**Forum: Week 1 Discussion:**

1. **Why a database?**

* **Consider your program or an industry to give an example of how database management systems are being used for business.**
* **Describe the meaning or the relationship between data and information in this context**

In today's modern world, everything is instantaneous, synchronous, and parallel in performing most of the tasks. We all seek tons of information daily from various sources and all these data are stacked and backed up by the large, complex, and intelligent machines called **'Databases'**. We do require lots of volumes of storage to respond to the user queries/ requests made in less/ no time. Many organizations, businesses, institutions, and governments work on huge amounts of data and it is necessary to maintain the database for easy retrieval process, highly-optimized queries, storage management, data privacy and security, regulations, and other standards that are designated to maintain it.

Now, Let's take an example of banks and financial organizations where massive amounts of data are being processed among millions of users every day. Here comes our in-house highly-secured, maintained, and managed **'Database management Systems'** into the picture. Mainly banks do work with financial institutions, businesses, governments, and people on most of the services. Banks use the latest technologies and methods to make the transactions and services go in a sleek, safe, and secure way. They need a database to store customer information, loans, account activities, transactions, statements, and other key aspects. This is used for an effective way of handling things without any major issues to the customers.

In simple views, **Data** is like anything and everything that is available anywhere and everywhere in any format. **Information** is something insightful and relevant to the circumstances and conditions which are drawn from the data. Here the example taken is Banking and its related fields, so based on the requirement, tasks are performed accordingly in an organized and secure way with proper information to a particular user or customer.

The most and latest Databases used by many organizations and companies are **MySQL, MongoDB, PostgreSQL, Oracle, MS SQL Server, and MariaDB**

The most and latest Cloud Platforms used by many organizations and companies are **AWS, Google Cloud Platform, MS Azure, Salesforce, IBM Cloud, and Alibaba Cloud**

1. **Consider how the infrastructure of computing services is changing and how this is having an effect on the ways we designate our database operations. Two types are named in the readings, distributed and centralized. From several contexts, we could draw the meanings at first glance but for this thread emphasize through your posts some of the benefits to both when applied to databases.**

* **What are the benefits of distributed databases?**
* **What are the benefits of centralized databases?**
* **What is your preference and why?**

The evolution of maintaining databases from the past 50 years has significantly changed a lot and now it's growing even more unprecedentedly. The use of the internet and the availability of data are important parameters for the spike in demand for maintaining information everywhere. Traditional databases cannot risk to maintain and manage the current and future demands of the user requests. So, most of the companies are transitioning to Cloud Computing, Distributed Computing, Parallel Computing, and Edge Computing depending upon the demand, users, devices, and networks. The future of computing relies on these sophisticated and complex technologies to meet the needs of the users.

**Pros and Cons of Centralized and Distributed Databases**

|  |  |
| --- | --- |
| Centralized | Distributed |
| 1. Stored in single place with limited amount of computers 2. Easier to manage, operate, and upgrade the resources 3. Backups are lot more convenient and easier 4. Usually, managed by DBMS 5. Discrepancies in handling the requests and granting the data to multiple users at a time 6. Chances of occurring huge data losses as it serves only in one location. Zero or less data recovery possibilities 7. Cannot be accessed from remote areas 8. Easier to keep track of data and its movements 9. Less reliable compared to distributed ones 10. Data redundancy is minimal and Data integrity is maximal | 1. Stored in multiple servers in diverse locations and connected by a network of computers and other devices 2. Used for better availability and reliability to offer continuous services 3. Usually, managed by DDBMS 4. Scaling can be done automatically based on the requirement, size, location, users, dependencies, and processors 5. Expensive when complex database architecture is maintained 6. Performs multiple queries at multiple locations and executes the tasks parallelly without any hurdles 7. Build on robust systems and capable of handling fault-tolerant affects 8. Better protection and safety of data 9. OLTP and OLAP Support 10. Complex and costly software setup required |

My preference is to go with **Distributed Computing**. Because of its easy accessibility, processing speeds, memory management, storage solutions, and restoration techniques. It is more efficient and effective than centralized ones to meet user needs.

1. **Explain what is meant by entity integrity and referential integrity. Why are both important in a database?**

Entity means any place, thing, or component in the [database](http://www.papertyari.com/general-awareness/it-knowledge/introduction-database-management-systems/). An entity represents a real-world object. Each table represents an entity and each row in a table is the instance of an entity. **Entity integrity** in databases is a condition or a constraint where the rows in all of the tables can be uniquely identified using the primary key of its corresponding table. The primary key assures the entity integrity constraint is applied to a table. It also states that value in primary key fields cannot be null. Therefore, no primary key column of any row in a table can have a null value.

**Referential integrity** constraint refers to the relationship between the tables. The relationship between tables is established by using foreign keys. This constraint states that a foreign key must have a matching primary key in another table or must be null. Hence, every foreign key must contain a null value or a valid key reference in another table.

**Importance: ﻿Both, Entity and Referential integrity plays a vital role in fetching a specific row of data from a table to ensure that only a valid foreign key can be assigned to a table. It's needed to maintain data integrity﻿﻿﻿﻿﻿﻿ and no way for data inconsistencies. ﻿﻿﻿﻿﻿﻿﻿﻿﻿﻿﻿﻿﻿﻿﻿﻿**

1. **Explain Metadata. How is it used? What's its importance to our context? Explain through a post that combines your own understanding and a source of reference in support.**

**Metadata** can be explained in below few ways:

* Data that provide information about other data in a database
* It summarizes basic information about data make instances of data easier to work
* It can be created manually to be more accurate, or automatically and contain more basic information

Classes, instances, relationships, properties, functions, processes, constraints and rules relating to items are a few attributes of a Metadata.

Every time when we take a photo using our smartphones/ cameras, a group of metadata is gathered and saved with it, like name, date, size, file format, time, location and other key information of a particular image. This same works with videos, audios, applications, docs, and whatnot.

Another example to quote is, when we place an order online, we usually give our personal information, payment details, our interests, location details, and etc., After placing the order, the information related to a particular user is stored in the database. So, this kind of information helps businesses to recommend custom products based on your preferences and information using predictions and modelings.

Metadata in some ways have interconnected elements like

* **What**
* **When**
* **Where**
* **Who**
* **How**
* **Which**
* **Why**

In short, metadata is important. Certainly, metadata schema can be simple or complex, but they all have some things in common. It can range from one-dimensional to multi-dimensional.

**Forum: Week 2 Discussion:**

1. **Choose and respond to one of the following questions on the relational database model:**

* **A database user manual note that "The file contains two hundred records, each record containing nine fields." Use the appropriate relational database terminology to "translate" that statement.**

**Or**

* **Explain why the data dictionary is sometimes referred to as "the database designer's database".**

**Relational Database Terminology:**

* **Entity:** Entity means any place, thing, or component in the database in which data are to be collected and stored. An entity represents a real-world object and are distinguishable. Each table represents an entity and each row in a table is the instance of an entity. Each entity occurrence is unique and distinct
* **Tuples:** A collection of ordered attributes for a single entity. A row or record is called a tuple
* **Attributes:** A property of an object, element, or file. A column or field. An attribute that must have some value
* **Relational Database:** A database that has a collection of tables of data items
* **Relation:** A set of tuples that have the same attributes is called a relation

**Translated Statement:**

The note can be translated into the following **Relational Database Terminology** as:

**"An entity set contains 200 tuples/ rows with each record consisting of 9 attributes/ columns"**

1. **Explain, for entity relationship modeling, both UML and Crow's Feet describing the benefits from the use of each in database design. Provide an example of each.**

**Essentials:**

* **Entity:** Entity means any place, thing, or component in the database in which data are to be collected and stored. An entity represents a real-world object and are distinguishable. An entity shape is a data object. In a database, an entity is normally represented by a table. Each table represents an entity and each row in a table is the instance of an entity. Each entity occurrence is unique and distinct
* **Attribute:** An attribute shape is a property of an entity. A property of an object, element, or file. A column or field. An attribute that must have some value. Attributes can be used to identify an instance of an entity (called a primary key). Attributes can also be part of a related table where the shared attribute is used to join the data
* **Relationship:** These shapes are used together to show the structure of a database and the relationship between tables. Database diagrams can be used to design database systems or show how to query the database
* **Database Design**: Depending on the scale of change, it can be risky to alter a database structure directly in a DBMS. To avoid ruining the data in a production database, it is important to plan out the changes carefully. ERD is a tool that helps. By drawing ER diagrams to visualize database design ideas, you have a chance to identify the mistakes and design flaws, and to make corrections before executing the changes in the database
* **Entity Relationship Modelling:** An entity-relationship diagram shows the relationships of entity sets stored in a database. An entity in this context is a component of data. In other words, ER diagrams illustrate the logical structure of databases. At first glance, an entity-relationship diagram looks very much like a flowchart. It is the specialized symbols, and the meanings of those symbols, that make it unique. In the ER Model, we disintegrate data into entities, attributes and set up relationships between entities, all this can be represented visually using the ER diagram
* **Primary Key**: Also known as PK, a primary key is a special kind of entity attribute that uniquely defines a record in a database table. In other words, there must not be two (or more) records that share the same value for the primary key attribute
* **Foreign Key:** Also known as FK, a foreign key is a reference to a primary key in a table. It is used to identify the relationships between entities. Note that foreign keys need not be unique. Multiple records can share the same values.

