In-Memory and Column Storage Changes IS Curriculum

Ronald Freeze
University of Arkansas
rfreeze@walton.uark.edu

Susan Bristow
University of Arkansas
sbristow@walton.uark.edu

Abstract

Random Access Memory (RAM) prices have been dropping precipitously. This has given rise to the possibility of keeping all data gathered in RAM rather than utilizing disk storage. This technological capability, along with benefits associated with a columnar storage database system, reduces the benefit of Relational Database Management Systems (RDBMS) and eliminates the need for Online Transactional Processing (OLTP) and Online Analytical Processing (OLAP) activities to remain separate. The RDBMS was a required data structure due to the need to separate the daily OLTP activities from the OLAP analysis of that data. In-memory processing allows both activities simultaneously. Data analysis can be done at the speed of data capture.

Relational databases are not the only option for organizations. In-Memory is emerging, and university curriculum needs to innovate and create skills associated with denormalization of existing database (legacy) systems to prepare for the next generation of data managers.

Keywords: In-Memory, Columnar storage, Denormalization

1. Introduction and literature review

In the 1950s the cost of computer random access memory (RAM) was exorbitant. Moore's Law, coined by Intel founder Gordon Moore, stipulated that the size of RAM would double every 18-24 months and resulted in the fact that the cost of RAM has significantly decreased (GlobalSecurity.org, 2022). With the decrease in cost and the increase in capacity, this makes way for more robust storage capability and increased processing efficiency. This phenomenon is known as In-Memory processing or In-Memory computing.

In-Memory computing houses data in RAM in place of disk hosted databases (Hewlett Packard Enterprise, 2022). In-Memory processing helps to eliminate obstacles when moving data between the processor and the primary memory. When the memory capacity of a system equals or exceeds the hard drive storage, this is the system specification point at which In-Memory can alleviate this challenge. Additionally, it provides real-time analytics of the data creating efficient reporting and better decision-making (Chi, et al., 2016, Plattner & Zeier, 2012). "With In-Memory computing and insert-only databases using row- and column-oriented storage, transactional and analytical processing can be unified. This development has the potential to change how enterprises work and finally offer the promise of real time computing" (Plattner & Zeier, 2012, p. xxxii). Division of operational database systems and data warehouse systems is no longer necessary. Of interest is the consequences for OLTP and OLAP applications (Loos, 2011).

Normalization, a concept that is linked to E. F. Codd, an IBM researcher, is used in relational databases to organize data into tables in a structured and non-repeating manner (Rouse & Vaughan, 2022). A necessary need when the cost of RAM was prohibitive. Now, with In-Memory processing, the need for normalization and a relational database format diminishes. A denormalized database, a database that has all data in one table or a minimum number of tables is being explored and put into practice. A denormalized database also makes way for columnar storage options. Since there is the possibility that all transactions, in a denormalized database, can reside in RAM, there exist the possibility that data analysis can be operated on the transactional system. If the analytical process can be accomplished within the same table as the transaction processing, the movement of data to a warehouse may have a reduced need. This would portend that the Entity Relationship Diagrams (ERD) taught and displayed for a RDBMS may give way to only Data models.

For enterprise systems, such as an Enterprise Resource Planning (ERP) system, In-Memory is creating faster processing and cost savings. "The most compelling case for in-memory technology may stem from the need of senior managers to view aggregated enterprise data in real-time" (Veague, 2017, para. 5). With In-Memory and ERP together, database traits of ACID (Atomicity, Consistency, Isolation, Durability) support transaction integrity. A hybrid approach by

URI: https://hdl.handle.net/10125/103403
978-0-9981331-6-4

maintaining In-Memory and traditional disc-based database, can be of benefit to aid with RAM reserves (Veague, 2017). Repeatedly needed queries can reside in RAM while data that is not as regularly needed would exist in a physical disc space. An ERP with full In-Memory should be reserved for high volume transactions. A hybrid configuration is best for an ERP system because of the need to have structured and unstructured data and because of costs associated with RAM (Veague, 2017).

