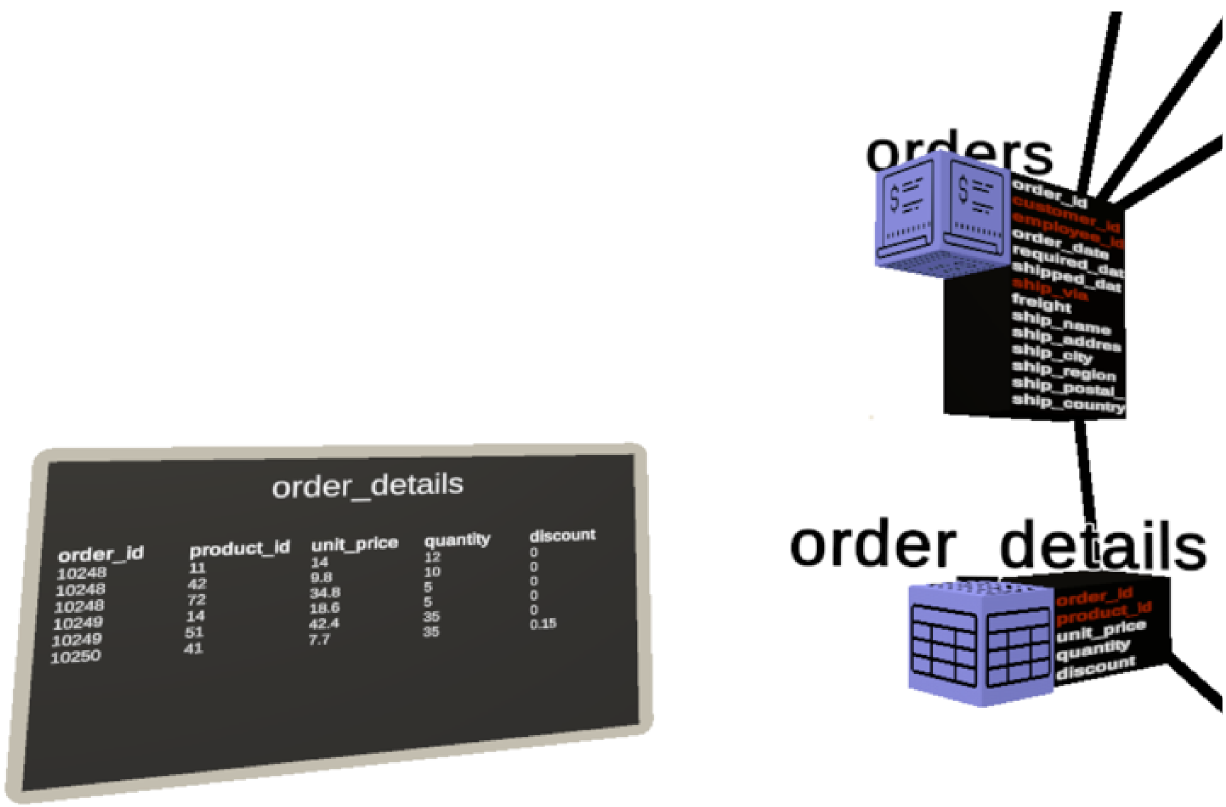

Figure 6. VRiDaM-N with Northwind database showing virtual tablet with row/tuple data for order_details table.

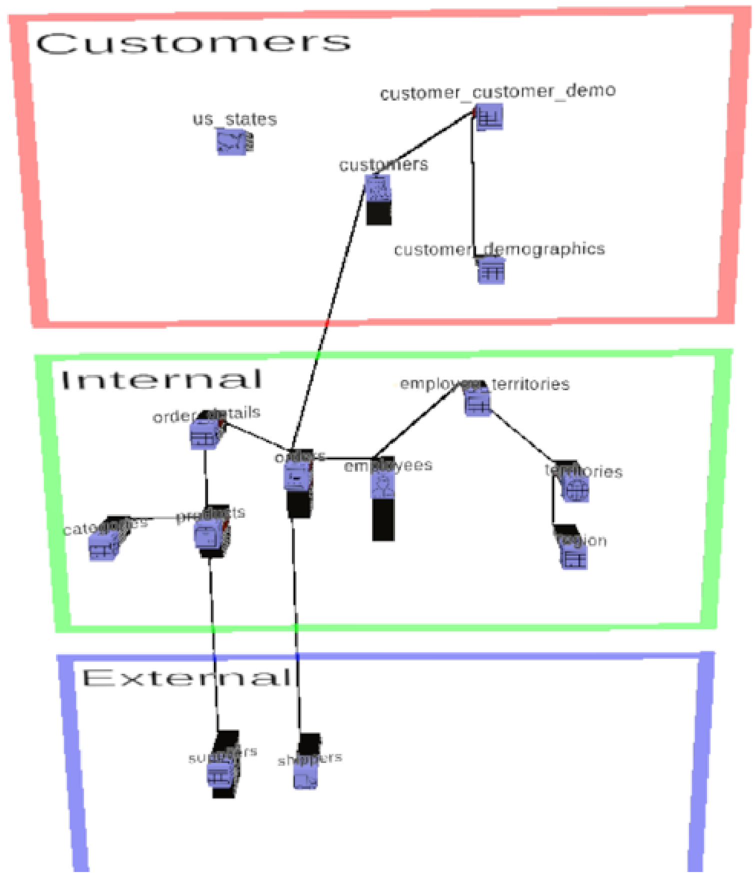

Figure 7. VRiDaM-N front view with Northwind database showing grouping and placement capability.