**Overview of Cardinalities in Data Modelling:**

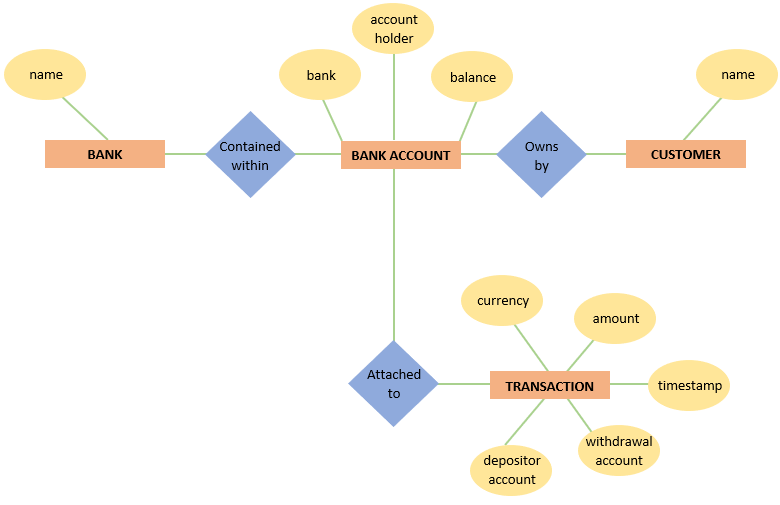
In data modeling, cardinality refers to the relationship of data in one database table with respect to another table. Two tables can be related as “one-to-one”, "one-to-many", or "many-to-many":

* **1: 1:** One row in table A relates to one row in table B. Using an entity-relationship (ER) model, 1:1 means that one occurrence of an entity relates to only one occurrence in another entity
* **1: Many:** One row in table A relates to many rows in table B. In ER modelling, 1: Many means that one occurrence in an entity relates to many occurrences in another entity
* **Many: Many (M: N):** Many rows in table A relate to many rows in table B. In ER terms, many occurrences in one entity relate to many occurrences in another entity. For example, a student (table A) may sign up for many classes (table B), and a class may have several students in it. Many-to-many relationships normally require a cross-reference table, AB with two one-to-many relationships A: AB and B: AB

**Chen’s Notation:**

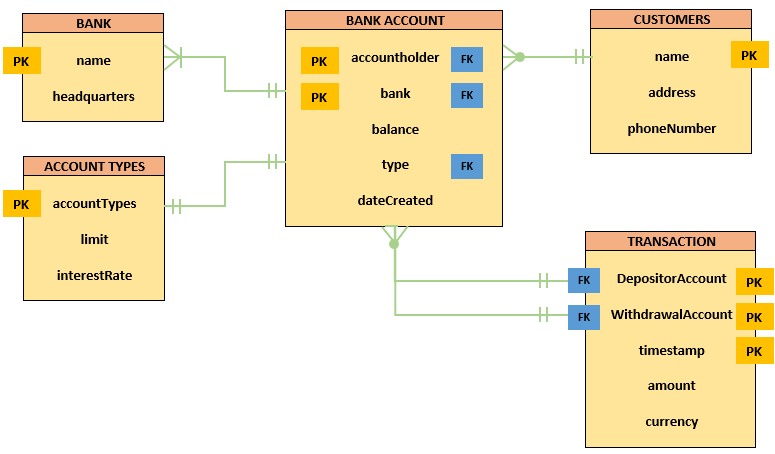
Chen’s is useful for modeling the basics of entities and relationships, because it presents an abstract view of the associations. Entities are represented by rectangles. Attributes are circular callouts to the entities. Relationships connect the entities with a diamond shape.

**Example:**



**Crow’s Foot Notation:**

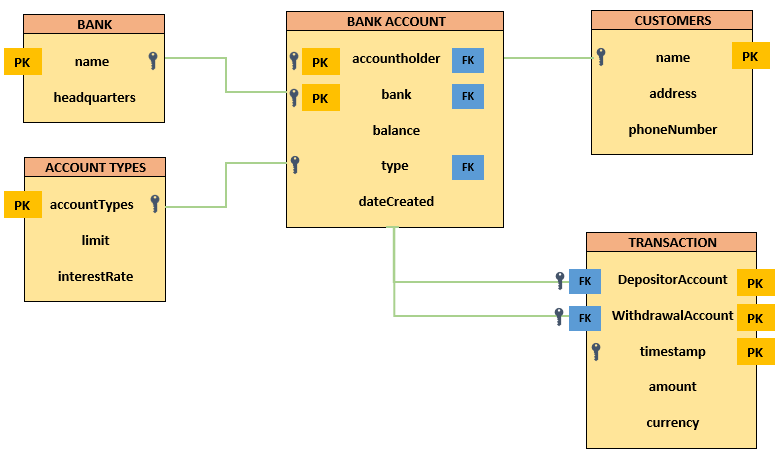
Crow’s Foot is the most commonly used notation. It’s good for a clean look when there are many attributes, such as a model of a complex database system. Attributes are encapsulated within the entity shapes. Relationship connectors are used to show an association between entities. Arrowheads are used to describe the cardinality of the relationship. Relationships can be attached to the entire entity or to a specific attribute.

**Example:**

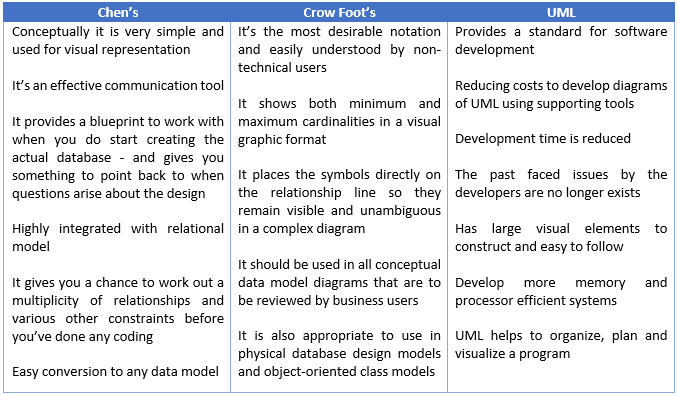
**UML Notation:**

UML, the Unified Modeling Language, is a standard for object-oriented design and used as a notation for modeling databases. While UML does not have specifications for data modeling, it can be a useful tool for diagramming, especially since data from databases can be used in object-oriented programming. The UML database shapes are very similar to the UML class shapes that are used to model object-oriented classes. UML database shapes are a good way to have a consistent notation for an entire system.

**Example:**

****

**Benefits:**

****

1. **Designing the Database:**

**Proper database design requires the designer to identify precisely the database’s expected use**

* **Designing a transactional database emphasizes accurate and consistent data and operational speed**
* **Designing a data warehouse database emphasizes the use of historical and aggregated data**
* **Designing a database to be used in a centralized, single-user environment requires a different approach from that used in the design of a distributed, multiuser database**

**Database Design:** It is a collection of processes that facilitate the designing, development, implementation, and maintenance of enterprise data management systems. The properly designed database is easy to maintain, improves data consistency, and is cost-effective in terms of disk storage space. The database designer decides how the data elements correlate and what data must be stored.

Proper database design requires the designer to identify precisely the database's expected to use

* Designing appropriate data repositories of integrated information using the two-dimensional table structures found in most databases is a process of decomposition
* The integrated data must be decomposed properly into its constituent parts, with each part stored in its own table
* Further, the relationships between these tables must be carefully considered and implemented so an integrated view of the data can be re-created later as information for the end user
* A well-designed database facilitates data management and generates accurate and valuable information
* A poorly designed database is likely to become a breeding ground for difficult to-trace errors that may lead to bad decision making
* Database design is simply too important to be left to luck. That's why college students study database design, why organizations of all types and sizes send personnel to database design seminars, and why database design consultants often make an excellent living

Good database design is, therefore, one that: Divides your information into subject-based tables to reduce redundant data. Provides Access with the information it requires to join the information in the tables together as needed. Helps support and ensure the accuracy and integrity of your information. Accommodates your data processing and reporting needs.

1. **Explain the differences between data, information, and a database. Draw from your own experience and knowledge on data processing. How has your view of data changed based over the last decade?**

**Differences between Data, Information, and a Database:**

|  |  |  |
| --- | --- | --- |
| Data | Information | Database |
| Data is like anything and everything that is available anywhere and everywhere in any format  Data are factual information such as measurements or statistics about objects and concepts  Data can be a person, a place, an event, an action or any one of a number of things | It is something insightful and relevant to the circumstances and conditions which are drawn from the data  Data are useless in their present form so it’s preprocessed and represented to the user  Converted into the meaningful form during the processing phase of the information cycle | A database is a shared collection of related data used to support the activities of a particular organization  It is defined once and then accessed by various users  A database is logical, coherent, and internally consistent. Each data item is stored in a field. A combination of fields makes up a table |

**Data Processing:**

Data processing occurs when data is collected and translated into usable information. It is important for data processing to be done correctly as not to negatively affect the end product or data output. Data processing starts with data in its raw form and converts it into a more readable format (graphs, documents, etc.), giving it the form and context necessary to be interpreted by computers and utilized by employees throughout an organization. In my recent project on COVID 19, I have gone through the following steps to maximize the availability of neat and clean data to the users. Data is collected from genuine data sources and then prepared as required. I have chosen a few input parameters to process the data and to involve as the key indicators and interpreting the outcome. This can be stored and shared on any cloud or file or pc or any smart devices

**Steps involved in this are:**

1. Data Collection
2. Data Preparation
3. Data Inputs
4. Processing
5. Data Output/ Interpretation
6. Data Storage

**Insights on Data Transformation:**

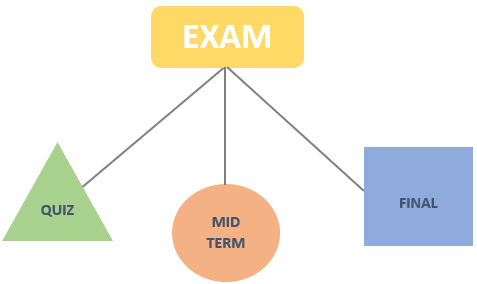
I have seen a lot of tremendous transformations over the past few decades especially in the fields of data and its related fields. The way its used, maintained, stored, distributed, shared has changed a lot. There are a lot of more diversified business use-cases and applications when compared to a decade ago one’s. It got a lot better on the Internet, Social media, AI, ML, NLP, Analytics, Computer Vision, Blockchain, Fintech, Healthcare, Security, IoT, Big Data, and so on and so forth. Information technologies have shaped fundamental change throughout society, driving it forward from the industrial age to the networked era. More and more opportunities have been created and providing new benefits to individuals. The substantial change in view of products and services offerings now-a-days is just mind-boggling. The future has just begun and will be fruited by us if utilized in the right way for the right things.