The technological advance associated with a reduced cost of RAM and increased operational speed, creates the need to rethink how both Database and System Analysis and Design courses are taught in both Management Information Systems (MIS) and Computer Science (CS) courses. The remainder of the paper will begin by addressing In-Memory concepts with respect to In-Memory vs. Physical storage, Normalization vs. Denormalization of relational databases and the concept that Data Structure will become key. Curriculum impacts of these concepts will then be addressed in two sections related to a typical System Analysis and Design (SA&D) course as well as a typical Database Course. To address the research question of "what is the state of the industry?", organizational impact will be presented by the results of three interviews from different organizations with emphasis on these three concepts. We will conclude with a review of the future course of action associated with curriculum topics and concepts.

2. In-Memory computing

Relational databases are found to be inadequate in distributed processing involving very large numbers of servers and handling big data applications (Kanade & Gopal, 2013). When ERDs become fully normalized within a large database, there can be many tables associated with the data capture (more than 15-20 tables). Each table must have primary and foreign keys to describe the relationships between the tables. These keys become redundant data within the data structure but provided the means to reduce the physical storage used for the entire database. However, the big data applications, when accessing specific information, may require I/O associated with multiple normalized tables (5-10 tables). Even with the improved speed of memory, accessing and correlating data from multiple tables reduces the anticipated speed of analysis retrieval.

In-Memory performance for relational databases, when all tables reside In-Memory, can show improvement over accessing the same tables and reading the data needed into memory. However, there is now an emphasis on NoSQL database use, to take advantage of data that resides In-Memory, in contrast to the limitations of the relational database management systems (RDMS). The terminology here is schema free databases or denormalized databases. The topics associated with NoSQL database use contrasts the RDMS concepts which relies on ACID consistency and schema free NoSQL databases which are based on BASE (Basically Available, Soft-state, Eventually Consistent) (Kabakus & Kara, 2017).

2.1. In-Memory vs. physical storage

Standard databases store data on disc and the I/O operations are slow compared to those made in RAM. However, storing data only In-Memory does not fully achieve all the ACID components. To capitalize on the speed associated with having data always In-Memory, the Durability component of ACID is the major component that needs to be addressed (Babeanu & Ciobanu, 2015). Anytime the system is rebooted or restarted, the data currently In-Memory is typically lost and must be reloaded. The Durability component must address where the data must be loaded from and how that backup system is kept current with the In-Memory data. In some cases, such as SAS VIYA 3.5, the data stored in a physical storage table is the "master" data representation. In fact, for In-Memory data tables, having a physical table as a backup is an adequate method of data storage and backup.

With a potential system reboot addressed in a design mode, how physical tables can be updated along with reading them into an In-Memory table can be addressed on an interval time frequency for maximum system utilization. With this and similar precautions, In-Memory databases can provide a significant performance advantage over disk-oriented databases since they avoid disk I/O (Lahiri, Neimat, & Folkman, 2013). In this fashion, "In-Memory Data Management (IMDM) may replace current data warehousing concepts completely". Multidimensional and relational OLAP implementations and corresponding data models may thus become less important (Piller & Hagedorn, 2011).

2.2. Normalization vs. denormalization

Normalization is a database design technique that reduces data redundancy and eliminates undesirable characteristics like Insertion, Update and Deletion Anomalies. Normalization rules divide larger tables into smaller tables and links them using relationships. The purpose of Normalization in SQL is to eliminate redundant (repetitive) data and ensure data is stored logically. (Peterson, 2022)

The process of normalizing a database created a physical memory saving benefit for the early use of capturing data. The main driving force was the cost of physical memory associated with storing data. A simple three table schema (Sales Transaction – Product – Customer) capturing sales transactions demonstrates the physical memory savings associated with the use of the schema (Figure 1). This schema allows the elimination of duplicating all the Customer information (7 variables) and Product information (4 variables) for the storage in physical memory associated with a customer buying products on multiple days and multiple locations. The physical memory savings can be illustrated based on the number of variables associated with 100 independent sales to a single customer.