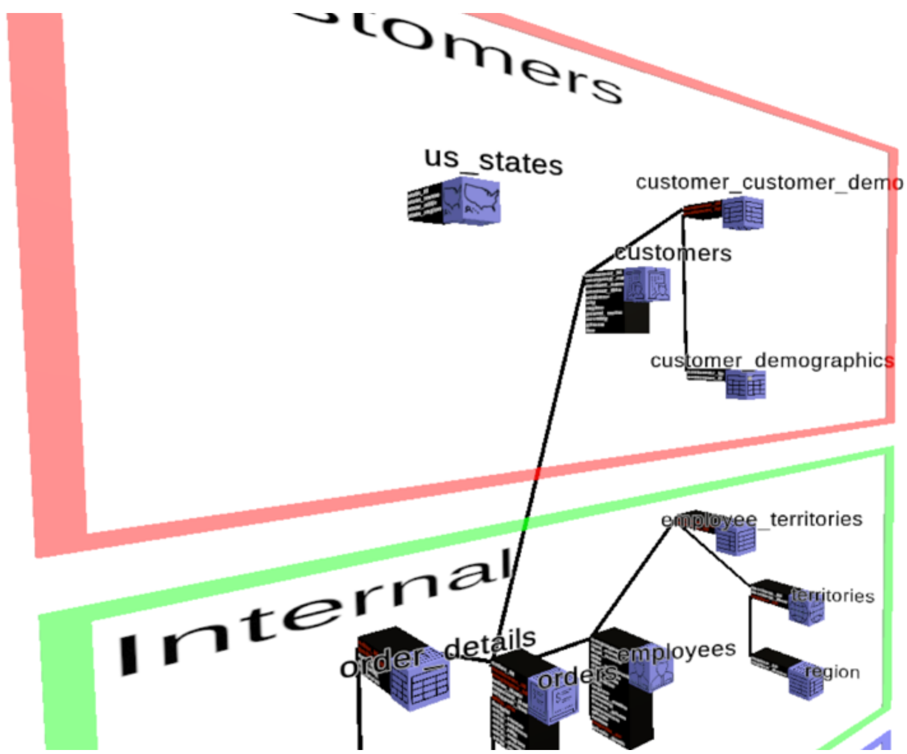

Figure 8. VRiDaM-N with Northwind database showing grouping with attribute names visible from the side.

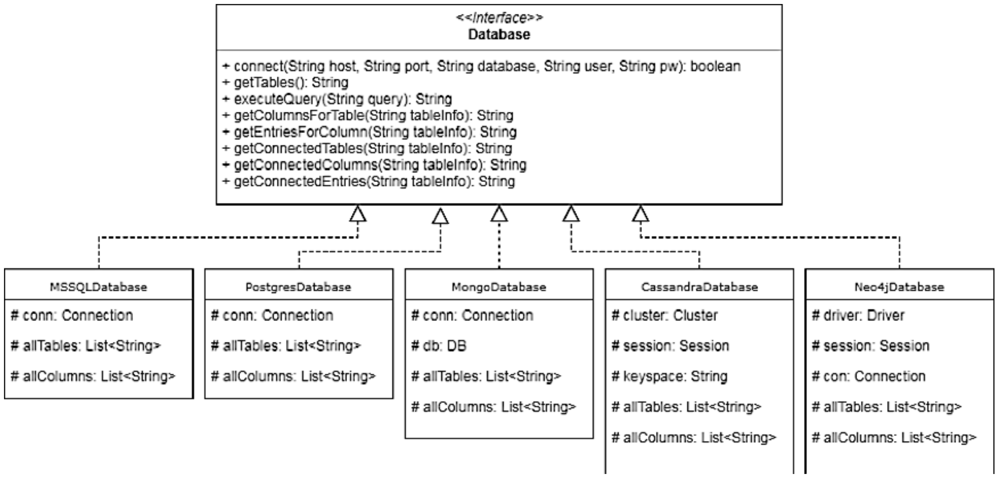

Figure 9. Class diagram of database integration.

