Final Research Paper

Database Architecture

Submitted by

Srireshmi Nittala

ICT 4010 Enterprise Architecture

Faculty: Edward G. Lewis

March 7, 2018

Table Of Contents

Background ……. 2

Major Enterprise Architecture issues…..……..……..…….. 6

Issues in delivering a robust data security plan……..……...….... 8

Islands of data increase……...……...…...…….9

Growing complexity of data landscape…….…….……..10

Limitations of data recovery…….…….…..…..11

Lack of experienced people…….……...……....12

Decentralized data management…..…….…….…….……12

Poor Plan …….…….……..…….……...13

Ignoring the value of normalization and stored procedure……...……....……..16

Conclusion…….……..…..……19

Recommendations…….……..…....……23

Appendices…….……...……..…..……..24

Endnotes…….....……..……...……..27

Bibliography…..……30

Background

Inception of Database architecture was in the year 1971 which was initially named as Database Task Architecture. Database Architecture (DB) is the most sophisticated and critical architecture in the modern enterprise architecture world[1]. DB architecture uses specific programming languages to design softwares for organizations or businesses and it focuses on implementation, design, development and maintenance of organizational data. An Enterprise Architect alone cannot design or suggest a blueprint of Database Architecture and will need the support of a Database Architect. A database architect handles the development and implementation of DB software to meet the requirements of a given project. An efficient Database Architecture in today's technologically oriented world is the resultant of immense research and development of Database Management Systems[2].

In the early versions of Database architectures, design attributes like scalability and reliability were not present. Later on, many DBMS experts contributed their R&D work and pioneered many DB design techniques for consistent scalability and reliability. It is very noticeable that the data aspect of an organization is majorly associated with different DB frameworks, its implementation, and usage. There are six major point of views to be considered for the development and maintenance of a DB framework[3].

First one is the planner point of view where the planner of the Database architecture reviews all the significant aggregated groups that build the domain of the business model. The second one is business owner’s view where the business holder focuses on the nature of business and is concerned with the tangible aspects that constitute the business environment. To

generalize this, both the planner and the business holder are responsible in defining major business needs. The third point of view is the architect's where the architect is responsible for the structural elements of Database architecture[4]. Architect also designs the conceptual-entity relationship model and further codifies it into an enterprise suitable language. This gives a suitable structure to the Database system for information processing and usage.

Fourth point of view is the Designer's view. In this view, the designer completely utilizes the architect's DB blueprint to form a substructure and define the data management technology to solve specific data related problems. Fifth one is the builder's point of view, where the builder decides the data storage locations and medium of storage[5]. Considering data storage is very important as it helps in risk management. Last but not the least, functioning of systems is the sixth point of view, which defines the consistency of physical databases and functioning of database architecture in holistic manner. If an organization considers these point of views while designing, developing and organizing the DB framework, it will help in optimizing their work. Database architecture can be two-tier or three-tiered depending on how the users are connected to the database systems[6]. Users can either directly connect to the database systems or the user requests can be received by an intermediary layer which are interpreted and sent to databases.

Two-Tier Architecture

In a two-tier framework, the application program directly communicates with the database. There is no presence of user interface or user involved database thread communication. Consider the example of a school interface, where the front end application displays the student reports and their association with distinct courses. In this example, the front end application

directly interacts with the database and fetches the required data and it is evident that in two-tier architecture, no specific user inputs or query processing is required[7]. The typical two-tier architecture will have an application layer and a database layer.

If we consider a ticketing system, another example in a two-tier architecture, the ticket requests are collected and stacked as a queue in the application layer[8]. The requests are sent to database layer for processing through a First Come First Serve (FCFS) approach, and the processed requests are sent back to the database layer. The following figure is the diagrammatic representation of Two-Tier architecture. (Refer Appendices)

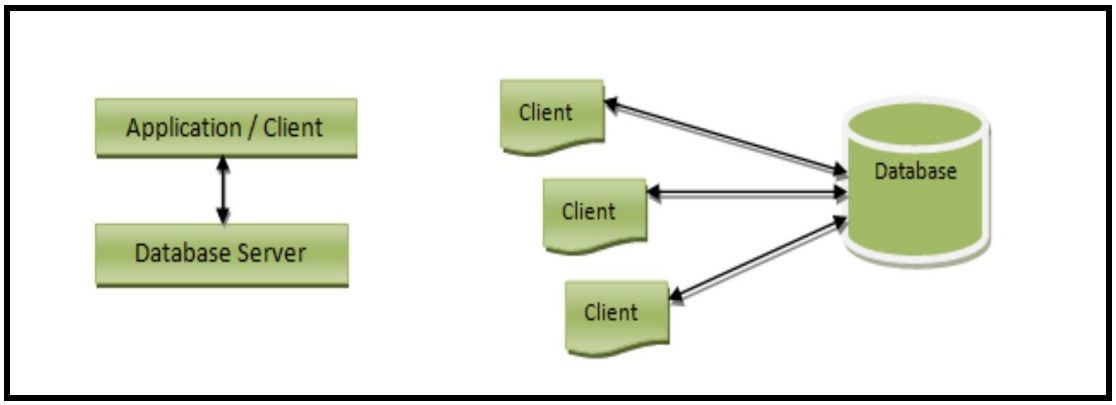

Figure 1: Two-Tier DB Architecture.

The two-tier architecture is easy to understand and develop as it directly interacts with the databases and also it is a reliable option for less number of users as it quickly retrieves the requested data[9]. On the other hand, if there are a large number of users, the two-tier architecture is not a good option as it consumes more time to process the requests. The two-tier architecture cannot handle multiple threads at a time and is also considered as an expensive framework[10]. The two-tier architecture is often suggested for few restricted businesses and startup organizations.