If all the customer information and product information were stored in a single table, each of the 7 variables associated with the customer and the 4 variables associated with the product sold would be duplicated with the use of a single table. This would be an additional 1,100 cells of data storage ((7+4) * 100) minus the duplication of the primary keys used to link the tables. However, for the Product and Customer tables, there is only a single primary key cell used. The Sales Transaction table would have 100 customer number duplications. Similarly, there would be 100 product numbers in the single table that would no longer be needed. The physical memory would save 898 cells (1,100 – 202) by normalizing the database and using the Figure 1 schema. This represents, for this example, a savings of 18.4% of the cells needed to store the data from the transaction. However, this analysis only represents the savings in terms of cells and does not consider the variability of cell size that is associated with data types (text vs numeric) and the physical memory necessary to accommodate these differences.

While denormalization carries a cost associated with physical memory use, the dollar value of physical memory has shrunk to be negligible in current systems. Additionally, there are operational benefits associated with denormalization. Since all transactions and their details are associated in a single table (Figure 2), the need to update a specific table is no longer needed. For instance, when a customer moves to a new address, the sales transaction starts with a simple lookup for the prior customer address in the sales transaction table. A verification step is initiated to ensure the address is still accurate. If the data is inaccurate, the current transaction can be updated to reflect the new address.

There is no updating required of a Customer table. Also, there is no need to store a customer's prior address as this can be found in the historical sales transaction table.

There is also a need to reconsider the function of the primary key. In a relational schema, the primary keys, or composite primary keys, were needed to insure data consistency within the database. An emphasis on the primary key was the need to "uniquely" identify the record associated with each entry of a table. Since a table would now be denormalized and contain all the information of the sales transaction at the time of the transaction, is any primary key necessary? Several approaches can be taken, but with this three-table schema (Figure 1), the need to uniquely identify each record is needed. Note that in the denormalized table (Figure 2), the Date/Time variable has been shown as a primary key with a question mark. When evaluating the number of transactions that occur at large retailers, a composite key should be considered. The suggestion to ensure that a sale is uniquely identified, location should be considered along with the Date/Time variable. Since large retailers will have multiple stores around the world and each store will have multiple checkout lanes, the Date/Time variable should be coupled with a store location (simply indicated by the Store #) and the register at which the sales transaction occurred (indicated by the Register variable).