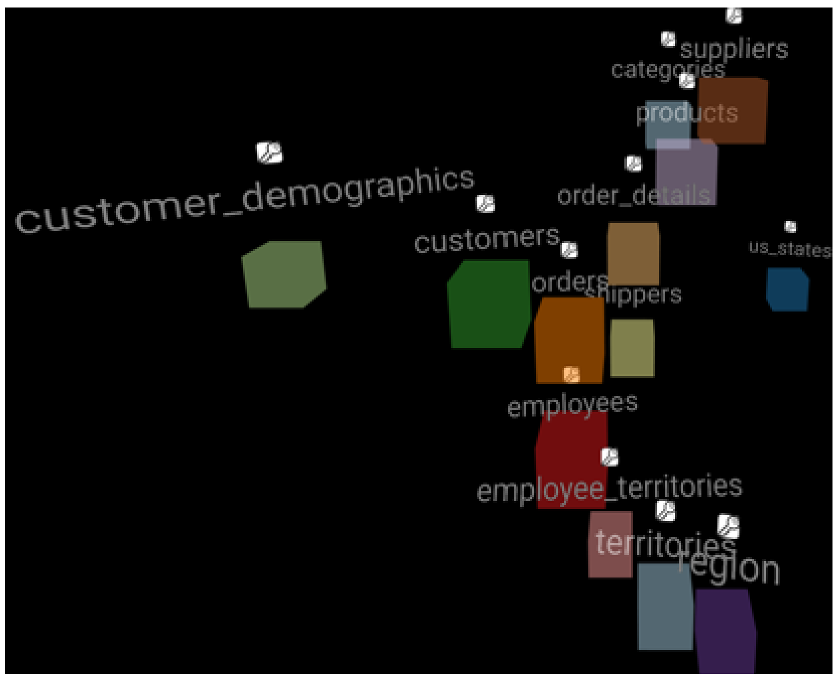

Figure 10. VRiDaM showing Northwind tables from PostgreSQL in Benediktine space.

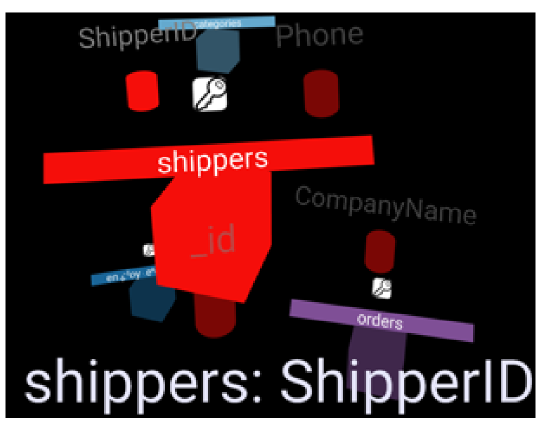

Figure 11. Columns visible orbiting selected table (identical color).

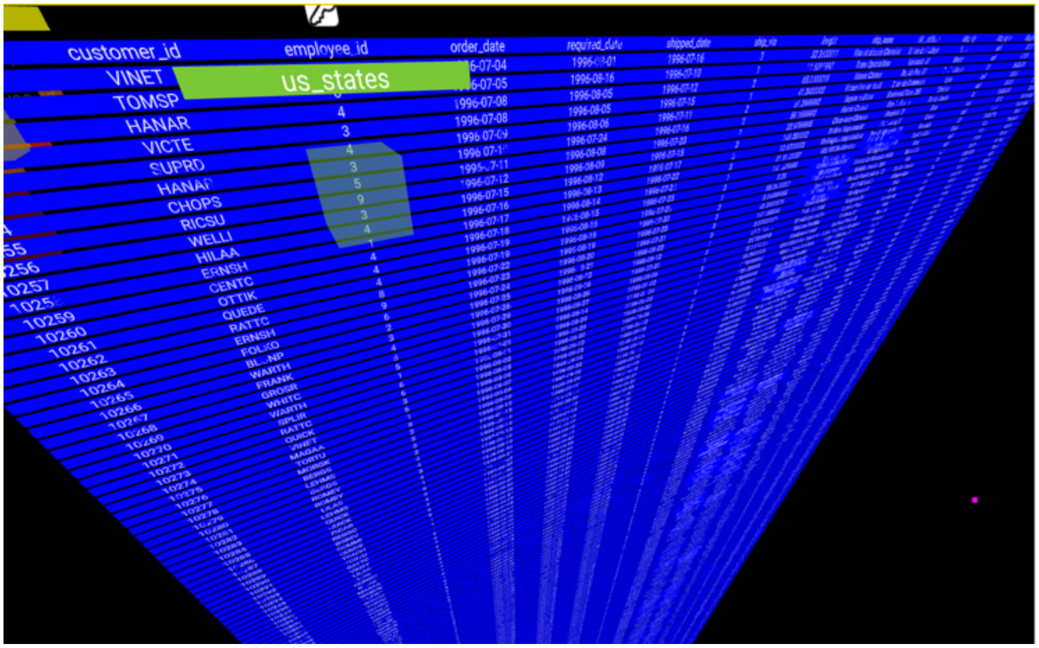

Figure 12. Table records (tuples) projected onto a plane.

Figure 13. Primary and foreign keys for table shown as popups.