**Forum: Week 3 Discussion:**

1. **Explain what is an entity supertype, and why is it used. What is an overlapping subtype? Give an example of each in your response.**

* **Entity Supertype:** An entity that is constructed with attributes that will be common to all of the subtypes related to it. Supertypes and related subtypes support specialization hierarchy, which is another set of features that entity types may have. The reason for using supertypes is to minimize the number of nulls and to minimize the likelihood of redundant relationships
* **Entity Subtype:** An entity that is based on a supertype, but is different from the supertype because of its specialized attributes and purpose. It may be obvious that each subtype is related to one supertype, and each supertype may be related to no subtypes, one subtype, or many subtypes

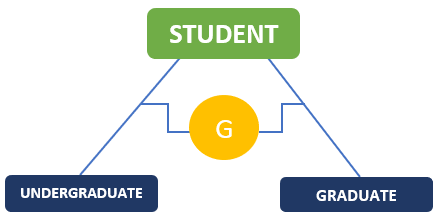
**Example:** The text presents the example of a supertype called Employee, which will have all the attributes that are common to various categories of employees in an organization. Subtypes of this entity might include managers, salaried staff, commissioned staff, part time staff, full time staff, probationary staff, or whatever categories the organization actually uses

****

**Subtypes can be disjoint or overlapping:**

* **Disjoint Subtypes**: These are also called non-overlapping. They contain attributes that can be one of two (or more) values, but cannot be a combination of those values, such as an employee being hourly or salaried, but never being both
* **Overlapping Subtypes**: These contain a combination of attribute states which prevent there being one attribute to represent the state. These are overlapping states and they are said to be non-unique because of the overlap. It may be best to express these states in two logical fields, showing the states of each as True or False

**Example:** An employee who has been promoted will be in a probationary status, but will also be either hourly or salaried. In this case the employee may be hourly, hourly and probationary, salaried, or salaried and probationary. The text offers an example of an employee of a school who may be an administrator, a professor, both, or neither

****

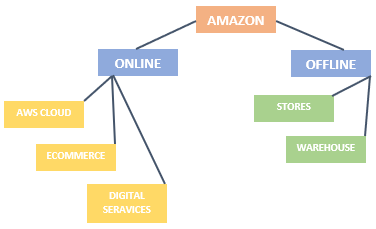
1. **Explain what is meant by a specialization hierarchy. Provide an example in your response.**

**Specialization Hierarchy:**

* It is based on the top-down process of identifying lower-level, more specific entity subtypes from a higher-level entity supertype
* It depicts the arrangement of higher-level entity supertypes (parent entities) and lower-level entity subtypes (child entities)
* It is s “is-a” relationship model
* Subtype entity is related to an entity supertype, where the supertype entity has common characteristics and the subtype entity has the unique characteristics of each entity subtype
* Specialization is based on grouping unique characteristics and relationships of the subtypes

**Example:**

* Here I took the example of Amazon Company to denote the top-down hierarchy
* In this, **Amazon** is at the top level, also known as supertype or parent entity
* **Online** and **Offline** streams of **Amazon** are known for subtypes/ supertypes. These are lower level to its parent i.e., **Amazon**
* **AWS Cloud, Ecommerce,** and **Digital Services** are few domains where the services and products are provided through **Online** and **Stores, and Warehouses** are few domains offered in **Offline.** These are lower level to its parent entities i.e., **Online** and **Offline**

****

1. **In the article published to CIO.com (found here), the author Kim Nash refers to use of data in support of business decisions, operations, and development strategies. Data in other words is the fuel and the fuse behind today's business. Her premise is one that with so many new leadership roles engaged in big data and data mining initiatives, the one typical responsible is often the one left out, the Chief Information Officer. So, I offer the following open question to establish a conversation on the article:**

**Why is it so important for the technical staff to be the leaders in database management and analytics strategies? What lessons came we extract from this article to support our Week 3 topics?**

In today’s digital world, businesses recognize the value of data as an asset and are leveraging new technologies and business models to generate insights and deliver customer value, thereby monetizing data in real-time. Digital transformation has put data at the center of every organization. Businesses are packed with data. They struggle to identify what is most important and what actions to take (or avoid).

Data and analytics leaders should actively monitor, experiment with or deploy emerging technologies. Don’t just react to trends as they mature. Use this list to educate and engage with other leaders about business priorities and where data and analytics can build a competitive advantage.

A true CIO should chair an enterprise data council that includes key business owners, the CFO, risk and legal, and establishes enterprise data policy and governance along with a consistent communications plan and strategy. This leader should own responsibility for data governance, definition, standards, strategy, controls, architecture, tools, and technology, which are easily centralized.

**The five core components of a data strategy**

* **Identify:** Identify data and understand its meaning regardless of structure, origin or location
* **Store:** Persist data in a structure and location that supports easy, shared access and processing
* **Provision:** Package data so it can be reused and shared, and provide rules and access guidelines for the data
* **Process:** Move and combine data residing in different systems, and provide a unified, consistent data view
* **Govern:** Establish, manage and communicate information policies and mechanisms for effective data usage

**Key goals:**

* Develop a comprehensive strategy to unlock value in data both internally and as a product
* Enable the organization with accurate, reliable and actionable data using a 360-degree view of the customer
* Serve as the champion for the data organization and drive a cultural shift to embrace and leverage data
* Improve the management and use of data through technology, governance, and processes

**Reply:**

Agree with you Tripti! We are in the age of fast-paced digital transformation﻿﻿﻿ and one needs to adapt and be flexible with the ongoing and upcoming change. We, as Developers, Analysts, C-Level Executives, Strategists, Mathematicians, Authors, Lawyers, Administrators, Regulators, Engineers, and others, need to be more integral and ethical towards what we deal, what we do, what we think, what we act, how we perceive things, how we analyze the user expectations, how to enrich the user experience, how to leverage businesses, how to implement various strategies to make the company lucrative, how to satisfy customers, what features need to be included, what are the essentials to consider and what not to be, how to retain customers loyalty, how to make companies more customer-oriented, how to face data privacy issues, how to consider all kinds of regulators and legalities, how to notify users personalization recommendations, how to train models, what are the implications might be faced, and more and more...

All these questions need to be answered or at least make them into considerations when developing a product or a service. So, the company can be in compliance with data privacy and data integrity standards. Users should get the feasibility of how their data can be utilized, what data can be provided, what cannot be, and so on so forth.

Let's play and work around with data in the coming days even more ethically, strategically, legally, morally, and genuinely to leverage the organization's goals.

1. **This will be an open forum, one in which you may choose a topic from this week's textbook reading to write about and develop a question for us to answer. Contributions should be a 200-word post, be sure to cite and including any references used in the primary post.**

**Business Rules:**

* Properly document and verify all business rules with the end-users
* Ensure that all business rules are written precisely, clearly, and simply. The business rules must help identify entities, attributes, relationships, and constraints
* Identify the source of all business rules, and ensure that each business rule is justified, dated, and signed off by an approving authority

**Entity’s:**

* Should be unique within the model. Each entity should represent a set of distinguishable entity instances
* All entities should be in 3NF or higher. Any entities below 3NF should be justified
* The granularity of the entity instance should be clearly defined. The PK should be clearly defined and support the selected data granularity

**Attributes:**

* Should be unique within the entity and be descriptive of the characteristic. Should be simple and single valued (atomic data)
* Should not be redundant unless this is required for transaction accuracy, performance, or maintaining a history
* Non-key attributes must be fully dependent on the PK attribute

**Relationships:**

* Should be active or passive verbs that clearly indicate the nature of the relationship
* Should clearly identify relationship participants
* Should clearly define participation, connectivity, and document cardinality

**ER model:**

* Should be validated against expected processes: inserts, updates, and deletions. Should evaluate where, when, and how to maintain a history
* Should not contain redundant relationships except as required. Should minimize data redundancy to ensure single place updates
* Should conform to the minimal data rule: All that is needed is there, and all that is there is needed

**Question:**

**What if the Business Rules were not followed and aligned to the compliances and regulations in an organization? What are the repercussions and risks that may occur? How an organization can tackle these situations? Quote any practical examples of organizations who had met with this kind of similar problems and how did they overcome this?**

**Reply:**

**Disjoint Constraints:**

* Also known as nonoverlapping subtypes. Implementation is based on the value of the subtype discriminator attribute in the supertype.
* You will need to decide whether a supertype instance may simultaneously be a member of two or more subtypes. The disjoint rule forces subclasses to have disjoint sets of entities

**Overlapping Constraints:**

* Implementation requires the use of one discriminator attribute for each subtype. Composite attributes whose subparts pertain to various subtypes.
* Each subpart has a Boolean value that indicates whether or not the instance belongs to the associated subtype.
* The overlap rule forces a subclass (also known as a supertype instance) to have overlapping sets of entities

**Completeness Constraints:**

* It decides whether a supertype instance must also be a member of at least one subtype
* **The total completeness**: It demands that every entity in the superclass belong to some subclass and is symbolized with a double line connection between entities
* **The partial completeness**: It allows an entity to not belong to any of the subclasses. It is represented with a single line connection

Constraints play an important role in determining the best database design for an enterprise. Constraints are necessary and essential in improving Enhanced Entity Relationship Models. They play a crucial part in the designs and drawing relationships among them.