Three-Tier Architecture

The American National Standards Institute (ANSI) has recognized that the three-tier architecture is the most extensively used Database Architecture in today's organizational culture. Unlike the two-tier architecture, three-tier has three layers including presentation layer, application layer, and database layer[11]. In the presentation layer, the user does not have any knowledge about working of the internal database. The user indirectly interacts with the database, as he has all the data in front of him[12]. In simple terms, presentation layer can be imagined as a registration form inputting the user details and followed by a submit button where the user details are taken and sent to the next layer[13]. Figure 2. below, depicts the typical diagrammatic representation of a three-tier architecture.

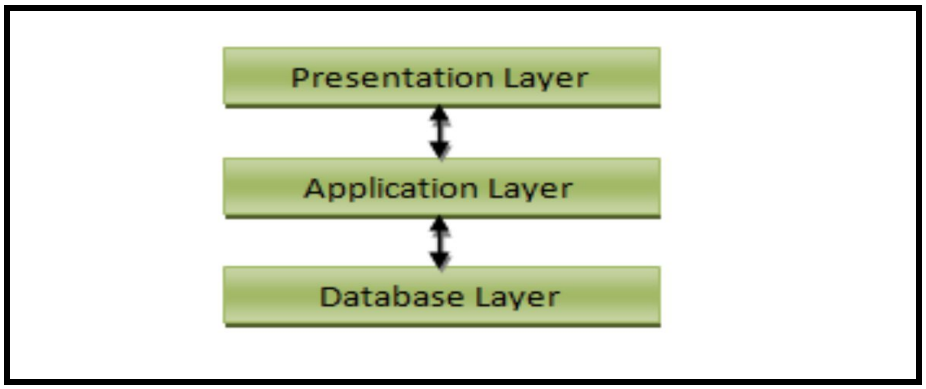

Figure 2: Three-Tier DB Architecture.

The second layer in a three-tier architecture is the Application layer. Application layer is responsible for saving and retrieving the details that the user has entered. This layer is an underlying program which performs tasks like calculations, validations and manipulation of data.

If this layer finds any ambiguity in entered data, it will not process it to the next layer and it bounces back to the presentation layer with a failure message.

Database layer is the third layer in three-tier architecture where all the user data is located. According to the type of schema, the data is present in the form of tables and mappings. When an action is started in the presentation layer, the data that is received will be saved in the respective tables of a database. Similarly, if the user wants to retrieve the entered details from the database, he uses web browser and sends request to database architecture where the request hits the database as a query and renders the output to the user. Though the architectural concepts may seem simple, there are some evolving database issues that need to be thoroughly understood.

Major Database Architecture Issues

- Limitations in scaling the database.
- Issues in delivering a robust Data security plan.
- Islands of data increase.
- Growing complexity of data landscape.
- Limitations of Data Recovery.
- Lack of experienced people.
- Decentralized Data Management.
- Poor designing and Planning of DB architecture.
- Inadequate testing.
- Lack of proper documentation.
- Poor naming convention standards.

- Ignoring the value of Normalization and stored procedure usage.

Analysis of the Major Enterprise Architecture Issues

- **Limitations in scaling the database.**

The ability to scale up or expand a database to let it hold increasing amounts of data without disturbing the performance is known as database scalability[14]. In today's contemporary world, maintaining islands of data is as challenging as designing an effective database architecture. Scaling up the database servers have many limitations depending on the architecture. One of the major problem of scaling up the database is the increased pressure on database servers, computers and other connected resources.

Even the most reliable databases like Oracle, RDBMS and MySQL are designed to work on a single server base and as the volume of data grows the organizations face many performance and functional infrastructure issues[15]. When an organization notices these issues they immediately follow up and typically first try to scale up the servers which is an affordable option. In later stages, they tend to understand and change the whole technology due to many surrounding issues. Due to this, when many organizations today experience performance or functional problems, they consider changing the database technologies in a holistic way rather than temporary scaling up solutions[16].

As technology evolved over the years, alternate cloud-based solutions are now available to overcome the scalability issue. For cloud-based technologies, the major problem is reliability and security. Though cloud-based storage seems like a good solution, organizations are very

reluctant to adapt and follow as they are concerned with security, privacy and reliability of their data. Another approach that ANSI approved is a distributed computing solution, where new servers can be added to the database clusters at any point of time and the systems are supposed to rebalance the data load and manage the new resources[17]. Distributed computing is still on its way to improve its identity and due to the complexity involved small organizations find it difficult to implement. Also, distributed computing is not cost effective which is another reason why many companies are not ready to try the scalable cloud-based approach. Database scaling is represented diagrammatically in appendices as (Figure 3: Database scaling)

- **Issues in delivering a robust Data security plan.**

For any given IT company, databases are the hidden pillars and it is a common practice that an organization's critical, public, and general data are stored in databases. Historically there were many incidents that targeted high-profile business information and caused data breaches. Typically a data breach for a company will cost around 4.5 million dollars including the loss of data, reputation, goodwill, and mental health[18]. Information Technology and tools are accessible to anyone today and every year 28% of breaches occur because of malicious intentions. Data breaches have many faces like, malware, malicious intentions, lack of experience in handling data, lack of security, etc.

With the increasing islands of data, Big-Data which is a new trend had emerged and unfortunately many of the tools associated with this technology and data analytics are open source[19]. Few of these tools were designed with single goal accomplishment without considering security aspect. Though the future of big data seems very promising and bright, the security

aspect of this technology is still a complicated issue. Many big data experts reveal that this technology is going to be more sophisticated in coming years which makes the security aspect more complex[20]. Finding the security specialist for big data is a burden because there are very limited number of experts in this field. To address this issue, the ANSI group need to take a strong initiative on security intelligence and set regulations according to it.