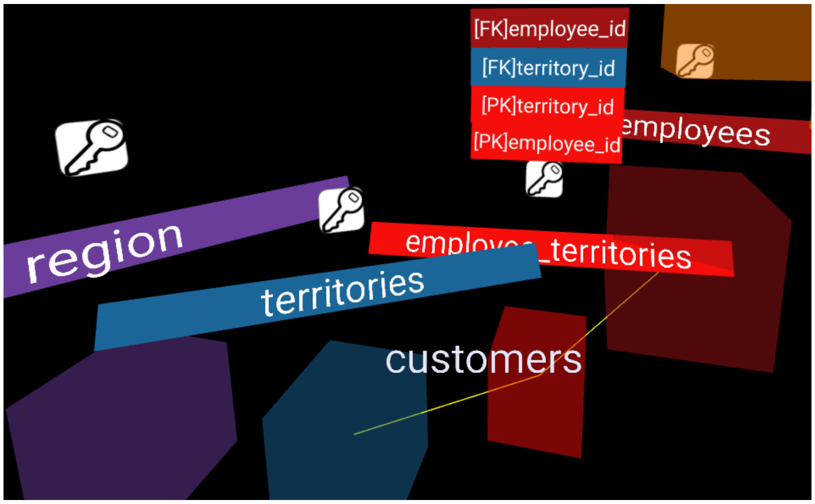

Figure 14. Example optional relation visualization using lines.

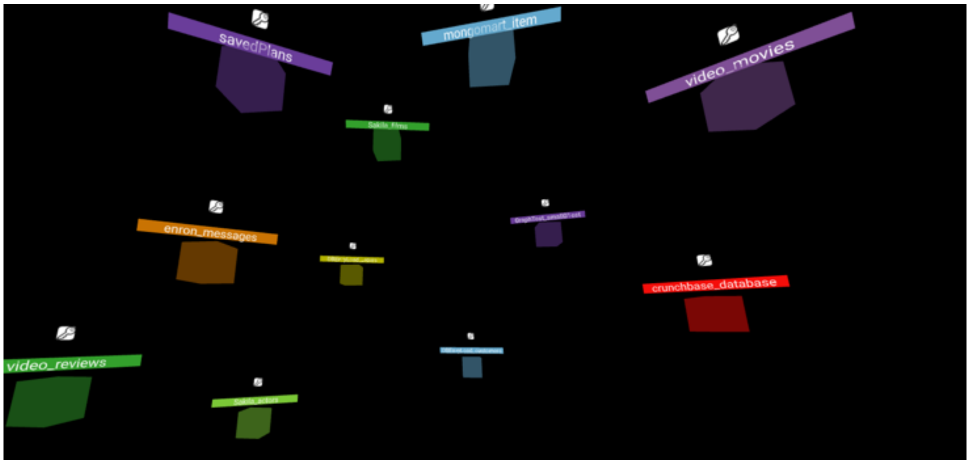

Figure 15. VRiDaM of MongoDB document store with dbkoda example data [26], showing spatial orientation and not intended to be readable.

Figure 16. VRiDaM of the Neo4j graph database with Northwind example data.