**Forum: Week 4 Discussion:**

1. **The database life cycle (DBLC) defines the stages involved in implementing a database, starting with requirements analysis and ending with monitoring and modification. Further, the DBLC is continuous with database monitoring, adjustment, and maintenance critical to the life cycle. For this discussion topic, research what is meant by the DBLC including the particular stages. What is the relationship between the SDLC and the DBLC?**

**Database Life Cycle (DBLC):**

* Database development is just one part of the much wider field of software engineering, the process of developing and maintaining software
* A core aspect of software engineering is the subdivision of the development process into a series of phases, or steps, each of which focuses on one aspect of the development
* The collection of these steps is sometimes referred to as a development life cycle
* The software product moves through this life cycle (sometimes repeatedly as it is refined or redeveloped) until it is finally retired from use
* Ideally, each phase in the life cycle can be checked for correctness before moving on to the next phase
* The overall purpose of the database initial study is to analyze the company situation, define problems and constraints, define objectives, and define the scope and boundaries

**The Six Stages in DBLC:**

**Database Initial Study:**

* Analyze the company situation. General conditions in which company operates, its organizational structure, and its mission. Discover what company’s operational components are, how they function, and how they interact. Define problems and constraints
* Interactive and iterative processes required to complete first phase of DBLC successfully
* As a result, analysis focuses on ‘What is required?’ not ‘How is it achieved?’ Data analysis is a highly skilled task and the analyst has a specialized role

**Database Design:**

* Necessary to concentrate on data characteristics required to build database model
* Two views of data within system are Business view - Data as information source and Designer’s view - Data structure, access, and activities required to transform data into information

**Implementation and Loading:**

* Implementation of the logical schema in a given DBMS requires a very detailed knowledge of the specific features and facilities that the DBMS has to offer
* Implementation may be staged, usually with an initial system than can be validated and tested before a final system is released for use

**Testing and Evaluation:**

* Testing is usually considered to involve two main tasks – validation and verification
* Validation attempts to confirm that the right database has been constructed with the right characteristics to meet the specified requirements
* Verification ensures that the processing steps, constraints and other ‘programmed’ components of the database (security, backup, recovery, audit trails, etc.) have been correctly implemented and contain no errors in program logic or execution sequences

**Operation:**

* Once database has passed evaluation stage, it is considered operational
* Beginning of operational phase starts process of system evolution. Problems not foreseen during testing surface
* Solutions may include Load-balancing software to distribute transactions among multiple computers and Increasing available cache

**Maintenance and Evolution:**

* Required periodic maintenance like Preventive maintenance (backup), Corrective maintenance (recovery), Adaptive maintenance
* System changes and enhancements occur during the maintenance and evolution phase
* It involves dealing with changes in the requirements, the implementation environment, operational maintenance, bug fixing or porting of the system to new environments

**Relationship between SDLC and DBLC:**

* SDLC (System Development Life Cycle): It traces the history (life cycle) of an information system from its inception to its obsolescence. It is composed of six phases: Planning, Analysis, Detailed System, Design, Implementation and Maintenance. Database design and application development mapped out and evaluated. Iterative rather than sequential process
* DBLC (Database Life Cycle): It traces the history (life cycle) of a database system from its inception to its obsolescence. It is composed of six phases: Database initial study, Database design, Implementation and loading, Testing and evaluation, Operation, Maintenance and evolution
* Since the database constitutes the core of an information system, the DBLC is concurrent to the SDLC

1. **Explain the relationship between a database and an information system. How does this relationship have a bearing on database design? What aspects of database design are essential for the administrator and developer alike?**

**Relationship between a Database and an Information System:**

An information system performs three sets of services:

* It provides for data collection, storage, and retrieval
* It facilitates the transformation of data into information
* It provides the tools and conditions to manage both data and information

Basically, a database is a fact (data) repository that serves an information system. If the database is designed poorly, one can hardly expect that the data information transformation will be successful, or is it reasonable to expect efficient and capable management of data and information. The transformation of data into information is accomplished through application programs. It is impossible to produce good information from poor data, and, no matter how sophisticated the application programs are, it is impossible to use good application programs to overcome the effects of had database design. In short, Good database design is the foundation of a successful information system.

**Database design must yield a database that:**

* Does not fall prey to uncontrolled data duplication, thus preventing data anomalies and the attendant lack of data integrity
* Is efficient in its provision of data access
* Serves the needs of the information system

In short, good database designers must pay close attention to the information system requirements. Systems design and database design are usually tightly intertwined and are often performed in parallel. Therefore, database and systems designers must cooperate and coordinate to yield the best possible information system.

**Database Administrator:**

* Planning for and allocating the database system’s physical requirements, such as memory, disk space, network requirements, etc.
* Modifying the database structure using information provided by application developers
* Ensuring compliance with database vendor license agreement, including number of installations, and taking care of licensing renewals
* Creating a backup and recovery strategy for the database, and regularly testing the backups to ensure usability
* Creating reports by querying from database. Monitoring and optimizing the database’s performance
* Migrating database instances to new hardware and new versions of software from on premise to cloud-based databases and vice versa

**Database Developer:**

* The database developer designs and creates new ways of using a software application
* A database developer on the other hand is more closely aligned to the developers. We call them Database Engineers or DBEs here
* A DBE embeds with various development teams to help design schema, write queries and stored procedures, etc.
* They also work with the DBAs to determine what queries can be tuned to provide better performance

1. **The ABC Car Service & Repair Centers are owned by the SILENT car dealer; ABC services and repairs only SILENT cars. Three ABC Car Service & Repair Centers provide service and repair for the entire state. Each of the three centers is independently managed and operated by a shop manager, a receptionist, and at least eight mechanics. Each center maintains a fully stocked parts inventory. Each center also maintains a manual file system in which each car’s maintenance history is kept: repairs made, parts used, costs, service dates, owner, and so on. Files are also kept to track inventory, purchasing, billing, employees’ hours, and payroll. You have been contacted by the manager of one of the centers to design and implement a computerized system. So far you have: Interviewed the shop manager. Interviewed the mechanics. Obtained a general description of company operations (inventory, payroll, work orders, customers, etc.). Created a description of each system process. Describe the database design for the combined system and create a conceptual model, using E R diagrams of the various modules that you believe the system should include.**

A good designer uses a specific and ordered set of steps through which the database design problem is approached. The steps are generally based on three phases: analysis, design, and implementation. These phases yield the following activities:

**Analysis:**

* Interview the shop manager and the mechanics
* Obtain a general description of company operations and Create a description of each system process

**Design:**

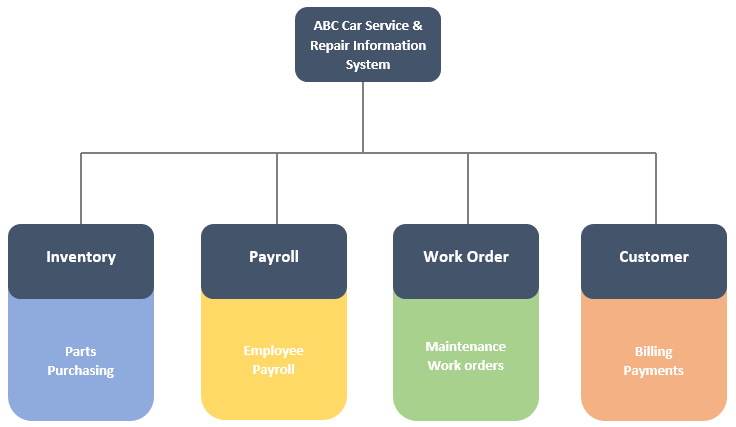
* Create a conceptual model, using E-R diagrams
* Draw a data flow diagram and system flow charts and Normalize the conceptual model

**Implementation:**

* Create the table structures. Load the database
* Create the application programs and Test the system

**There are 4 modules and each module consist of various sub-modules.**

* **Inventory Module:** It include the Parts and Purchasing sub-modules
* **Payroll Module:** It handles all employee and payroll information
* **Work Order Module:** It keeps track of the car maintenance history and all work orders for maintenance done on a car
* **Customer Module:** It keeps track of the billing of the work orders to the customers and of the payments received from those customers

****

Based on the nature of the business, the best way to produce this conceptual database design would be to use a centralized and top-down approach. The designer must keep the design sufficiently flexible to make sure that it can accommodate any future integration of this system with the other service stations in the state.