- **Islands of data increase.**

Factually, the data is the useful resource to derive required information for a company. Data does not specifically help the organization if it is not transformed into knowledgeable information[21]. Over several years, with the increasing technologies, algorithms, and powerful computers the data collection, storage and manipulation has enormously increased and it reached a point where organizations are proliferating their data without using it. According to economic times, for the past 5 years, there has been an increase in data by 40 fold out of which only 18% of data is being used for drawing important insights[22].

Increasing demand for Chief Information Officers, Data Scientists, Data Analysts, Executive Data Programmer roles clearly shows the data prominence and data increase and these employees are being more significant in today's industry to manage and direct these islands of data. Oracle, IBM, Microsoft have spent more than $50 billion alone on purchasing software firms for data management and analytics. It is also anticipated that there is over 10% of data increase every 6 months[23]. These statistics are overwhelming and unpleasant as they lead to data explosion at some point and also impact the organizations revenue.

However, there are many reasons for this data explosion and the most evident reason is availability of technology. This increasing technology is not only limited to IT organizations but also to customers and individuals across the globe that contribute to the data generation[24]. Though the increase of data and information every five years can be associated with the Moore's law, the world seems to be moving beyond Moore's law and data collection and technology is taken for granted which eventually will impact the way businesses are done using data. The diagramatic representation of growing data is mentioned in appendices as (Figure 5: Data growth)

- **Growing complexity of data landscape.**

Database is evolving rapidly and the database market has become very dynamic. Many organizations are finding it very difficult to choose a single solution from multiple options. Every vendor in the market is producing their own spin of technologies including but not limited to Columnar databases, Relational databases, Object-Oriented databases, In-memory databases, T-SQL, Nosql, and MongoDb[25]. These vendors are changing their platforms much faster than what small and large companies can handle and are suggesting those changes to the organizations with the excuse of migration to newer and better platforms.

This scenario is very confusing to organizations and they are unable to process which kind of platform is usable for their business needs. Even after selecting a solution, the flood of emerging technologies are pushing the companies towards unnecessary changes[26]. There are also other limitations like the hybrid platform providers, where the application vendor is different

from infrastructure and platform vendors. With their combination, different data management platforms are created in the same organization which leads to disintegration.

For example, some IT organizations use SQL, DB2, and Oracle servers with multiple versions which are provided by different vendors. This creates a major management overhead leading to disintegration[27]. This even impacts constraints like employee management, processing tool sets, consolidation and policy management. This scenario also restricts the centralization and creates friction in data management teams due to their varying resources and functions surrounding the infrastructure.

- **Limitations of Data Recovery.**

Though data recovery falls under the risk management category, people have been focusing on this aspect from the very beginning of electronic data collection. This is because of the increasing demand for data and the consequences that surround it. The complexity of recovering data is being increased day to day with the increase in data[28]. Generally it is less complex to recover the entire data set than just a specific data set. But when an organization losses a part of data set and wants to recover it, it is very difficult. Adding burden to this scenario, few factors such as 24x7 availability of data in the respective websites, increase in the rate of electronic transactions and having no conventional methods of backup are the disadvantages[29]. Due to these factors many database admins in the organizations do not allow delete statements to run in the databases which further leads to islands of data.

Data recovery is not only complex but also expensive. Even if the organization wants to recover a partial dataset, the solution providers extract full data which increases cost to

organizations. Statistically it is shown that for data recovery alone, every year organizations spend 28% of their revenue[30]. Whether it is a natural disaster, improper data administration, or other reasons, it costs overhead to the organizations to recover their data and sometimes it is impossible to retrieve the data. Simultaneously, regardless of whether the data is being used or not, admins do not let their employees delete the data or tables, which leads to data redundancy.

- **Lack of experienced people.**

Lack of experienced people in the industry has always been an issue in the Information Technology organizations. Some organizations do not focus and invest on money or time to give better training to their employees on technologies and some do not hire qualified people for the role. Especially in the Database management area, it is very hard to find qualified and experienced people that genuinely know the different loopholes in database management.

With the rapid advancements and emerging technologies in the database management it has become even more complex to find the talent[31]. This is leading to shortage of skilled people in this area and organizations are hiring employees who are less experienced and less aware of the newer trends. This situation can be undermined when organizations invest more time and money in training their employees and also by conducting effective recruiting and hiring processes which really identifies good expertise in the area.

- **Decentralized Data Management.**

Decentralized database management is the most popular methodology these days in the DBMS history and often it is described as the most beneficial type of data management. The

main problem in this type of database management is how the data is distributed. How does one design and manage decentralization with zettabytes of data? What is the decent degree of decentralization?

If a given organization cannot understand and address these questions before moving to decentralization then it will reflect unpredictable issues in the long run. Many data experts have approved that the distributed data management concepts are good to implement when the organization is globalized and also there is lack of standardization in decentralization[32]. It is also difficult as each business unit creates their own data ecosystem which separates them.

- **Poor designing and Planning of DB architecture.**

A structured and effective design and efficient planning is important while implementing the database architecture. A good database is built only when it is forethought with the business model in mind. Database architects need to be careful from day one when they start work on designing the database architecture. Higher priorities should be given to the needs of data and amounts of data that are going to hit the servers[33]. If the architects are not as attentive as they should be even in single aspect, it leads to a bad database architecture and impacts the organization severely.

Since database architecture is the major pillar of every business, essential care should be taken throughout the whole architecture. Also, organization should take utmost care in explaining clearly and coherently the nature of business and their business model. This leverages the database architects to understand more and to deliver better architecture to the organization. Unfortunately for many organizations, once they assign the contract to database architects, the

management does not inquire much about the production of architecture, considering that architects would deliver the best to them[34]. Organizations should invest more time on researching what kind of architecture is to be developed and align the technical steps involved in it.

This helps in better understanding of whole process instead of being out. The planning phase of architecture is often ignored by many organizations in the conception of just getting it done soon. Once the whole project done it is even tough to go back and fix some changes as it involves changing whole system which again costs the organization an overhead.