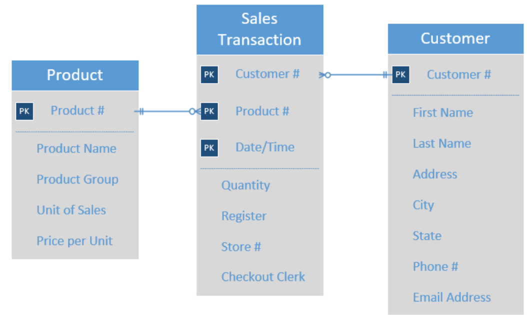

Figure 1. Three table schema

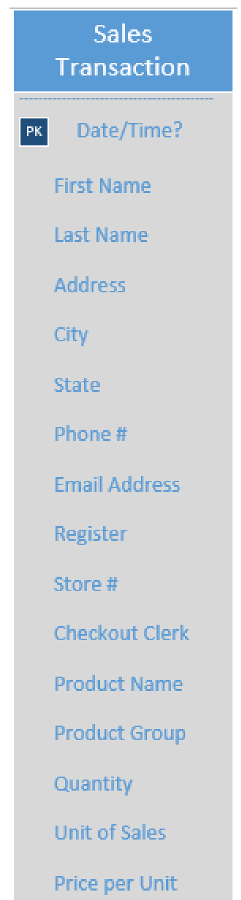

Figure 2. Denormalized table

2.3. Data structure is key – column vs. row orientation

Both the use of In-Memory computing and a move toward denormalization has opened the data storage methods to include a column-oriented storage method as opposed to a row-oriented storage method for databases. Both storage methods must be able to perform the standard database activities of inserting and updating of data. While there have been studies provided on the different benefits, there is little solid evidence that points to one storage method being more beneficial than the other. The results of one review of a hybrid CPU/GPU system investigating break even points indicated that row orientation performs better for the CPU while column oriented performs better in the GPU system for inserts. Column store performed better in both systems for updates. The importance of the data structure and projections were noted as important to their results (Arefyeva, Broneske, Pinnecke, Bhatnagar, & Saake, 2017). Another article reviewing column versus row-oriented data structures indicated that changes must be made to both the storage layer and the query executor to fully obtain the benefits of a column-oriented approach (Abadi, Madden, & Hachem, 2008).

More research is needed to develop exactly how to address Column vs Row storage techniques. Early research indicates that column-stores do not see a performance degradation when storing extremely wide tables (Dwivedi, Lamba, & Shukla, 2012). This is an important impact if denormalization occurs. Tables will be very wide and may become a new barrier to efficiency. It has also been noted that row stores can use some of the same techniques as column stores for optimization (Halverson, Beckmann, Haughton, & Dewitt, 2006). Conceptual data models along with comparisons of a logical data model and the star schema ERD have been investigated along with parameters provided in processing simulations (Kamal & Gupta, 2014).

3. Curriculum impacts

Data storage and use typically involve several course sequences the core of which reside in two classes. These two classes are generally named System Analysis and Design (SA&D) and some variation of Database. While the technical advances will impact more than just these two courses, these two comprise the core of how systems are designed and therefore will be most impacted by the advent of In-Memory computing. Denormalization and columnar-oriented data storage are coupled and necessary to include with the In-Memory computing addition to both MIS and CS curriculum.

3.1. Systems analysis and design course

A SA&D course is traditionally a core course in the MIS or related academic programs. Students learn about project management methodologies and techniques and have experiential learning with the concepts. Typically, the SA&D course incorporates a project that has a need for a database. Before the database is even realized, gathering requirements, and analyzing the requirements must take place. Part of the requirements comprise data the project needs. This data is closely tied to the database development. Students receive exposure to Entity-Relationship Diagrams (ERD), Data-Flow Diagrams (DFD), and Normalization as part of the database module within the SA&D class. Usually, the project will be completed until the database is to be implemented. Depending upon the order of course sequencing, the database course will then take the database concepts from the SA&D overview and theory and move towards deeper knowledge and hands-on implementation of a database. In-Memory concepts as well as columnar versus row architecture will need to be presented and discussed as part of this segment of the course as they are new concepts and realities that companies are facing and experiencing. These additional topics need to be implemented at the potential loss of time spent on normalization.

3.2. Database course

Database courses exist in Business, Computer Science, and Engineering Schools. The textbook names used for these courses vary as much as the course titles and include: Database Modeling and Design, Database Design, and Database System Concepts (Silberschatz, Korth, & Sudarshan, 2019; Watt, 2014; Teorey, Lightstone, Nadequ, & Jagadish, 2011). There are many topics that are covered in these courses, but we will focus on four of these topics that will need the most revision. Those four topics are: 1) the entity relationship data (ERD) model, 2) Normalization, 3) OLAP and 4) Structure Query Language (SQL).

An accurate ERD model is a pictorial representation of an operational database. The ERD provides a list of tables and their relationships to each other as the example in Figure 1. Complex ERDs can encompass more than 15 different tables. Planning of a database design will result in a theoretical ERD that can accurately be constructed in a database system. Each table listed in an ERD will include the table name, table variables, and variable data types. In addition, appropriate variables will be listed as primary keys, composite keys, or foreign keys. Attached to these keys will be relationship lines connecting tables and indicate the type of relationship. These relationships typically are represented as 0-1, 1-1, or 1-M.