1. **Suppose you have been asked to create an information system for a manufacturing plant that produces nuts and bolts of many shapes, sizes, and functions. What questions would you ask, and how would the answers to those questions affect the database design? Are some questions relevant only to the database design, in other words what are the answers to questions that help the end users but are irrelevant to the database design?**

Any information that is collected during the initial study and all subsequent phases will have an impact on the database design. The information is collected to establish the entities, attributes, and the relationships among the entities. Specifically, the relationships, connectivity’s, and cardinalities are shaped by the business rules that are derived from the information collected by the designer

**Questions:**

* How you want to track these products and what tools will be integrated?
* How will you assess various design processes like top-down vs bottom-up, centralized vs decentralized, system scope and boundaries?
* Do you want to develop Inventory, production, shipping, billing, etc.? Modules separately or all together?
* What are the latest technologies used for establishing the database with more security and following more advanced techniques to minimize the data anomalies?
* Do you want to keep track of the nuts and bolts by lot number, production shift, type, and department?
* What are the database designs will be affected by these changes?
* What type of databases are used and how would you like to scale, balance, and automate the processes?
* How you analyze, monitor, track the inventory and other materials?
* Do you want to keep track of the customers who received the batches of nuts and bolts?
* How reports are managed and generated. What are the ways to report the data fetched?

**The answers to these questions affect the conceptual and logical database design, the database’s implementation, it’s testing, and its subsequent operation. Not all problems come from databases, few triggers from the end-users, external scenarios, and other key aspects.**

**Forum: Week 5 Discussion:**

1. **First explain the statement, a transaction is a logical unit of work. Then, list and discuss the five transaction properties.**

**Transaction:**

In the database terms, a transaction is any action that reads from and/or writes to a database. A transaction may consist of a simple SELECT statement to generate a list of table contents; it may consist of a series of related UPDATE statements to change the values of attributes in various tables; it may consist of a series of INSERT statements to add rows to one or more tables, or it may consist of a combination of SELECT, UPDATE, and INSERT statements.

**A transaction is a logical unit of work...**

* It is a sequence of process that must be entirely completed or aborted
* Intermediate states are not acceptable in the process of transaction
* In DBMS, transaction act as a unit of logically related commands in which all the steps must be fully completed
* If any of the SQL command fails, the entire transaction is rolled back to the original database
* If the transaction is acceptable by the DBMS, the transaction changes the database from one consistent state to another
* To avoid inconsistent state, the DBMS must accept all database operations of the transaction are completed before the database committed
* **Example:** The database “sales” requires a minimum of two database operations. One is the creation of a product and another one is payment. If the first operation is completed, the database will be left in an inconsistent state and the entire database is canceled. The transaction is accepted when all two operations are completed in the database

**5 Transaction Properties:**

* **Atomicity:** All operations of a transaction must be completed, otherwise the transaction is aborted. A transaction is treated as a single, indivisible, a logical unit of work
* **Consistency:** Permanence of database's consistent state. A transaction takes a database from one consistent state to another consistent state. When a transaction is completed, the database must be in a consistent state; if any of the transaction parts violates an integrity constraint, the entire transaction is aborted
* **Isolation:** Data used during a transaction cannot be used by the second transaction until the first is completed. This property is particularly useful in multiuser database environments because several users can access and update the database at the same time
* **Durability:** Ensures that once transactions are committed, they cannot be undone or lost, even in the event of a system failure
* **Serializability:** Ensures that the schedule for the concurrent execution of several transactions should yield consistent results. This property is important in multiuser and distributed databases, where multiple transactions are likely to be executed concurrently

All five transaction properties work together to make sure that a database maintains data integrity and consistency for either a single-user or a multi-user DBMS.

1. **Your data warehousing project group is debating whether to create a prototype of a data warehouse before its implementation. The project group members are especially concerned about the need to acquire some data warehousing skills before implementing the enterprise-wide data warehouse. What would you recommend? Explain your recommendations.**

**Data warehouse:**

Data warehouse contains the collection of data that are used for decision making and business intelligence. It is a unique kind of database so it focuses on business intelligence, time variant data, and external data It is a subject-oriented, integrated, time-variant, and non-updateable data.

**Why prototype?**

* It demonstrates what is possible - both what can be done beyond the bounds of end-user thinking, and by the implication of what cannot be done
* It permits actual data-driven project scoping and budgeting
* It identifies the key performance indicators (KPIs), business metrics, dimensions, reports and queries to be identified in practice
* It does a far more thorough job to derive business value from the warehouse
* It socializes the design and scope of the project tangibly
* It builds and maintains project momentum
* It enables organizations to make and keep commitments to business users

**Recommendations:**

* Knowing that data warehousing requires time, money, and considerable managerial effort, many companies create data marts, instead
* Data marts use smaller, more manageable data sets that are targeted to fit the special needs of small groups within the organization
* In other words, data marts are small, single-subject data warehouse subsets. Datamart development and use costs are lower and the implementation time is shorter
* Once the data marts have demonstrated their ability to serve the DSS, they can be expanded to become data warehouses or they can be migrated into larger existing data warehouses

1. **A university has extensive dataset on its alumni, including past studies, demographic information by zip code, and past donations. The university is planning to send a deluxe brochure and a donation request to some of the alumni (Total targeting cost is $15 per alum) and has sufficient budget for constructing targeting models and running experiments under the following assumptions:**

* **Donation amount may vary**
* **Alumni may spontaneously make a donation (even when not targeted)**
* **Targeting cost is fixed (C= $15)**
* **Other than the targeting cost, there are no additional costs for alumni who are targeted and decide not to donate**

**Identify and build into a response the data mining model(s) that would suggest which alumni should be targeted. How might the expected value framework aid in determining which models should be used?**

**Data mining**:

* It is a process that employs automated tools to analyze data in a data warehouse and other sources
* Data mining tools are based on algorithms that form the building blocks for artificial intelligence, neural networks, inductive rules, and predicate logic
* The data mining tools will automatically search the data for anomalies and possible relationships, thereby identifying problems that have not yet been identified by the end-user
* In other words, data mining tools analyze the data, uncover problems or opportunities hidden in the data relationships, form computer models based on their findings and then use the model to predict business behavior without requiring end-user intervention
* Therefore, the end-user is able to use the system's findings to gain knowledge that may yield competitive advantages

**Neural Networks:**

* It’s a set of connected I/ O Units, where each connection has a weight associated with it
* It’s a case of Supervised, Inductive or Classification learning. Prediction accuracy is high and are robust
* It learns by adjusting the weights so as to be able to correctly classify the training data and hence, after testing phase, to classify unknown data

**Regression:**

* Regression is a data mining function that predicts a number. It begins with a dataset in which the target values are known
* These models are tested by measuring the difference between the predicted and expected values
* Age, weight, distance, temperature, income, or sales are predicted using regression techniques

**Decision Trees Models:**

* Decision trees (or a series of IF/THEN rules) as a commonly used machine learning algorithm are powerful and popular tools for classification and prediction
* These algorithms will try to identify important data clusters of features within a database
* It may be generated from past experience (data) and can be used to decide what to do according to weather conditions
* The more data available for training the decision tree, the more accurate its knowledge extraction will be, and thus, it will make more accurate decisions
* The deeper the tree, the more complex the rules and fitter the model
* Data mining tools incorporating machine-learning algorithms such as CART (classification and regression trees), CHAID (chi-squared automatic integration detection), ID3 (Interactive Dichotomizer) or C4.5 or C5.0 will segment a data set into statistically significant clusters of classes based on the desired output
* **Classification**: It is a two-step process, learning step and prediction step, in machine learning. In the learning step, the model is developed based on given training data. In the prediction step, the model is used to predict the response for given data (categorical class labels)
* **Prediction**: These predict continuous-valued functions
* **Root Node:** It represents the entire population or sample and this further gets divided into two or more homogeneous sets
* **Splitting**: It is a process of dividing a node into two or more sub-nodes
* **Decision Node:** When a sub-node splits into further sub-nodes, then it is called the decision node
* **Leaf / Terminal Node:** Nodes do not split is called Leaf or Terminal node
* **Pruning:** When we remove sub-nodes of a decision node, this process is called pruning. It’s the opposite process of splitting
* **Branch / Sub-Tree:** A subsection of the entire tree is called branch or sub-tree
* **Parent and Child Node:** A node, which is divided into sub-nodes is called a parent node of sub-nodes whereas sub-nodes are the child of a parent node
* Each node in the tree acts as a test case for some attribute, and each edge descending from that node corresponds to one of the possible answers to the test case. This process is recursive in nature and is repeated for every subtree rooted at the new nodes