- **Inadequate testing**

The first thing to be checked when the business systems performance is low is the database systems. It is known that database is center to the business and if data fails it is most likely related to database systems. Most of the times the database architects follow the database framework template and perform minimal testing. According to Zachman's database architectural framework, a fine database architect should put testing as the paramount thing in project plan and the testing should be done for several days in various transactional conditions[35].

Due to time constraints and if the project does not move as planned, to quickly deliver the project, architects limit their testing and deliver the product. Failure to test can have immediate effects or have impacts in the long run. Lack of testing makes the database suffer from functionality problems to major performance issues. For example, if the user notices and complains that the save button in registration form does not work and the form cannot be saved in certain scenarios it is a functionality problem[36]. If in-depth process testing, user acceptance

testing and validation testing are made beforehand these mistakes would not occur. Performance issues usually emerge after a few runs on the data, workflow, and system processes.

Sometimes it is even hard to identify where the problem has occurred and the employee who is sitting and working on the database may not understand the issue. Database architecture is very sophisticated and fragile, and organizations should make sure that their architecture had proper testing before usage. A proper test plan will include all the possible test cases and all the probable failures[37]. A fine testing plan never eliminates all bugs, but at least it allows the organization to a point where most of the issues that are associated with the design are sorted out.

- **Lack of proper documentation.**

Documenting every project and every model is very important because it stands as the best reference point when someone needs a good understanding or someone lost track of what is happening. Any given project in Information Technology starts with step one and that is documentation. This is known as technical documentation and it is done by either the respective employee or business analyst.

The well written data documentation not only helps the individual but also the organization to go through and understand the processes. Sometimes companies allot non technical employees for this work and this is again leads to problem in technical communication and understanding. Some organizations do not set their own documentation standards and the technical writers across the team proceed in their own documentation template. This situation of having a variety of documentation standards sometimes leads to chaos and misunderstanding.

Also, the documentation should reflect the highlights, implementation, deployment and other key aspects but never the unnecessary details which can be limited internally.

- **Poor naming convention standards.**

Database management is done by few query languages using application softwares. Naming convention does not seem like an important aspect but it is very important for maintaining the standards of language and it reflects the excellence of the code. If the DB admin does not consider the naming convention methods, there is a probability that when an error occurs it leads to disorientation of his work.

By naming the columns, objects and other attributes by following the standard SQL guide style, it provides clarity not only to the particular coder but also to everyone. For example, if a person is working on T-SQL language and later changes to another platform, the documentation done while working in T-SQL, helps to analyze the historical approaches while working on the new platform.

- **Ignoring the value of Normalization and stored procedure usage.**

Normalization is, the set of methods that break down a table to their essential parts until each table shows the only one thing or one specific column where the other columns serve to contribute that one thing or column. The concept of normalization existed for more than 2 decades and it is the basis on which databases relational like SQL, T-SQL and Oracle databases are implemented.

In other words, Structured Query Language was created to work with normalized data structures. Normalization is considered as a different plot by database architects but it has been misunderstood. If there are fragments of data, a novice SQL developer tends to create different sets of values or results. Normalization concepts are used to better deal with fragments of data. For example, if we consider the FROM clause, the developer can take a table, or a set of data and can JOIN it to another table. Also, SQL allows you to merge as many tables of data as you like, to create a final set.

Normalization is the added advantage for this additive nature of SQL and this is very important in making development and performance easier. In addition, SQL also has the option of "indexes", which are most effective when you have larger datasets. For example, indexes help in to split the person's last name out of his first name where the SQL starts to break down the data even more quickly as the input criteria makes the searching pattern less searchable. Unfortunately, with the rise in new data management techniques, normalization is being ignored.

Database developers and Database managers should understand the importance of normalizing and incorporate the normalization methods in their daily query processes to deliver good results and ease of development[38]. There are multiple forms of normalization out of which the third, fourth and fifth normal forms are really useful. Once the developer figures out the normalization implementation the first time, it is quite easy to follow on a daily basis. However, in reality even the first form of normalization is not being implemented. The diagrammatic representation of Normalization is in appedices, refer to (Figure 4: Basic normalization representation)

Stored procedures are another gift for database management where they can be used possibly as a method of insulating the database layer from the end users. Though using stored procedures takes more effort in the beginning, they are time effective. With the usage of Stored procedures, database development will be cleaner and bridges the gap between database and functional programmers. Stored procedures also encourage the collaborative development between users[39].

The four main features that drive a developer to use stored procedures are Encapsulation, Maintainability, Security, and Performance. Through interfaces, stored procedures allow change in the characteristics of code without involvement of an external resource. If the developer wishes to change the SQL syntax, it is much easier to make changes with stored procedures. Even in terms of security, stored procedures provide two types of access to the system: granular and specific accesses. For example, if user-1 has ten stored procedures that updates table A, and if user-2 needs to update a particular column in the same table A, user-1 can set specific permissions which allows the user 2 to only access that particular table and column[40].

Stored procedures also deliver effective performance as they take very less time in processing the data. The procedures can be written in a way that they would not impact the performance of database systems and they would not go through all the data for results. SQL also provides the advantage of storing the plan and to reuse the set, to avoid manual compilation everytime[41]. On a positive note, for every upgraded version of SQL-Server, the performance factors are being constantly improved. As stored procedures take effort and time to code, the

developers do not show much interest and use generic T-SQL queries[42]. However, considering the performance benefits, stored procedures are worthy to implement.

Conclusion

Database architecture is the most sophisticated and important element in Enterprise Architecture. This paper focused on the major architectural elements of Database Management systems and also discussed the design principles and issues. Database architecture varies slightly for every business model but the basic architectural principles are the same for any data architecture. In this data driven world, Database Architecture provides a strong foundation and powerful solution to meet all the data needs of organization and it is the foundation of an organization. As discussed in the analysis, the major technical issues that are related to database architecture are solved in the early stages of architecture planning and designing, the lack of which will impact the organization later.