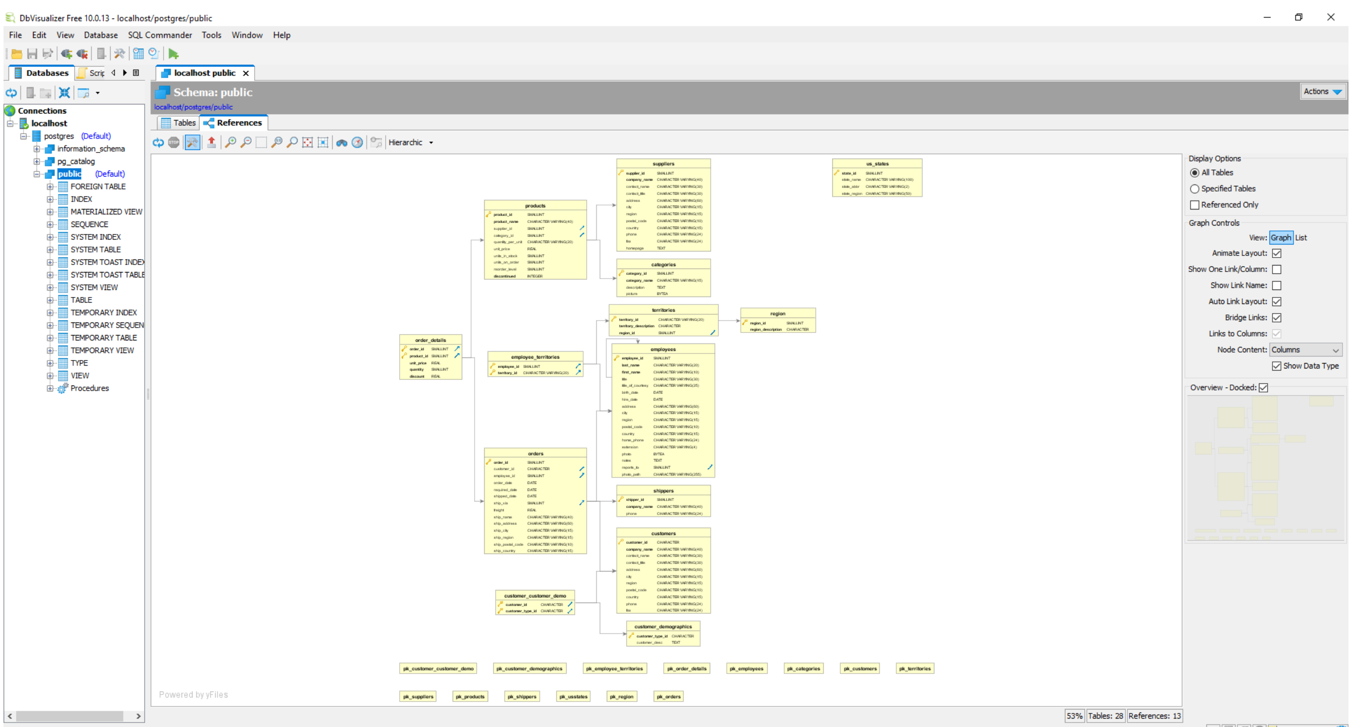

Figure 17. Northwind Traders in DbVisualizer in the hierarchic view.

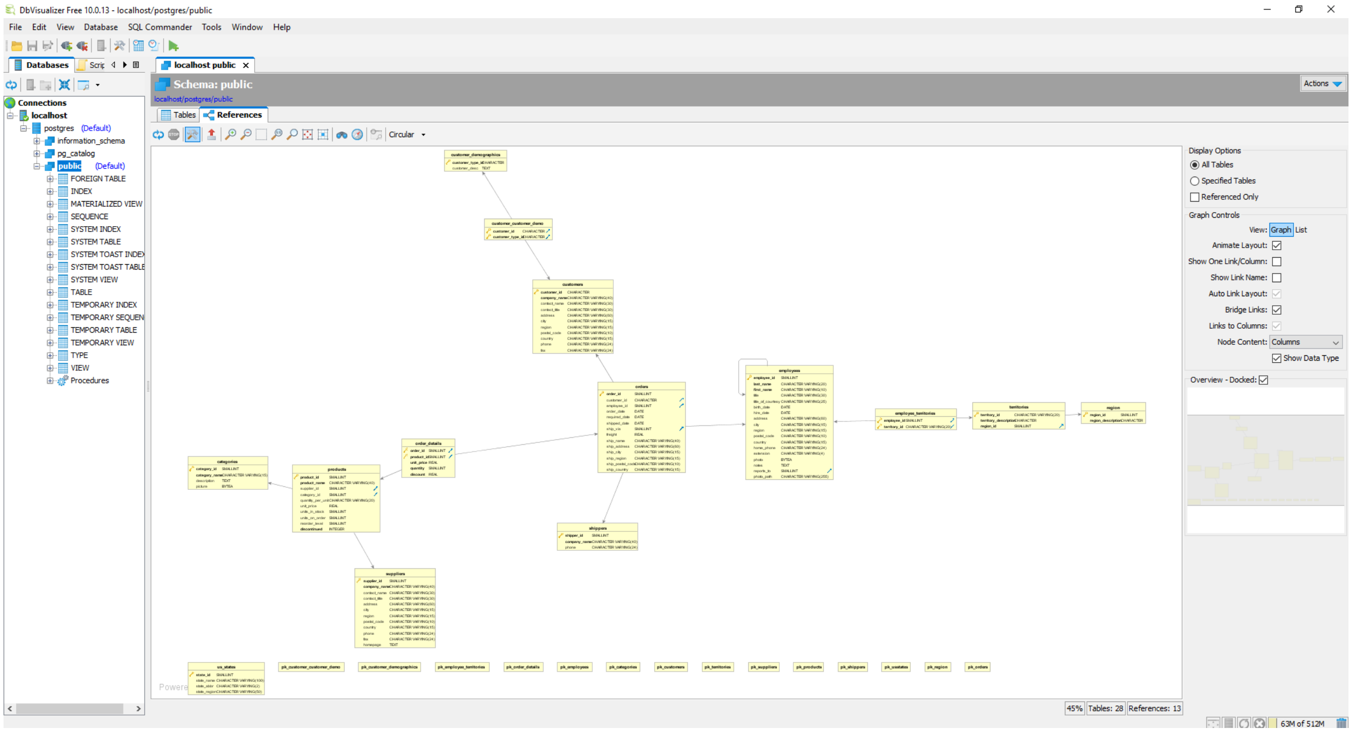

Figure 18. DbVisualizer Circular View with Northwind Traders data model.