**Advantages of Decision Trees:**

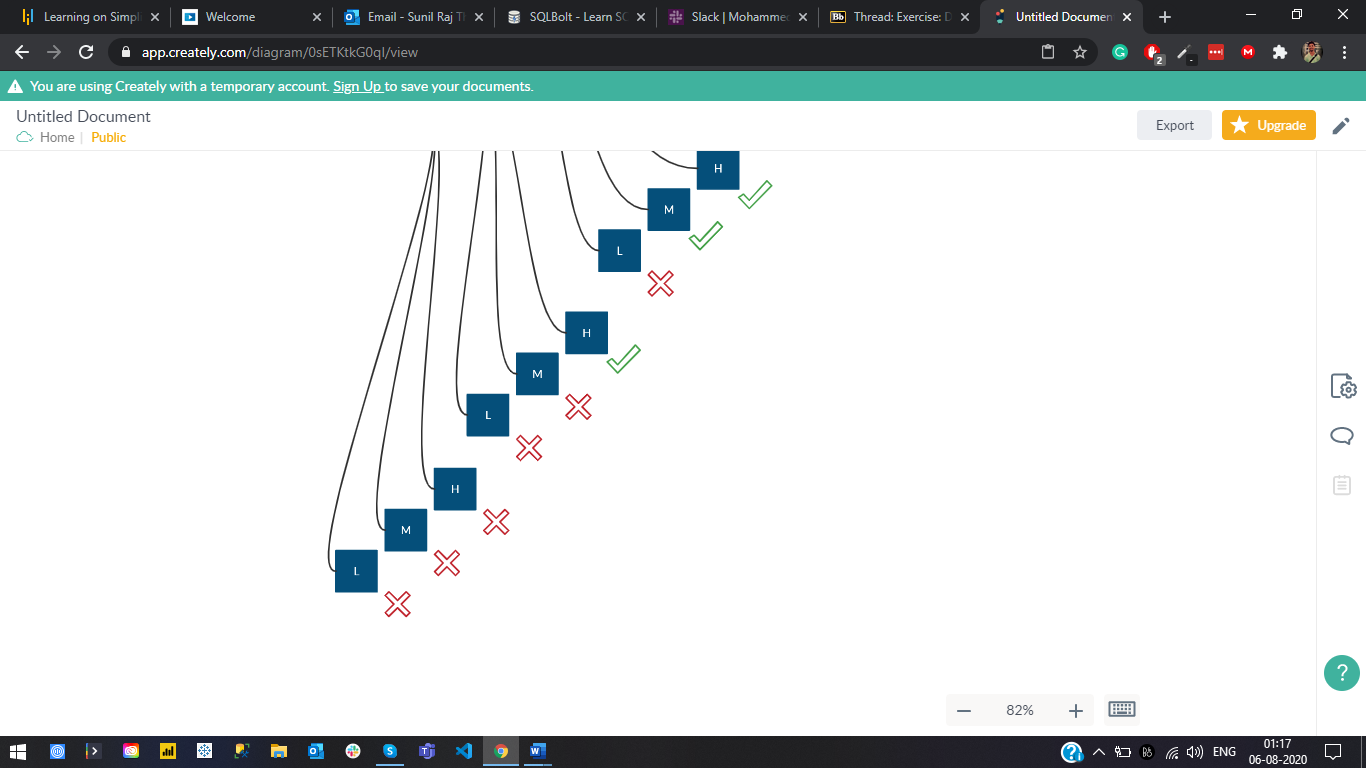
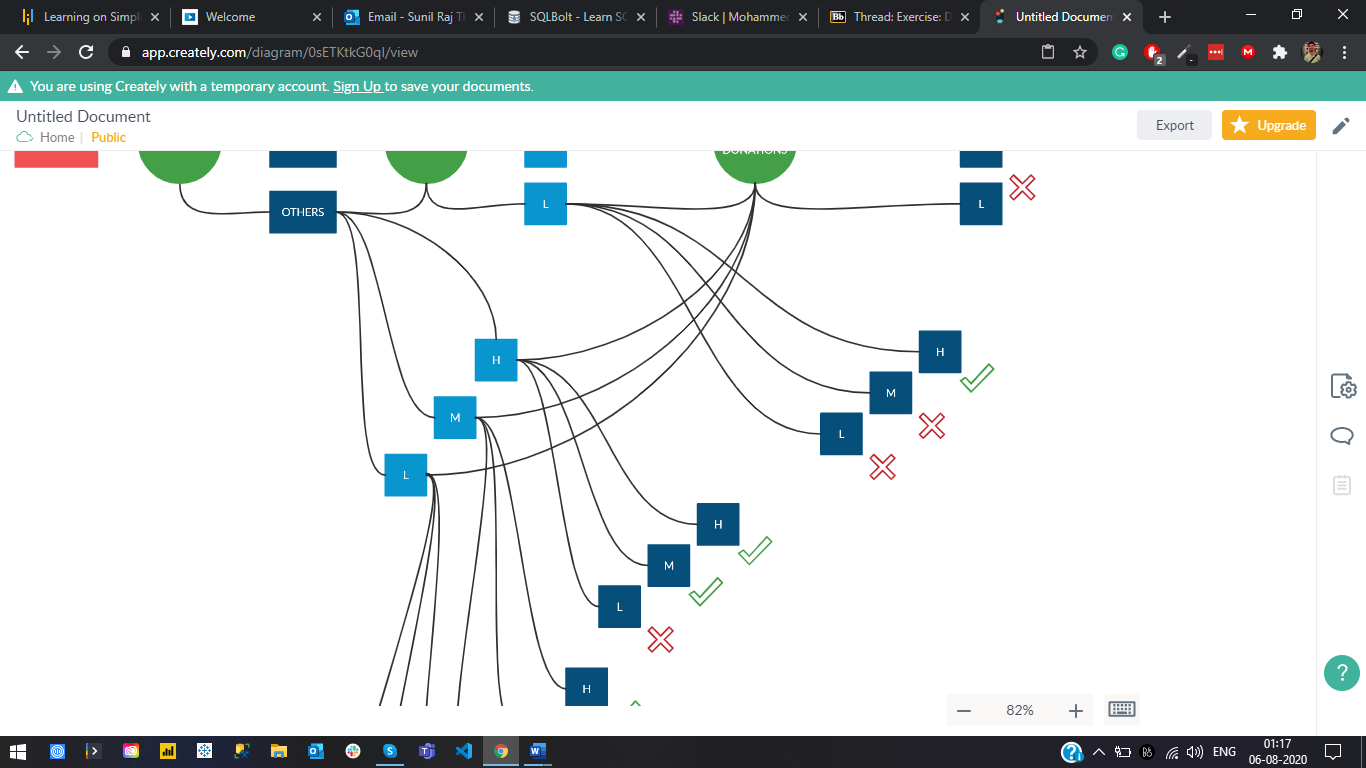
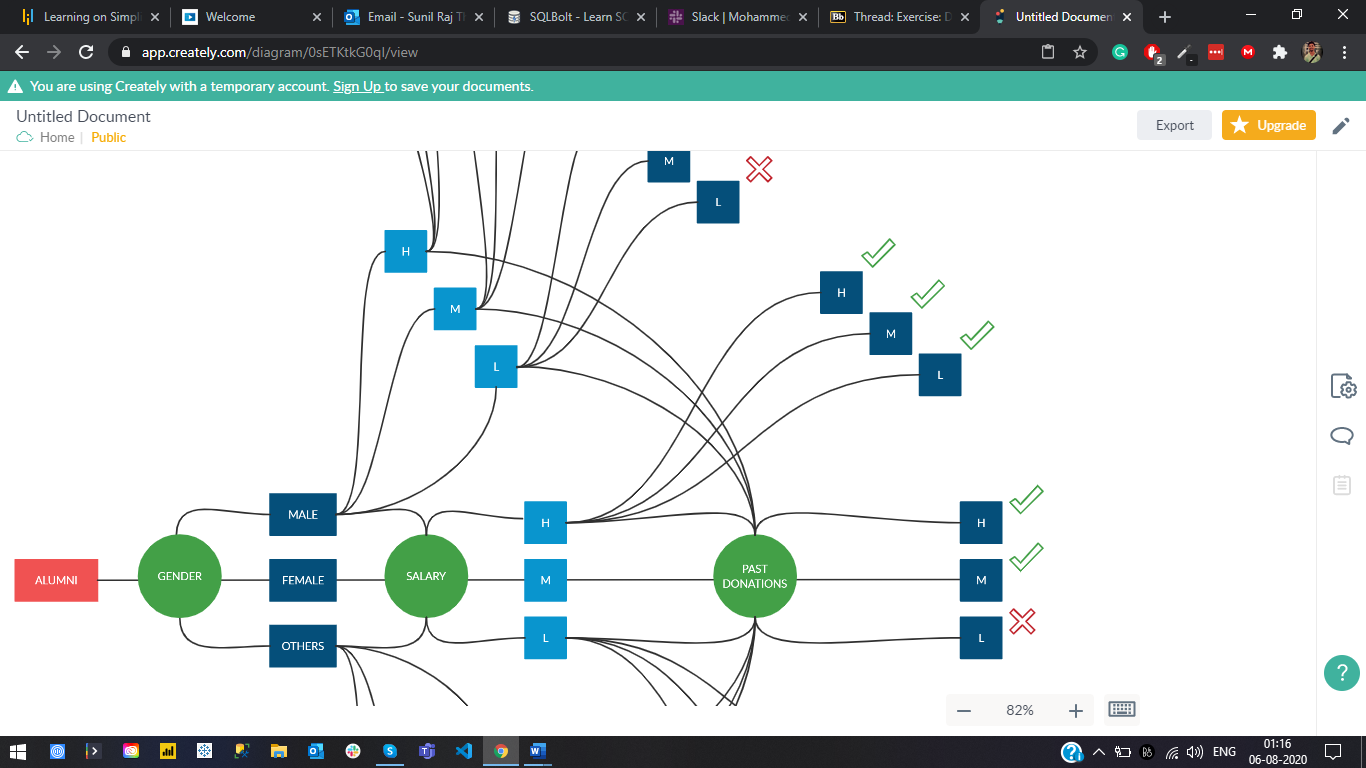
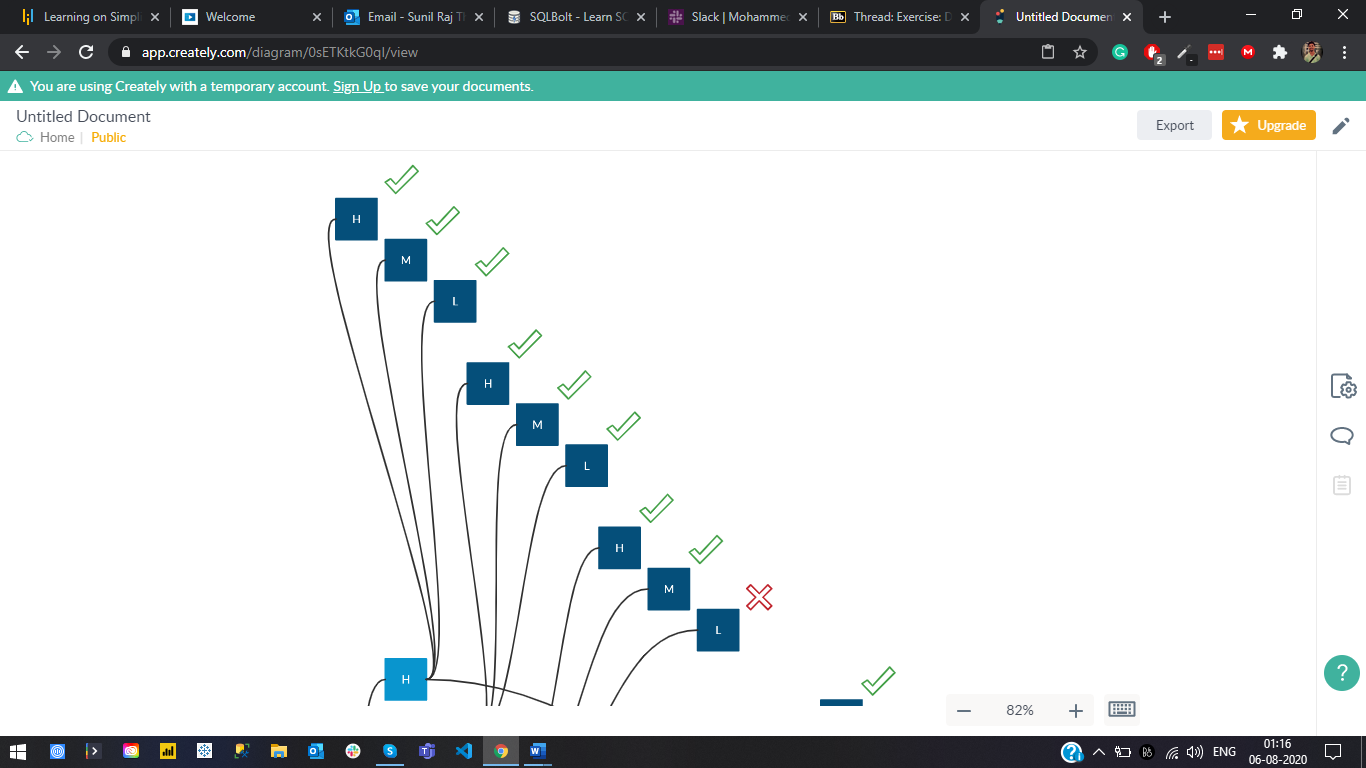
* Easy to use and understand
* Can handle both categorical and numerical data
* Resistant to outliers, hence require little data preprocessing
* New features can be easily added
* Can be used to build larger classifiers by using ensemble methods