The role of Database Architects has become more critical as businesses are giving more importance to their data. In previous years, Information architect was the one who used to handle the database architecture, but in today's situation, organizations are expecting expertise in every different enterprise architecture and every single design system. I strongly believe that if a company desires to leverage and operationalize the data proactively, it needs to invest time and money on its data architecture and on the information technology mapping. It is obvious that today, even the search history of a random person on the web server has become important to companies to draw insights.

In my perspective, increase of data and information helps the organizations and opens a broad category of jobs. At the same time, information technology industry should understand the limitation and create certain barriers for their data collection. Otherwise, after a few years it is going to have a negative impact on the IT industry. Collecting any random data about the customer or any random data about other sources does not help the industries in any way and data experts should transfer their learnings to employees and organizations. Quality of data should be given importance while collecting the data.

Data quality is also a key ingredient in a Database architecture. If an organization collects quality data, it is going to be meaningful for them to categorize, correlate, and validate that data into information. Also, companies should focus on cleaning unnecessary and historical junk data. Most of the companies do not concentrate on cleaning up useless data thinking that it can be used in some way or other in the future. This kind of mindset directs to islands of unimportant and useless data which leads to data explosion. In my point of view this situation can be overcome by continuous data governance. In general, business people in the organization see Information Technology architecture as a very expensive and uninteresting model, but a solid information architecture is a combination of several strong embedded architectures and heart of information architecture is Database architecture.

Database architecture is a foundation for data governance program and it assigns the business meaning to it. I also believe that a good database architecture only helps in the function and manipulation of data, but if an organization can hire experts who can assess the data usage and who can define the information from data, then that is going to make a huge positive impact on their organization. Database architecture needs a data modeling tool to develop schemas out

of it and also glossaries that define all the metadata and data efficiently so that useful insights and decision making can be done.

Enterprise architects and organizations should consider following the Enterprise Resource Planning while planning, developing and using the database architecture. With the Enterprise Resource Planning suite model, it is less complex for organizations to collect proper data with valuable meaning to their business in correct time and correct way.

Implementing the big data technologies to the existing models also makes the business insights more valuable and interesting. Every business has a unique set of data sources and data stores. Before proceeding to create a database architecture, organizations should be able to define their business and data needs clearly. This helps the architects to understand and judge what kind of data sources, data repositories, and data platforms need to be designed. Misinterpretation or misunderstanding of business or data needs leads to poor database architecture. Also, with emerging technologies like big data platforms, Hadoop, ERP, SaaS etc., organizations should invest time and money on the research and development. This helps in broadening their perspective about technologies and makes it easier to evaluate suitable technologies.

The ability to visualize the models and mapping the sources out of data makes the organization unique and this automatically drives the industry towards success. In this competitive world of industries, it is very easy to lose track of business, and organizations need to be very careful in utilizing their resources. Organizations need a business driven database architecture to effectively assemble and utilize the data. Various technical and business teams in the organizations should collaborate with each other in the foundation and development of

database architecture. They should communicate as a unit and come up with a unified solution that helps the organization to create best results and to avoid conflicting differences. For this to happen, organizations should create an amicable and flexible culture and opinions should encouraged.

On the other hand, developing a good architecture and making a good use of it is not the end, it also includes optimization and it is an ongoing process. Optimization helps in consistent usage and well being of the architecture. Over a period of time database architecture and its technologies have advanced in a way that Database Management Systems are a very complicated and critical software systems. With decades of research and development in databases, database engineers pioneered reliable design solutions for various business problems. It is not only limited to the data industry but also in various other Information Technology industries as well.

These advancements should be used according to the business requirements otherwise a flood of technologies surround and submerge the organization. For several reasons, the concepts of database architecture and issues in database management system are not as broadley noticed as they should be. The main reason is that the database concepts were considered very complicated diverting many students and employees to other technical domains.

The database design and development communities are very limited and schools are failing to bring up the database architectural point of views and architectural issues. The curriculum concepts of databases include traditional query language, theoretical issues, database software installation and algorithms. While these concepts are taught to serve the purpose of

grades and tests, the holistic discussion about the entire database architecture, working and implementation are not being taught.

In this paper, I have made an effort to cover the major database architectural aspects and major database architecture issues with a comprehensive discussion. I understand and strongly believe through this research paper that a solid design and implementation of Database Architecture is the major ingredient and is fundamental in defining a highly successful organization. An ideal database architecture drives the organization towards better execution of its goals.

Recommendations

- Database design experts should focus more on database scaling.
- Organizations should not maintain too many data ecosystems.
- Robust and powerful data security plan must be implemented in Database architecture.
- Organizations should focus on quality of data.
- Unnecessary data should be reviewed and cleaned from the systems.
- Organization should understand the limitations in scalability.
- Organization should focus strongly on good documentation.
- Organization must research before considering the solutions from vendor.
- The solution providers should not pressurize the organizations about the changing technologies frequently.
- Data issues in an organization should be given utmost importance.

- The database architecture should consider all aspects of business model before implementation.
- Organizational standards should be followed regarding naming convention.
- Database managers should encourage the usage of stored procedures.
- Normalization concepts should not be neglected.
- Organization's data should not solely rely on user's views and user's data.
- Organizations should not allow excessive collection of data which is not productive.
- Database architects must allot stipulated time period to test the functionality in holistic manner.
- Organization should focus on optimization techniques.