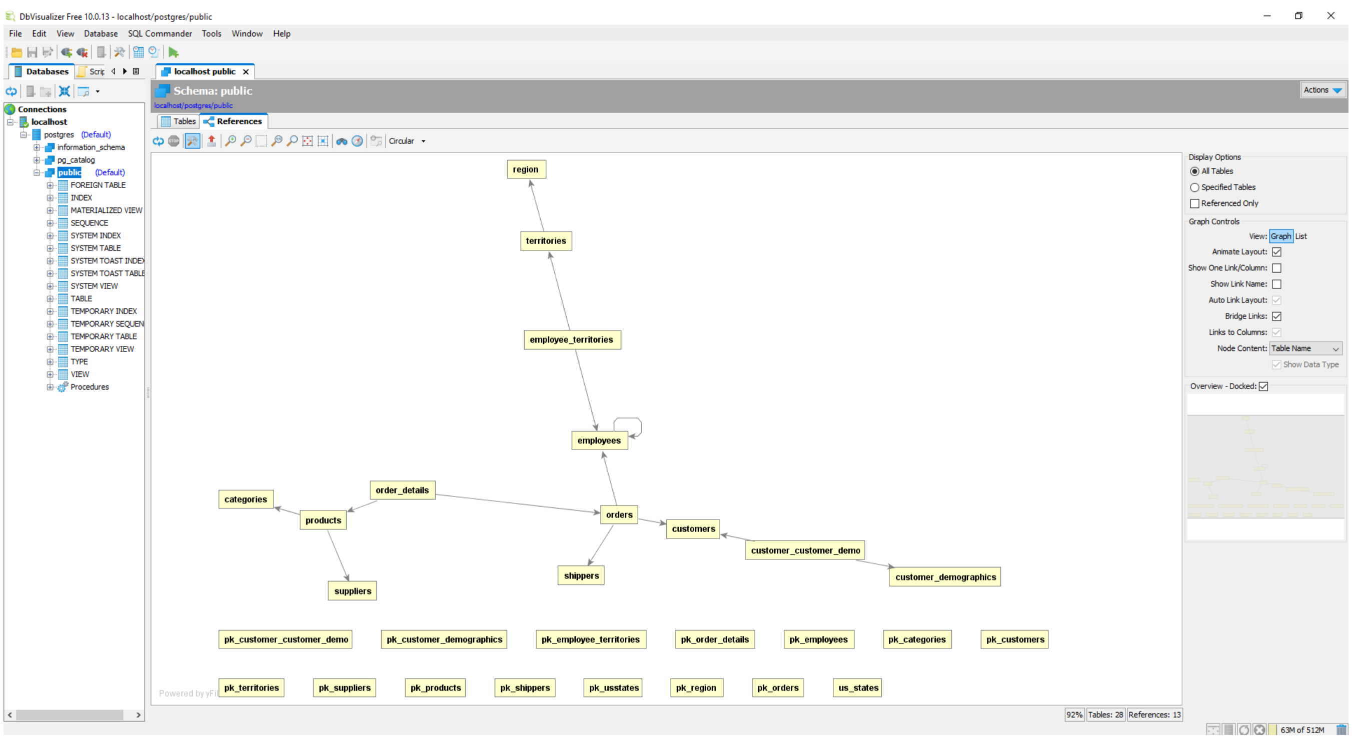

Figure 19. DbVisualizer in circular view (table names only).