**Disadvantages of Decision Trees:**

* Prone to overfitting
* Require some kind of measurement as to how well they are doing
* Need to be careful with parameter tuning
* Can create biased learned trees if some classes dominate

**Response:**

* So as per the given example the University already has ample data of its alumni and we have a couple of attributes, to build a model. We can have ‘Past donation’ as the Target variable. Zip code can be used to estimate the regional average salary expectations and other variables to consider
* For drawing model responses and making connections between the variables I had restricted a few variables that can be considered for better outputs. Those are Age, Marital Status, City, Country, Occupation, Country, Schools-Business, Engineering, Law, Humanities, Journalism, Arts, Science, Alumni Education Level – UG/ PG/ Ph. D., and Payment methods - Cash, Card, Cheque
* I have chosen these variables for the design purpose: Alumni, Gender (Male, Female, Others), Salary (High, Medium, Low), Past Donations (High, Medium, Low). Based on these attributes I have concluded few decisions made by the Alumni to donate the money which was indicated by a **Tick Mark (Green Color),** if not, **Cross Mark (Red Color)**
* With the help of these models the University can determine the donation amount that can be collected from the Alumni members by considering all the datasets related to them



1. **One might say an organization's overall readiness for a data warehouse initiative or even strategy reflects the business value that is being placed on fact-based design-making and reliance on the results from business analysis. After all, a data warehouse is a mechanism to better leverage data and support sophisticated analyses. For the organization that does not place a lot of value on analysis, this limits their ability to participate and contribute to their DW initiatives and thus lesson the success. Certain industries have been upfront developing performance-based techniques and one such technique is noted in the article shared in the Instructor Perspective area. From this, and the week's readings describe what is industry-defined as data warehousing. What are some the benefits, and barriers to data warehousing? Are businesses today placing enough value and resources on the implementation of data warehousing? Be sure to support your responses.**

**Business intelligence:**

It is a term used to describe a comprehensive, cohesive, and integrated set of applications used to capture, collect, integrate, store, and analyze data with the purpose of generating and presenting information used to support business decision making. As the names imply, BI is about creating intelligence about a business. This intelligence is based on learning and understanding the facts about a business environment. BI is a framework that allows a business to transform data into information, information into knowledge, and knowledge into wisdom.

**Data Warehouse:**

A data warehouse is a centralized store of all data generated by the departments of a large organization. It is specially designed for data analysis, generating reports, and other ad-hoc queries. A data warehouse extracts the huge streams of data from a company’s operational and external databases and turns them into meaningful data, so business decisions can be made based on this information.

**Benefits:**

* Better decision-making
* Quick and easy access to data
* Data quality and consistency
* Accessible and Easy to integrate
* Enhanced BI and High ROI
* Secure
* Dashboards, Visualizations, ETL, OLAP, and Data Mining
* Real-time Analysis and provides Reporting and Forecasting
* Competitive Advantage

**Barriers:**

* Not an ideal option for unstructured data
* Time-consuming and requires a lot of resources to get trained
* Difficult to make changes in data types and ranges, data source schema, indexes, and queries
* The Project scope will always increase

**Analysis:**

* A goal common to all businesses is to make better business decisions than their competitors
* To get the most out of your data teams, companies should define their objectives before beginning their analysis
* Speed, flexibility and foresight are the primary characteristics that distinguish successful organizations in this information age
* Performing an effective BI readiness assessment is a critical element of business-centric BI development methods because it helps “design” ROI into BI initiatives from the outset
* Businesses today must have management processes in place to monitor and control the organization, while at the same time decentralizing decision making in order to react to competitive changes and take advantage of unexpected opportunities
* Data warehouses store credible facts and statistics and decision-makers will be able to retrieve that information from the data warehouse based on their personal needs
* It can also assist in marketing segmentation, inventory management, financial management, and sales, etc.,
* Since data warehouses gather information from different sources and convert it into a single and widely used format, departments will produce results that are in line and consistent with each other
* Verifying data and ensuring you are tracking the right metrics can help you step out of your decision patterns

**Forum: Week 6 Discussion:**

1. **The integration between Web servers hosting the first Web sites and database management systems transformed how Web pages were designed, rendered, and operationalized. If not for this integration, we would not have e-commerce or search engines. While I can remember the static WWW, its true adaptation by business came once they get and retrieval processing was returning dynamic data over text. The text depending on the version covers a number of examples of Web application servers, for this discussion topic you are asking to go further.**

**Research an example of a Web application server. Explain the base characteristics of what, why, and how but do so from a database perspective. What purpose do these have today for businesses?**

**Server:**

It is a device or a computer program that accepts and responds to the request made by another program, known as the client. It is used to manage the network resources and for running the program or software that provides services.

There are two types of servers:

* **Web Server:** 
  + It contains only web or servlet container
  + In general web server can be used to host the web sites but there also used some other web servers also such as FTP, email, storage, gaming, etc.
  + It can be used for the servlet, JSP, struts, and JSF
  + The Web server (HTTP server) can invoke a variety of scripts and services to query databases and perform business processing and application servers often come with their own HTTP server which delivers Web pages to the browser
  + **Examples:** Apache Tomcat, Resin
* **Application Server:** 
  + It contains Web and EJB containers. It can be used for servlet, JSP, struts, JSF, EJB, etc.
  + It is a component-based product that lies in the middle-tier of a server-centric architecture
  + It provides the middleware services for state maintenance and security, along with persistence and data access
  + It is a type of server designed to install, operate and host-associated services and applications for the IT services, end-users, and organizations
  + **Examples:** JBoss, Glassfish, Oracle WebLogic, IBM WebSphere
  + A computer that executes commands requested by a Web server to fetch data from databases. Also known as app server
  + There is an overlap between an application server and a Web server, as both can perform similar tasks

**Web Application:**

* A web application is a computer program that utilizes web browsers and web technology to perform tasks over the Internet
* Web applications use a combination of server-side scripts (PHP and ASP) to handle the storage and retrieval of the information, and client-side scripts (JavaScript and HTML) to present information to users
* The web application requires a webserver to manage requests from the client, an application server to perform the tasks requested, and, sometimes, a database to store the information
* Application server technology ranges from ASP.NET, ASP, and ColdFusion, to PHP and JSP
* **Flow:**
* The **User** triggers a request to the web server over the Internet, either through a web browser or the application’s user interface
* **Web server** forwards this request to the appropriate web application server
* **Web application server** performs the requested task – such as querying the database or processing the data – then generates the results of the requested data
* **Web application server** sends results to the webserver with the requested information or processed data
* The **Web server** responds back to the client with the requested information that then appears on the user’s display