Appendices

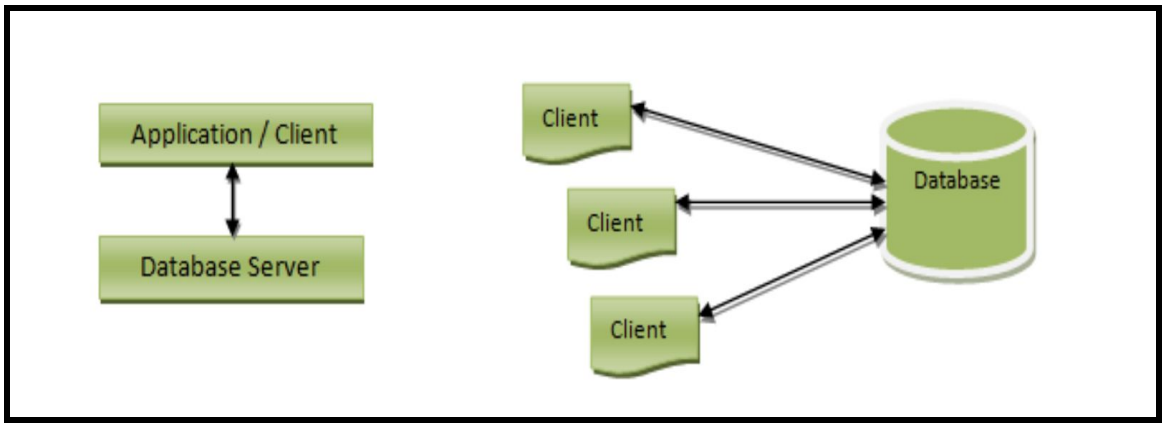

Figure 1: Two-Tier DB Architecture (source: Tutorials point)

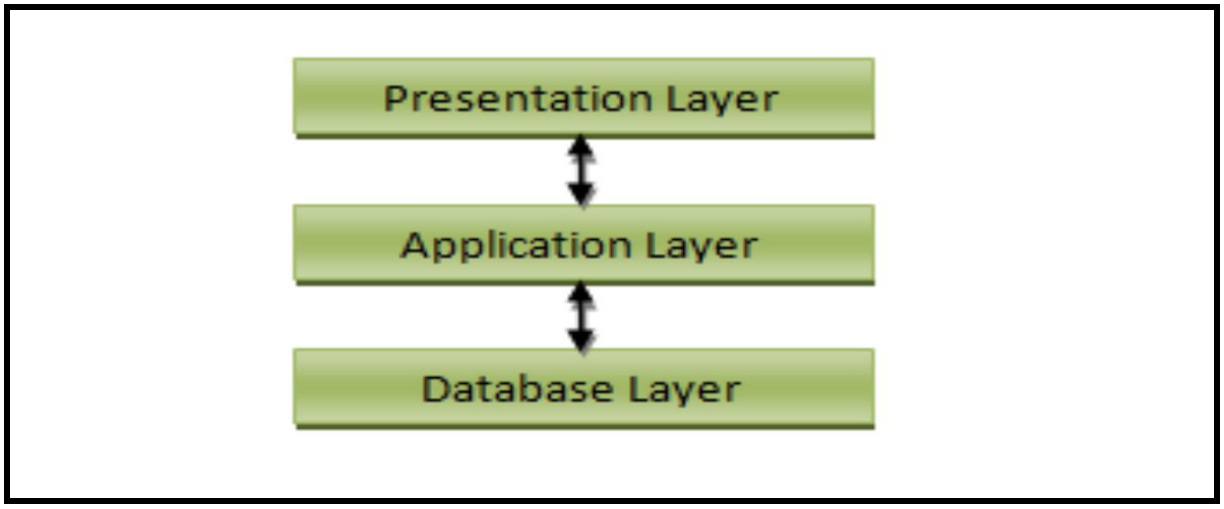

Figure 2: Three-Tier DB Architecture (Source: Tutorials point)

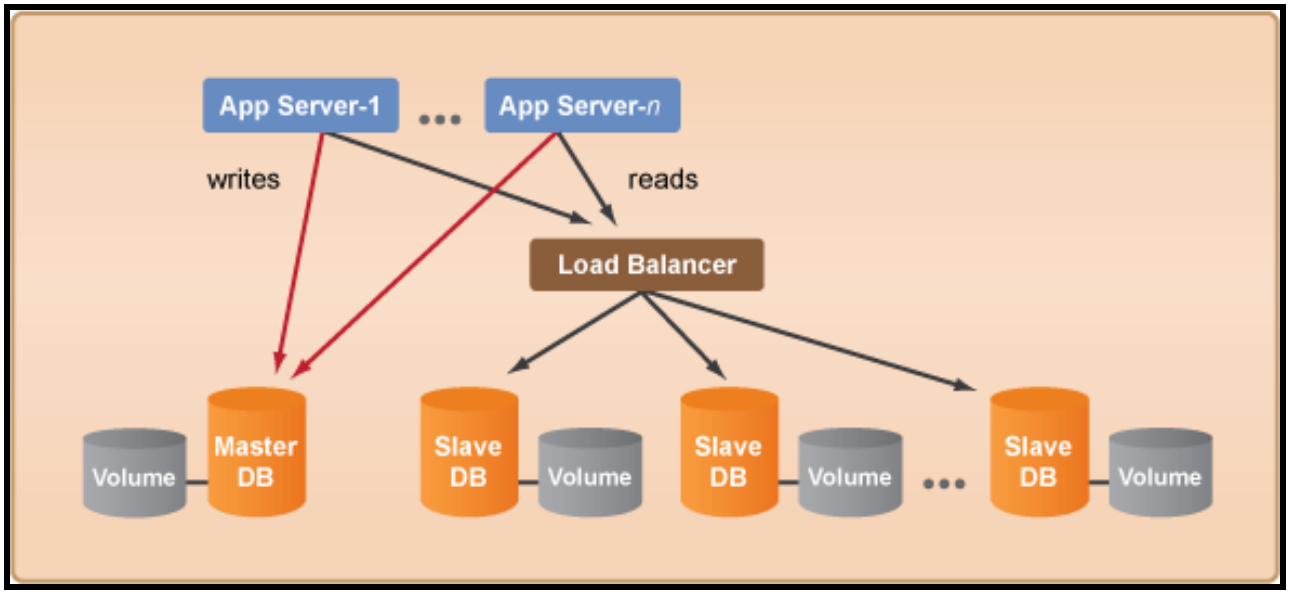

Figure 3: Database scaling example (Source: Rightscale.com)