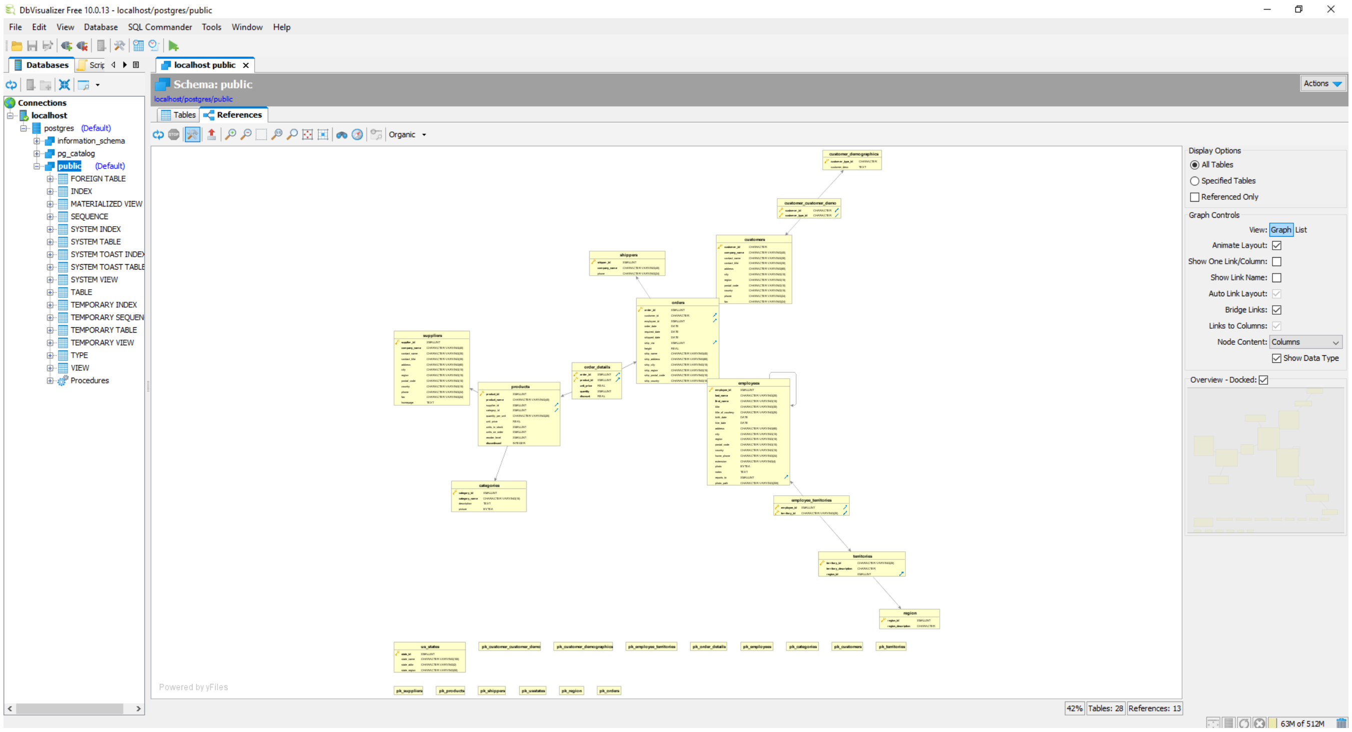

Figure 20. Northwind Traders data model in DbVisualizer in organic view.

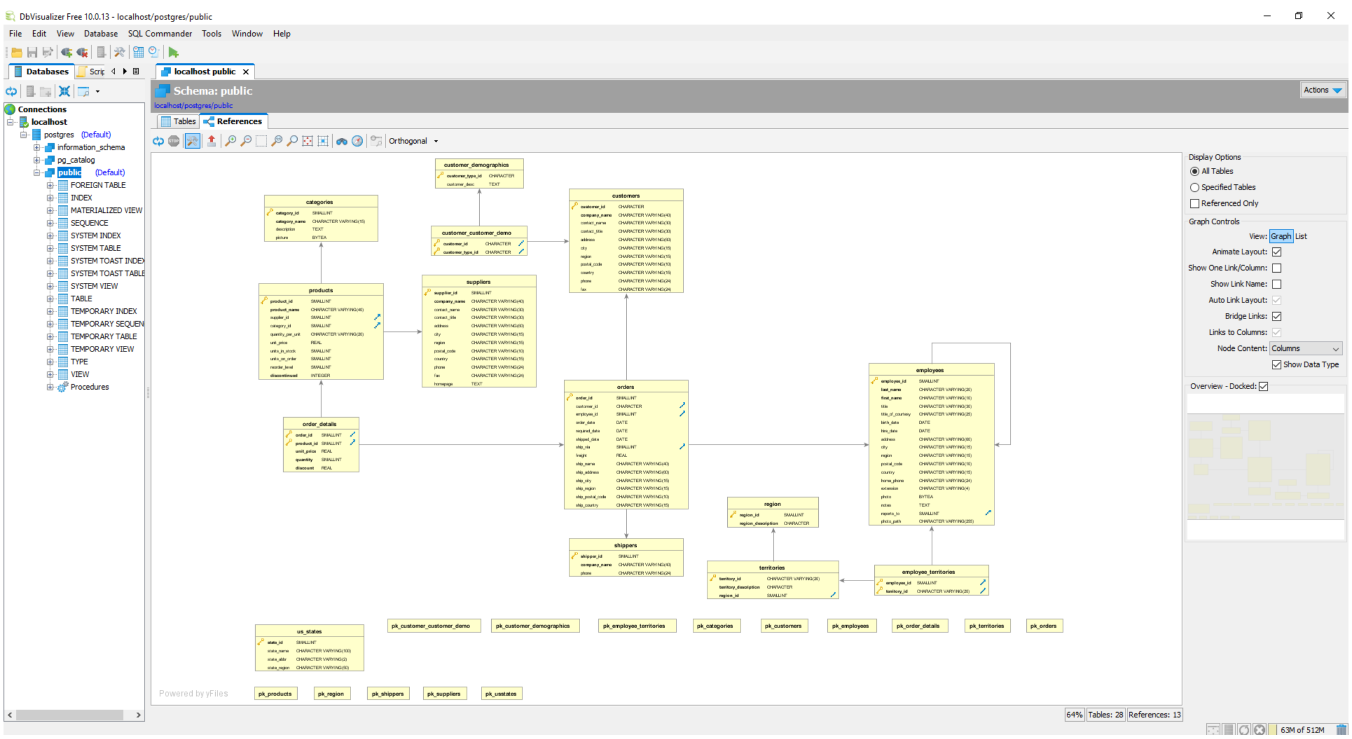

Figure 21. Northwind Traders data model in DbVisualizer in orthogonal view.

ACKNOWLEDGMENT

The author thanks Christoph Bauer, Joshua Lutz, and Camil Pogolski for assistance with the design, implementation, and evaluation.

REFERENCES

[1] R. Oberhauser, "Database Model Visualization in Virtual Reality: A WebVR and Benediktine Space Approach," In: Proceedings of the Thirteenth International Conference on Software Engineering Advances (ICSEA 2018). IARIA, pp. 108-113, 2018.

[2] D. Reinsel, J. Gantz, and J. Rydning, "Data Age 2025: The Evolution of Data to Life-Critical," IDC Whitepaper (April, 2017).

[3] Cisco. The Zettabyte Era: Trends and Analysis. (Jun 7, 2017). Whitepaper. Available from https://www.cisco.com/c/en/us/solutions/collateral/service-provider/visual-networking-index-vni/vni-hyperconnectivity-wp.html 2019.12.04

[4] R. Cattell, "Scalable SQL and NoSQL data stores," Acm Sigmod Record 39, no. 4, pp. 2-27, 2011.