Normalization is a primary topic in database courses. Recall that normalization rules divide larger tables into smaller tables and links them using relationships. The different normalization forms teach how to recognize the situations where a table can be divided to eliminate redundant storage of data. There are three typically taught normal forms (NF) referred to as First NF, Second NF, and Third NF. These forms are numbered such that the easiest form to achieve is the First NF. Achieving the Second NF requires that all tables be in First NF prior to the creation of the Second NF ERD. Each step requires the addition of another table with the appropriate relational primary keys. There are also four other NFs that are frequently taught (Boyce-Codd, Fourth NF, Fifth NF, and Sixth NF). A parallel topic that is taught with normalization is Functional Dependency Theory. This theory helps explain and identify relationships or dependencies between two attributes. The ability to recognize dependencies assist in the creation of an ERD with the appropriate number of tables to achieve the highest normal form possible.

Once an ERD has been operationalized, organizations want to use the data to analyze their operations to reduce costs, improve sales and identify opportunities. The term taught in database courses is Online Analytical Processing (OLAP). Efficient analysis of the organizational processes required organizations to move the data from the Online Transaction Processing system (OLTP) to another physical storage system referred to as a Data Warehouse. This was done since the In-Memory resources were not sufficiently large enough to run analysis on the OLTP system. This business need created another set of operations requiring the rejoining of the table data into multiple data warehouses focused on specific aspects of the organizational needs (operations, sales, purchasing, etc.), In many cases, the denormalized data was brought to the data warehouse in either singular or an aggregated format. The joining of tables was accomplished through one of many relational query languages. The most prominent language is referred to as Structured Query Language (SQL).

SQL is the basis for accessing much of the data in a database. There are four main types of language statements or commands in a SQL server: Data Manipulation Language (DML), Data Definition Language (DDL), Data Control Language (DCL), and Transaction Control Language (TCL). While these languages and commands were created and refined in an ERD based environment, all these commands reference actions associated with the data that is captured in a database whether there is one table or many tables. While the basic SQL statements are still relevant in a single table environment, there may be needs to modify some statements to utilize more effectively an In-Memory column-oriented data structure.

4. Organizational impact study

The new phenomena of In-Memory computing, normalization versus denormalization, and columnar storage, necessitates organizational input to assist in the academic direction of growing the skillset of the upcoming workforce. Providing a workforce with the knowledge and experience of the innovative technology, assists in supplying transitional support to organizations. To answer the paper's research question, "what is the state of the industry?", three companies were interviewed to obtain information regarding the current usage of In-Memory computing, normalization versus denormalization, and columnar data storage. The information gathered will be used to direct current and future curriculum.

All three companies interviewed are located in the midsouth region of the United States. The first company, Company A, is a large privately owned firm that manufactures packaging for consumer product goods (CPG). The second company, Company B, is a private not-for-profit mutual insurance company. The company offers health related insurance plans. The third company interviewed, Company C, is a Fortune 100 multi-national company, specializing in providing protein to consumers, foodservice customers, and businesses. The Interview questions were open ended and can be found in the Appendix.

4.1. Company A

Company A currently does not use any In-Memory computing, denormalization, or columnar storage. However, the company is moving from an SAP ECC5 system to the new SAP S/4HANA system. The S/4HANA system provides the opportunity to take advantage of the benefits associated with In-Memory computing, denormalization, and columnar storage. As the next system is implemented and at the point of "go live", the company anticipates that all three elements will be in place for the organizational processes.

Competency development with current personnel for In-Memory computing, denormalization, and columnar storage was inquired with the company. There are several personnel who are experts in normalization versus denormalization. Training is needed to assist with In-Memory computing, but aptitude is there for learning. Regarding column-oriented data storage, much research has been done, though a high level of practical application experience has not been obtained.