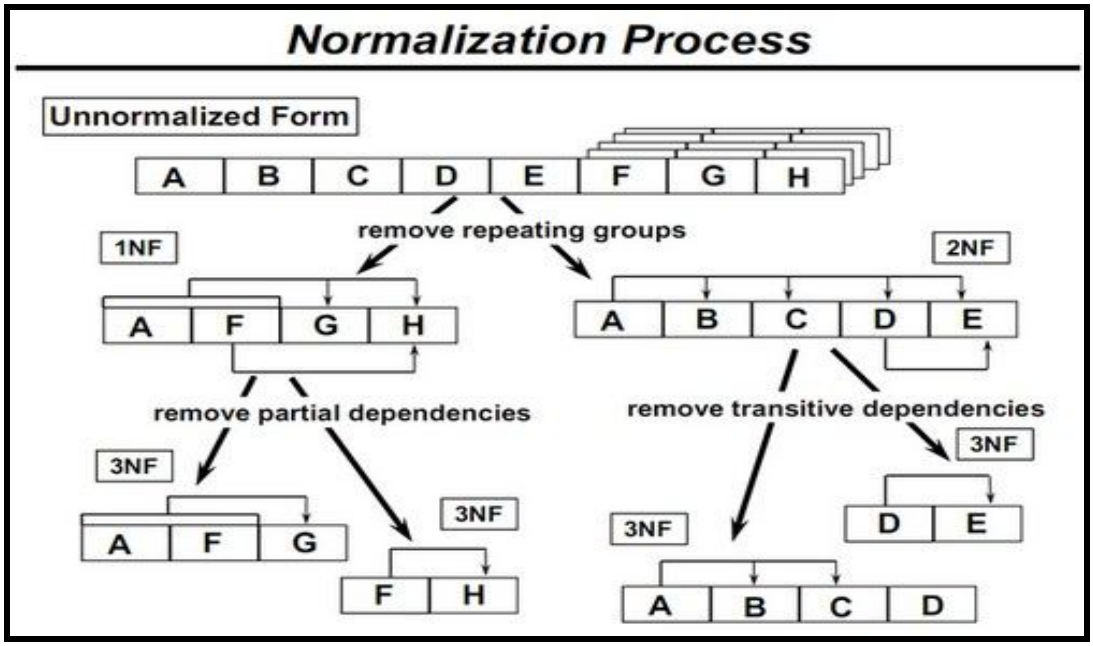

Figure 4: Basic Normalization process (Source: Tutorials point)

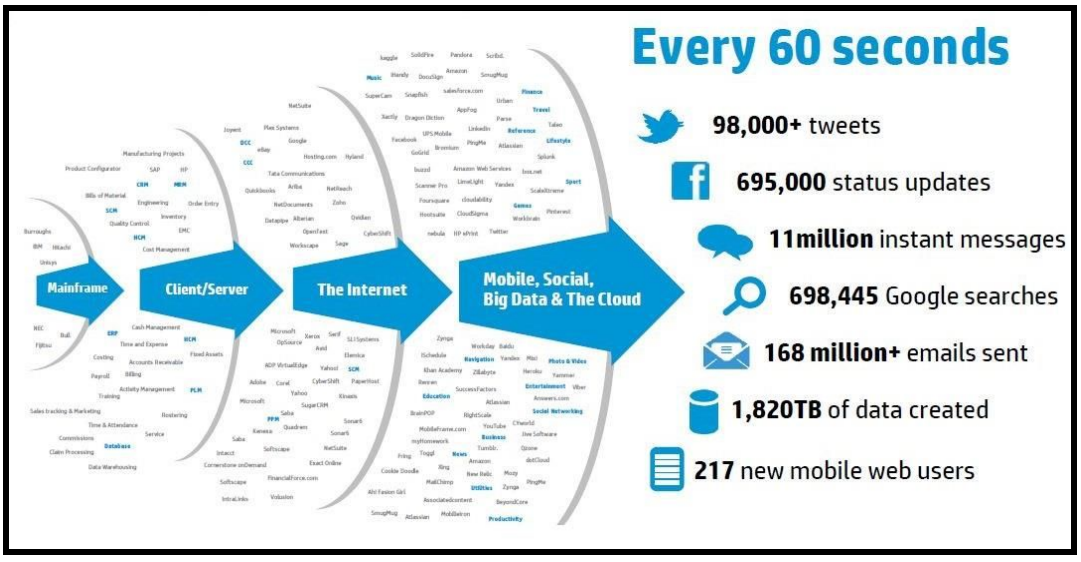

Figure 5: Data-Growth (Source: statista.com)

Endnotes

[1]http://learn.org/articles/What_is_Database.html

[2]http://coronet.iicm.edu/Dbase1/scripts/rdbh12.html

[3]http://ecomputernotes.com/fundamentals/what-is-a-database_architecture.html

[4] http://www.exceltrick.com/formulas_macros/vlookup-in-vba/

[5] https://eng.archinform.net/index.htm

[6] https://www.tutorialcup.com/dbms/architecture-of-database.htm

[7] http://jcsites.juniata.edu/faculty/rhodes/dbms/dbarch.htm

[8] https://docs.oracle.com/cd/B28359_01/server.111/b28318/part_2.htm

[9] http://www.studytonight.com/dbms/architecture-of-database.php

[10] www.irmuk.co.uk/articles/Ron_Huizenga_Top_5_Data_Architecture_Challenges.pdf?