**Characteristics of various Web Application Servers:**

|  |  |  |
| --- | --- | --- |
| Apache Tomcat | IBM WebSphere | Adobe ColdFusion |
| It is a long-lived, open-source Java Servlet container, or web container, that provides the extended functionality to interact with Java Servlets, JSP, Java EL and WebSocket  It can also be configured to run multiple web applications on different ports and also cross-platform compatible  It consumes less in terms of memory and resource utilization, thus allowing the application to run smoothly on most systems without specific system requirements  Studies have claimed that >60% of java applications make use of apache tomcat  It brings simplicity, flexibility, performance, and value to the table  Tomcat supports SSL (Secure Socket Layer | IBM WebSphere Application Server is IBM’s answer to the JEE application server  It combines the worlds of Web servers, databases, servlets, and other Java standards to create a platform for creating and managing enterprise-class Web applications  The result is that customers can take full advantage of this without sacrificing manageability, reliability, scalability, performance, and security  The WebSphere Application Server security model is based on the services provided in the operating system and the Java EE security model  A company can choose to run just the application server and a few tools or it can buy WebSphere suites to address a particular need | It is a rapid development platform for building modern web applications  It is designed to be expressive to perform programming tasks at a higher level than most other languages  The ColdFusion platform is built on Java and uses the Apache Tomcat J2EE container  One can interact with the user-friendly ColdFusion Mark-up Language (CFML) to write programs  Develop websites as distinct parts or modules that can be saved in a database after which can be reassembled or incorporate for applications or functions at further stages  Can develop both web applications and mobile apps. Also, used to generate SOAP web services or Flash Remoting |

1. **Explain how and why database management systems have become the database management standard in organizations. Consider the advantages of the database approach over a file-system approach?**

**File System:**

A file management system is a DBMS that allows access to single files or tables at a time. In a File System, data is directly stored in a set of files. It contains flat files that have no relation to other files (when only one table is stored in single file, then this file is known as flat file)

**Disadvantages of the File system:**

* Data redundancy
* Data inconsistency
* Data Isolation
* Dependency on application programs
* Atomicity issues
* Data Security

**DBMS in Organizations:**

* Organizations use large amounts of data. A database management system (DBMS) is a software tool that makes it possible to organize data in a database. Used to maintain and control the database and its main purpose is to construct the database for multiple users and applications. This system permits us to flexibly access the data.
* The ultimate purpose of a DBMS is to store and transform data into information to support making decisions. Within an organization, the development of the database is typically controlled by database administrators (DBAs) and other specialists.
* This ensures the database structure is efficient and reliable. Proper database management systems help increase organizational accessibility to data, which in turn helps the end-users share the data quickly and effectively across the organization.
* A management system helps get quick solutions to database queries, thus making data access faster and more accurate. End-users like salespeople will have enhanced access to the data, enabling a faster sales cycle and a sounder decision making.
* With better data, management systems, and procedures come higher-quality information. A database management system helps in providing a framework to facilitate data quality initiatives. In turn, higher quality information helps in making better, faster decisions in an organization

A DBMS consists of the following three elements:

* **Physical database:** The collection of files that contain the data
* **Database engine:** The software that makes it possible to access and modify the contents of the database
* **Database scheme:** The specification of the logical structure of the data stored in the database

It organizes your files to give you more control over your data. A DBMS makes it possible for users to create, edit, and update data in database files. Once created, the DBMS makes it possible to store and retrieve data from those database files.

**Advantage of DBMS over the file system:**

There are several advantages of DBMS over the file system. Few of them are as follows:

* Concurrency
* Security
* Backup and Recovery
* Integrity
* Data Descriptions
* No redundant data
* Data Consistency
* Data Privacy
* Data Sharing
* Flexible
* Data Searching
* Performance

1. **Watch the video** [**Q&A - Database Administrator – What they do**](https://www.youtube.com/watch?v=0LFa8DH_6f4)

* **While viewing the presentation, take notes on the video focusing on the different tasks of the data base administrator (dba)**
* **Which are the most important tasks/roles?**
* **What are the ramifications to the business if security is not kept up-to-date?**

**DBA: What they do?**

Database administrators (DBAs) use specialized software to store and organize data. The role may include capacity planning, installation, configuration, database design, migration, performance monitoring, security, troubleshooting, as well as backup and data recovery. Also, to ensure that data is available, protected from loss and corruption, and easily accessible as needed. Below are some of the chief responsibilities that make up the day-to-day work of a DBA

**A database administrator's responsibilities can include the following tasks/ roles:**

* Installing and upgrading the database server and application tool
* Allocating system storage and planning storage requirements for the database system
* Modifying the database structure, as necessary, from information given by application developers
* Enrolling users and maintaining system security
* Data Extraction, Transformation, and Loading
* Ensuring compliance with database vendor license agreement
* Controlling and monitoring user access to the database
* Monitoring and optimizing the performance of the database
* Specialized Data Handling
* Planning for backup and recovery of database information
* Maintaining archived data
* Identifying user needs to create and administer databases
* Ensuring that the database operates efficiently and without error
* Be able to provide a strategic database direction for the organization
* Making and testing modifications to the database structure when needed
* Maintaining the database and updating permissions
* Backing up and restoring databases
* Experience and knowledge in migrating code, database changes, data and menus through the various stages of the development life cycle
* Merging old databases into new ones
* Recommend and implement emerging database technologies
* Create automation for repeating database tasks
* Contacting database vendor for technical support
* Generating various reports by querying from the database as per need
* Managing and monitoring data replication
* Troubleshooting
* Acting as liaison with users
* Administrative DBA, Development DBA, Architect, Data Warehouse, OLAP DBA, and Application DBA are few roles that can be undertaken

**Ramifications to the businesses if security is not kept up-to-date:**

* Lost Data, Time, and Money
* System Failures
* Virus, Malware, and Injection Attacks
* Data Breaches
* Law, Compliances, and Regulatory Affairs
* Performance degradation
* Downtime and less availability
* Reduction in business opportunities
* Huge time for migrations

1. **Read and reflect on the article** [**The Best Database-as-a-Service Solutions for 2020**](https://www.pcmag.com/picks/the-best-database-as-a-service-solutions)**. Also, read the full review for each database. While reading the article and full reviews think about the pros and cons of the different databases, interfaces, ease of use, toolbox and the services each one provides**

* **Which database would you select for a mid-size business and why? What would be your second choice and why?**
* **For each choice take into account the pros, cons, ease of use, toolbox, services that are provided and pricing model. What are the competitors of each and what reason would you chose the one you did over the competition? Be specific in your answers**

**Database as a service (DBaaS):**

* It is the process of application owners paying an outside provider that launches and maintains a cloud database for storage, as opposed to having the application owners control the database themselves
* Payments are per-usage and application owners can access their application data as they please
* These databases provide the same functionality as a standard relational or non-relational database
* Database as a Service or DBaaS is a cloud-based system whereby an off-site service provider furnishes the hardware, software, and infrastructure required for businesses to run their database operations
* The DBaaS provider supplies both the system and the IT expertise needed to help business subscribers use the database, and make decisions on how to store, query, and analyze their information
* DBaaS is beneficial for companies that are trying to avoid the work of configuring, maintaining, and upgrading their own databases
* DBaaS lives in the overall realm of software as a service (SaaS), similar to the platform as a service (PaaS) and infrastructure as a service (IaaS), where all products are hosted as a service
* Other types of databases include relational database tools, NoSQL database tools, graph database tools and more

**DB Choices:**

|  |  |  |
| --- | --- | --- |
| Features | 1st Choice | 2nd Choice |
| Database Name | Amazon RDS | MongoDB Atlas |
| Why? | Great relational DB service  Intuitive to use  Provides a ton of functionality  They have CLI and other API's  Easy to automate processes  Top Tier Support System | It is a developer's database  Simple user interface  More automation  Flexibility and controls  Built-in replication  Zero lock-in |
| Best For… | Application Developers  Database Administrators  System Administrators  DevOps Engineers  Mid Markets | Startups  Industry-leading Organizations  Government Agencies  Developers  Small Businesses |
| Pros | Stable  Powerful  Provides plenty of controls  Very Secure  No hardware maintenance  Automated Patching  Simplified disaster recovery  Additional storage allocation  Automated Backups | DB Provisioning and Management  Open-source  Powerful scaling  Sharding  Serverless  Machine learning capabilities  Built-in data browser  Granular role-based access controls  Use it anywhere |
| Cons | Complex to set up and move data  Setup – Need professional help  Expensive  No scale-out for write workloads  Downtime vital for scaling  No root access to the server  Zero data loss is not guaranteed | It's NoSQL  Unsuitable for complex queries  Inapt for projects - HIPAA compliance  Poor index optimization  Application crashes  Cluster Limitations  Connections Limits |
| Product Features | Access Controls/Permissions  Backup and Recovery  Data Migration  Data Replication  Mobile Access  Multi-Programming Languages Performance Analysis  Relational  Virtualization | Access Controls/Permissions  Backup and Recovery  Data Migration  Data Replication  Database Conversion  Multi-Programming Languages Performance Analysis  Relational  Virtualization |
| Ease of Use | 88/100 | 92/100 |
| Pricing Model | Free up to 750 Hours  Pay-as-you-go  Yearly Commitments | Free Forever  Essential Tier @ $9 per month  Professional Tier @ $25 per month |
| Alternatives | Azure SQL Database | Amazon DynamoDB |
| Reasons | Can scale from Mid Markets to Enterprise levels  The Windows Azure management portal itself  The ease at which services can be provisioned  Easily available services  The value for money is incredible | Can scale from Mid Markets to Enterprise levels  It’s hard to beat DynamoDB  Data insertion from the CLI or other libs  Pricing is very simple  More throughput - More pay  Automated backups  Easy Server management |