The implementation of S/4HANA is the main catalyst for moving towards denormalization and columnar-oriented data storage for Company A. With In-Memory computing the company anticipates an increase in its effectiveness and efficiency for the analysis of their process data. It is a significant goal for the company and is a major reason for implementing the new system.

To supplement the overall skill level, Company A views hiring new employees as a high priority. It wants to find talent to assist with working in the new In-Memory, denormalized, and columnar-oriented system. Senior and junior level personnel are needed and is most vital to meet their business needs. The hiring of consultants will also be considered based on the success of their full-time hiring of employees. Having recent graduates with In-Memory computing, denormalization, and columnar-oriented data structure knowledge, would provide the necessary increase in their workforce skillset for Company A.

4.2. Company B

Company B is new to the In-Memory computing, denormalization, and columnar storage technology. The company is exploring and moving towards In-Memory computing with its Oracle system and hopes this also includes Microsoft technology as well. The company currently utilizes an In-Memory grid that they are still working to maximize its benefits. By focusing on In-Memory, the company believes it can achieve high data efficiency. Additionally, with the use of Splunk logs and In-Memory computing, the company can learn from feedback through operational analytics. Database personnel are in training to develop further their In-Memory skills.

For database normalization, Company B informs it is a significant necessity for reporting and accuracy purposes especially with customers' information. However, Company B is denormalizing a portion of its database to help with search and read queries in the Operational Data Store (ODS) which is used for member eligibility. The company feels confident in the database personnel's normalization versus denormalization competency percentage, a high forty percent. Company B will continue to strive to have a better understanding of what it means to have normalization and denormalization as it desires to enable data science. Additionally, the concept of Durability in the ACID equations, is being addressed by storing customer information in multiple locations across the country to reduce latency of inquiries.

Row-oriented is still the current data storage methodology with some columnar-oriented data structures in their MongoDB database. Relational databases dominate and therefore the row orientation continues. Columnar-oriented data storage is a finite skillset for a small percentage of engineers (5% of 800) within Company B. However, Company B has developed a table that includes one customer for each row. A very wide column structure that they are still exploring the benefits for analysis.

Company B desires to hire new employees as opposed to upskilling current employees in the areas of In-Memory computing, denormalization, and columnar-oriented data storage. It is a high priority. Many of the current employees are in mid to late career and the desire or the need to add another skillset is not optimal. Universities are well positioned to aid in providing potential employees with this skillset by incorporating the concepts in core classes.

4.3. Company C

Company C, the largest of the three companies interviewed, has a relatively large footprint of its data In-Memory. The estimate is that approximately 40% of its 11.5 Terabytes of data currently resides In-Memory. Company C has multiple database platforms although their main platform is an SAP S/4HANA installation. They currently have a small team that is focused on the use of In-Memory to improve the performance and response of their data inquiries. This team focuses on tool availability to maximize the use of In-Memory computing.

The emphasis on the normalization of data and the use of columnar oriented data structures is largely driven by their vendor. In the day-to-day operations, the status of the normalization of their data tables is not discussed or has much of a concern with respect to decision making. However, the use of columnar oriented data structures is used, but predominantly in the storage of text. Additionally, changes to the data structure are predominantly driven by the data analytics portion of the organization. This includes how data analytics is addressed with respect to In Memory computing. Most notable is the lack of a data modeler footprint within the organization.

4.4. Interview summary

The initial investigation into what organizations are using with respect to In-Memory computing, Normalization vs. Denormalization of data tables and Column vs. Row oriented data structures have supported the need to address the content of SA&D and Database courses. The summary of these interviews has been provided in Table 1. Interview summary.

Both Companies B & C are currently making heavy use of In-Memory computing. This eliminates or reduces the need for accessing the physical storage of their data and, at the very least, necessitates a shift in how the physical storage of data interacts with the storage of data In-Memory. Company A is planning on the usage of In-Memory data as an expected future need.