[5] P. Chen, "Entity-Relationship Modeling: Historical Events, Future Trends, and Lessons Learned," In Software pioneers. Springer-Verlag, pp. 296-310, 2002.

[6] T. Merel, "The reality of VR/AR growth," Jan 11, 2017. Available from http://techcrunch.com/2017/01/11/the-reality-of-vrar-growth/ 2019.12.04

[7] M. Benedikt, "Cyberspace: some proposals," In Cyberspace, Michael Benedikt (Ed.). MIT Press, Cambridge, MA, pp. 119-224, 1991.

[8] P. Young, "Three Dimensional Information Visualisation," Technical Report, University of Durham, 1991.

[9] U. Fayyad, A. Wierse, and G. Grinstein (Eds.), Information Visualization in Data Mining and Knowledge Discovery. Morgan Kaufmann, 2002. ISBN: 1558606890.

[10] S. Fabrikant and B. Buttenfield, "Formalizing Semantic Spaces for Information Access," Annals of the Association of American Geographers, 91(2), pp. 263-280, 2001.

[11] D. Butler, J. Almond, R. Bergeron, K. Brodlie, and R. Haber, "Visualization reference models," In Proc. Visualization '93 Conf., IEEE CS Press, pp. 337-342, 1993.

[12] K. Okada, M. Yoshida, T. Itoh, T. Czauderna and K. Stephens, "VR System for Spatio-Temporal Visualization of Tweet Data," In Proc. 2018 22nd International Conference Information Visualisation (IV). IEEE, pp. 91-95, 2018.

[13] K. Okada, M. Yoshida, T. Itoh, T. Czauderna and K. Stephens, "VR system for spatio-temporal visualization of tweet data and support of map exploration," In: Multimedia Tools and Applications. Springer, pp. 1-20, 2019.

[14] A. Moran, V. Gadepally, M. Hubbell, and J. Kepner, "Improving Big Data visual analytics with interactive virtual reality," In Proc. 2015 IEEE High Performance Extreme Computing Conference (HPEC). IEEE, pp. 1-6, 2015.

[15] B. Sun and W. Xu, "An Immersive Visual Analytics Platform for Multidimensional Dataset," Available from https://www.osti.gov/servlets/purl/1542284 2019.12.04

[16] A. Bierbaum et al, "VR Juggler: A virtual platform for virtual reality application development," In Proc. IEEE Virtual Reality. IEEE, pp. 89-96, 2001.

[17] F. Fittkau, A. Krause, and W. Hasselbring. "Exploring software cities in virtual reality," In Proc. IEEE 3rd Working Conf. on Software Visualization (VISSOFT). IEEE, pp. 130-134, 2015.

[18] A. Teyseyre and M. Campo, "An overview of 3D software visualization," In IEEE Trans. on Visualization and Computer Graphics. IEEE, 15(1), pp. 87-105, 2009.

[19] A. Kashcha, "Software Galaxies," Available from http://github.com/anvaka/pm/ 2019.12.04

[20] J. Rilling and S. Mudur, "On the use of metaballs to visually map source code structures and analysis results onto 3d space," In Proc. 9th Work. Conf. on Reverse Engineering. IEEE, pp. 299-308, 2002.

[21] P. McIntosh, "X3D-UML: user-centered design, implementation and evaluation of 3D UML using X3D," Ph.D. dissertation, RMIT University, 2009.

[22] M. Krzywinski, "Schemaball: A New Spin on Database Visualization," In Dr. Dobb's: The World of Software Development, 2004.

[23] E. Olshannikova, A. Ometov, Y. Koucheryavy, and T. Olsson, "Visualizing Big Data with augmented and virtual reality: challenges and research agenda," J. of Big Data, 2(22), 2015.

[24] I. Herman, G. Melancon, and M. Scott Marshall, "Graph visualization and navigation in information visualization: A survey," In IEEE Transactions on visualization and computer graphics, 6(1), pp. 24-43, 2000.

[25] G. Di Battista, P. Eades, R. Tamassia, and I. Tollis, Graph drawing: algorithms for the visualization of graphs. Prentice Hall PTR, 1998.

[26] S. Card, "Information Visualization," In: Human Computer Interaction Handbook: Fundamentals, Evolving Technologies, and Emerging Applications (Ed: Jacko, J.), Third Edition. CRC Press, pp. 544-545, 2012.

[27] S. Palmer, Vision Science. MIT Press, Cambridge (USA) 1999, ISBN 978-0262161831.

[28] D. Norman, The Design of Everyday Things: Revised and Expanded Edition. Hachette UK, 2013.

[29] A. Van der Heijden, "Perception for selection, selection for action, and action for perception," Visual Cognition, 3(4), pp. 357-361, 1996.

[30] S. Card, J. Mackinlay, and B. Schneiderman (editors), Readings in Information Visualization: Using Vision to Think. Morgan Kaufman, 1999.

[31] [Online] Available from https://github.com/SouthbankSoftware/dbkoda-data 2019.12.04