[11]http://www.b-eye-network.co.uk/view/9313

[12]http://www.waterstechnology.com/inside-reference-data/advertisement

[13]https://www.ibm.com/developerworks/library/ar-keyques/

[14]http://www.slideshare.net/embarcaderotechnet/top-5-data-architecture-challenges

[15]http://www.baselinemag.com/storage/slideshows/complex-data.html

[16]https://eavoices.com/2015/07/17/top-5-data-architecture-challenges/

[17]http://Identifying-data-quality-issues-via-data-profiling-reasonability

[18]https://www.20141123164317-13674513-the-top-5-mistakes-in-data-architecture

[19]https://viewevent/188-top-5-data-architecture-challenges

[20]http://www.baselinemag.com/-focus-on-the-business-issues.html

[21]https://www.researchgate.net/publication/269723351_Challenges_and_requirement.htm

[22]https://www.simple-talk.com/sql/database-administration/database-design-mistakes/

[23]https://www.simple-talk.com/sql/database-administration/-errors-you-should-avoid/

[24]www.ijcer.org/index.php/ojs/article/viewFile/497/229

[25]www.oracle.com/technetwork/topics/entarch/articles/oea-big-data-guide-1522052.pdf

[26]https://www.allerin.com/blog/data-that-organizations-must-aim-to-overcome-3

[27]https://www.progress.com/docs/library/Progress/Documents/Papers/-of-Big-Data.pdf

[28]www.dataarchitect.cloud/3-challenges-for-artificial-intelligence-in-medicine

[29]http://knowledge-integrity.com/blog2/wp-content/uploads/2015/12/Enterprise-Data-Architecture-and-Data-Governance-final.pdf

[30]https://www.topcoder.com/tc?module=CompList&ph=118

[31]http://www.ey.com/Publication/vwLUAssets/EY_-_Big_data.pdf

[32]http://blogs.informatica.cominformatica-architects-challenge/#fbid=C8VlJMbJCFp

[33]https://experts.umich.edu/en/publications/challenges--data-architecture-supp

[34]http://datamanagementreview.com//challenges-t-data-governance-strategies

[35]https://books.google.com/books?id=NxO8BQAAQBAJ&pg=PA5&lpg=PA5&false

[36]https://docs.oracle.com/cd/E11882_01/server.112/e40540/logical.htm#CNCPT1095

[37]http://www.vertabelo.com/blog/technical-articles/7-common-database-design-errors

[38]http://searchsdn.techtarget.com/tip/Six-considerations-for-big-data-networks

[39]https://technet.microsoft.com/en-us/library/ms190619(v=sql.105).aspx

[40]www.wingenious.com/database.pdf

[41]https://equizine.wordpress.com/disadvantages-of-database-management-system/

[42]https://www.semanticscholar.org/paper/Building-a-network-highway-for-big-data.html

Bibliography

"Architecture of Databases "http://coronet.iicm.edu/Dbase1/scripts/rdbh12.htm. (Accessed February 23, 2018).

"Addressing five emerging challenges of database". https://www.progress.com/docs/default-source/default-document-library/Progress/Documents/Papers/Addressing-Five-Emerging-Challenges-of-Big-Data.pdf (Accessed February 23, 2018).

"Big data architecture". http://www.ijcer.org/index.php/ojs/article/viewFile/497/229 (Accessed February 24, 2018).

"Building a network highway for big data". https://www.semanticscholar.org/paper/Building-a-network-highway-for-big-data-Yi-Liu/11f7f594f4516deb192197b021dc745a8fd2364d (Accessed February 24, 2018).

"Challenges for developing data architecture". https://experts.umich.edu/en/publications/challenges-and-requirements-for-developing-data-architecture-supp (Accessed February 25, 2018).

"Database performance". https://technet.microsoft.com/en-us/library/ms190619(v=sql.105).aspx (Accessed February 23, 2018).

"Database architecture". http://www.wingenious.com/database.pdf (Accessed February 25, 2018).

"Database management system". https://equizine.wordpress.com/2012/10/22/advantage-ans-disadvantages-of-database-management-system/ (Accessed February 25, 2018).

"Enterprise Data Architecture". http://www.b-eye-network.co.uk/view/9313 (Accessed February 26, 2018).

"Focused Business Issues". Accessed November 6, 2016. http://www.baselinemag.com/analytics-big-data/to-do-analytics-well-focus-on-the-business-issues.html (Accessed February 26, 2018).

"How to manage complex data environment". http://www.baselinemag.com/storage/slideshows/how-to-manage-todays-complex-data-environment.html(Accessed February 27, 2018).

"Key challenges in data architecture". http://www.waterstechnology.com/inside-reference-data/advertisement/2266335/key-challenges-in-data-architecture (Accessed February 27, 2018).

"Key Questions for a data architect". https://www.ibm.com/developerworks/library/ar-keyques/ (Accessed March 1, 2018).

"3 challenges for artificial intelligence". http://www.dataarchitect.cloud/3-challenges-for-artificial-intelligence-in-medicine/ (Accessed March 2, 2018).

"Enterprise data architecture and data governance"..

http://knowledge-integrity.com/blog2/wp-content/uploads/2015/12/Enterprise-Data-Architecture-and-Data-Governance-final.pdf (Accessed March 2, 2018).

"Challenges for developing data architecture".

https://experts.umich.edu/en/publications/challenges-and-requirements-for-developing-data-architecture-supp (Accessed March 5, 2018).

"The challenges of data governance Strategies".

http://datamanagementreview.com/regulation-and-compliance/blog-entry/challenges-and-opportunities-smart-data-governance-strategies (Accessed March 6, 2018).

"Logical Storage Structures".

https://docs.oracle.com/cd/E11882_01/server.112/e40540/logical.htm#CNCPT004 (Accessed March 6, 2018).

"7 common database design errors".

http://www.vertabelo.com/blog/technical-articles/7-common-database-design-errors (Accessed March 7, 2018).