The normalizing of data is already not a concern for the largest of the companies interviewed Company C. Company B still uses normalized tables and believes there is still importance but have begun experimentation into having only a single row for each customer and all their interactions with the company. Company A is expecting to substantially reduce the number of tables when they "Go Live" at their new installation.

Table 1. Interview summary

	Company A	Company B	Company C
In-Memory	No In-Memory usage Expected future need	In-Memory Grid Storage across Country	Heavy emphasis Analytic result focused
Normalization	Substantial skill level Denormalizing is future skillset	Table with 1 row/customer Normalized Tables in Use Denormalizing some tables	Driven by vendors Smaller emphasis by Analytic team
Column Oriented	No current usage Expected future need	Small internal skill set Row orientation dominates	Driven by vendor Some column storage

Finally, the column vs. row data structure is the least clear with respect to advantages for the companies. They have begun to acquire some skills but do not have a substantial skillset or a clear advantage that the column data structure orientation would accrue to the organization. This still speaks to the need to at least include this discussion in the SA&D and Database curriculum.

5. Future course of action

This article has identified the core need for university programs from both Management Information Systems, Computer Science, and Engineering university departments to begin planning for the future needs of industry with respect to data organization, storage, and technology. The initial impact on courses has been explained in the two courses of SA&D and Database. The primary impact in these two courses is focused on ERD diagrams and the adjusted relationship or merging of OLAP and OLTP activities. There are many tangential impacts that need to be explored while adjusting the curriculum.

The primary topics that would be impacted next is the technical infrastructure associated with an organization's use of the housing of data. Historically, organizations have moved data from their processing of data to a data warehouse or multiple data marts. The interviews of the three companies have shown there is substantially less movement of data to a data warehouse. This movement of data has been reduced and with the advent of In-Memory computing less important as organizations can do and are doing analytics on their transaction system without loss of performance. However, this move will shift the focus of organizational storage of historical data. Time frames for the analysis of data will impact how much data is readily available (In-Memory) and how much data is stored for historical purposes. This question focuses directly on how long data has value.

A secondary topic would be how the focus on movement of data from physical memory to In-Memory use becomes valued. The movement of data may become unidirectional from the transactional In-Memory capture of data to a long-term storage of historical data. Again, this raises the question of how fast data value becomes obsolete. Answering these questions has an impact on the technological landscape of organizations.

6. Appendix: interview questions

6.1. Platform

1. What database platforms do you use in your organization?
2. Do these platforms have an option for In-Memory computing?
3. Do these platforms have an option for columnar oriented storage?
4. Are your current data storage methods row or column oriented?
5. Are you investigating or planning on investigating platforms the have these options?

6.2. In-Memory

6. To what extent do you use In-Memory computing?
7. What percentage of your data is always In-Memory?

6.3. Normalization and denormalization

8. How important is the level of normalization in your databases?
9. To what extent are you denormalizing your databases?

6.4. Competency development

10. What competency percentage would you say is the skill set of your database personnel with respect to normalization vs. denormalization in your database schema?

11. What competency percentage would you say is the skill set of your database personnel with respect to In-Memory computing?

12. What competency percentage would you say is the skill set of your database personnel with respect to column-oriented data storage?

6.5. IT support of organizational goals

13. Is there a move in your organization away from relational databases toward more of a denormalized data capture?

14. Is there a move in your organization where analytics is being performed on an In-Memory transactional system?

15. Is there a move in your organization toward a column-oriented data storage approach as opposed to a row-oriented data storage approach?

16. With respect to any shortfalls, what combination of activities are you planning to close the gap: 1) new hires – graduate vs undergraduate vs industry, 2) internal & external training of existing personnel, 3) hiring of consultants and/or 4) others?

7. References

Abadi, D. F., Madden, S. R., & Hachem, N. (2008). Column-stores vs. row-stores: how different are they really? *Proceedings of the 2008 ACM SIGMOD International conference on Management of data, (pp. 967-980).*

Arefyeva, I., Broneske, D., Pinnecke, M., Bhatnagar, M., & Saake, G. (2017). Column vs. Row Stores for Data Manipulation in Hardware Oblivious CPU/GPU Database Systems. *Grundlagen von Datenbanken*, 24-29.

Babeanu, R., & Ciobanu, M. (2015). In-memory databases and innovations in Business Intelligence. *Database Systems Journal*, *6*(1), 59-67.

Chi, P., Li, S., Xu, C., Zhang, T., Zhao, J., Yongpan, L., Wang, Y., & Xie, Y. (2016, 10). PRIME: a novel processing in-memory architecture for neural network computation in ReRAM-based main memory. *Computer architecture news, 44*(3), 27-39. Retrieved 05 27, 2022, from https://dl.acm.org/doi/10.1145/3007787.3001140

Dwivedi, A. K., Lamba, C. S., & Shukla, S. (2012). Performance Analyss of Column Oriented Database versus Row Oriented Database. *International Journal of Computer Applications, 50*(14), 31-34.

GlobalSecurity.org. (2022, April 5). *Moore's Law*. Retrieved from https://www.globalsecurity.org/military/intro/moores-law.html

Halverson, A., Beckmann, J. L., Haughton, J. F., & Dewitt, D. J. (2006). *A comparison of c-store and row-store in common framework.* University of Wisconsin-Madison, Computer Science.

Hewlett Packard Enterprises. (2022, September 1). *In-Memory computing.* https://www.hpe.com/us/en/what-is/in-memory-computing.html.

Kabakus, A. T., & Kara, R. (2017). A performance evaluation of in-memory databases. *Journal of King Saud University - Computer and Information Sciences*, 29, 520-525.

Kamal, A., & Gupta, S. C. (2014). Query based performance analysis of row and column storage data warehouse. *9th International Conference on Industrial and Information Systems* (pp. 1-6). IEEE.

Kanade, A. S., & Gopal, A. (2013). Choosing right database system Row or column-store. *2013 International Conference on Information Communication and Embedded Systems* (pp. 16-20). IEEE.

Lahiri, T., Neimat, M.-a., & Folkman, S. (2013, June). Oracle Times Ten: An In-Memory Database for Enterprise Applications. *Data Engineering, 36*(2), 6-13.

Loos, P. (2011). In-memory Databases in Business Information Systems. *Business & Information Systems Engineering, 3*(6), 389-396.

Peterson, R. (2022, April 16). *What is Normalization in DBMS (SQL)?* Retrieved May 23, 2022, from Guru99: https://www.guru99.com/database-normalization.html

Piller, G., & Hagedorn, J. (2011). Business Benefits And Application Capabilities Enabled By In-Memory Data Management. *Proceedings zur tagung Innovative Unternehmensanwendungen mit In-Memory Data management*, (pp. 45-56).

Plattner, H., & Zeier, A. (2012). *In-Memory Data Management: Technology and Applications* (2nd ed.). Heidelberg, Germany: Springer. doi:10.1007/978-3-642-29575-1

Rouse, M. & Vaughan, J. (2019). *Database normalization.* https://www.techtarget.com/searchdatamanagement/definition/normalization.

Silberschatz, A., Korth, H. F., & Sudarshan, S. (2019). *Database System Concepts* (7th ed.). McGraw-Hill.

Teorey, T., Lightstone, S., Nadequ, T., & Jagadish, H. V. (2011). *Database Modeling and Design - Logical Design* (5th ed.). Elsevier. Retrieved May 23, 2022, from https://www.elsevier.com/books/database-modeling-and-design/teorey/978-0-12-382020-4

Veague, R. (2017). *In-memory – A new approach to ERP*. Retrieved September 1, 2022, from https://www.datacenterknowledge.com/industry-perspectives/memory-new-approach-erp.

Watt, A. (2014). *Database Design*. Open Textbook Library. Retrieved May 23, 2022, from https://open.umn.edu/opentextbooks/textbooks/354