**1. Describe stored procedures:**

a. Provide context to where they are used and by whom.

b. Explain what they are as a database object.

c. Explain why and how they are valuable and essential to database applications.

d. Provide an example of a stored procedure being coded and executed.

Stored procedures are wonderful ways to automate processes. We do it on the database layer and almost every single application gives us a lot of flexibility and the ability to do repetitive code.

It is coded by database developers and used by developers and users when they want to insert or update something in the database.

Stored procedures are compiled code which means they are converted into machine language. It’s in the memory and ready to go. So we can execute the code very quickly and repeat it over millions of times an hour. It also had an execution plan which means we already know the directions that we want to go.

Stored procedure can be beneficial in terms of executing repetitive code to update or insert some rows in the database. It was written by professionals and tested, so it runs at a 100% accuracy. It offers a layer of abstraction, people have to know its existence to call it. People also need permission to execute it. When we have pre-written code, we can avoid the catastrophe created by human error, so we don’t need to do error handling to troubleshoot random errors.

As a result, stored procedure has simplified the environment and protected against rouge attacks.

We can speed up the process between an application and the data layer because we don't have to connect to the database, we can take to the middle tier and make a call the call rpc road procedure code. We pass in what we know (name) to find what we don’t know(ID), so we don’t have to memorize all the foreign keys and ID values. Thus, there is a middle layer for us to call the stored procedure and speed up the process.

Example:

CREATE PROCEDURE …

@name VARCHAR(60),

@DOB DATE,

…

AS

DECLARE @ID INT

SET @ID = (SELECT FROM )

BEGIN TRANSACTION T1

INSERT INTO ()

VALUES()

COMMIT TRANSACTION T1

GO

**2. Describe user-defined functions:**

a. Provide context to where they are used and by whom.

b. Explain what they are as a database object.

c. Explain why and how they are valuable and essential to database applications.

d. Provide an example of a user-defined function being coded and executed.

User-defined functions are wonderful ways to automate processes such as Business rule and Computed Columns. Business can enforce their rule by defining a user-defined function on the database layer and repeatedly check constraint when new data comes in. They can also analyze data by applying Computed Columns to calculate, for example, the total sales of a product.

The code is written by professionals and tested. Therefore, we don’t have to do error handling and troubleshoot random errors due to human errors.

It offers a layer of abstraction, people have to know its existence to call it. People also need permission to execute it.

User-defined functions are compiled code which means they are converted into machine language. It’s in the memory and ready to go. So we can execute the code very quickly and repeat it over millions of time.

It does SELECT instead of INSERT UPDATE. It answers yes or no, how many, and when was the last time something happened, but they do not change data.

Example:

CREATE FUNCTION …(@PK INT)

RETURNS INT

AS

BEGIN

SET @ RET INT = (SELECT COUNT(\*)

FROM

WHERE ID = @PK)

RETURN @ RET

END

GO

ALTER TABLE …

ADD … AS (dbo. …(ID))

GO

**3. Discuss the purpose and benefits of following the normalization process during database design.**

a. Provide context to what normalization is used and by whom.

b. Explain what normalization is as a database process.

c. Explain why and how normalization is valuable and essential to database design.

d. Provide an example of an entity being normalized (before and after).

Normalization is the central core of relational database theory. We have to get a structure of the database into a shape that allows us to very quickly write data and maintain the integrity of every transaction. It also allows us to answer basic questions about the operations of the business.

It is used by the database designers and developers. It’s in the logical design phase of SDLC.

When we are done with the conceptual design phase, we are able to understand the core component of the business.

Normalization is a process for us to iron out the wrinkles. During the design of the diagram, we have discovered every attribute that is significant to the operations of business. Normalization is to make sure every one of those attributes are in the right table. By eliminating multivalued attributes, domain violation, and row dependencies, it’s 1NF. By eliminating partial dependencies and transitive dependencies, we have reached 3NF. By creating bridge tables to track many to many relationship, the database has reached 4NF.

When we need to write in a normalized database, we are optimizing the writes. How? Because we are pushing everything into its own entity and reference each other by foreign keys. It makes the writing process a lot faster and less complicated. The brilliance of relational theory is foreign keys and we don't have to pack all of this data into a single entity. If we want to find information across the entities, we just need to reference the foreign key.

For example, if we are keeping track of a student in college, we need to keep track of which dorm he had lived in. We may also want to know the year built of a building. If we are tracking the building by student’s name, Year build is out of place in the student table. Therefore, we need to go through the process of normalization to recognize when we have data that does not directly relate to the object, so we need to move it to a new entity. In this case, Year built entirely depends on the dorm name. It has nothing to do with the student. So we need to create a new entity for dorm to keep track of Year built and use foreign keys to reference student.

**4. Discuss the purpose and benefits of following the Systems Development Life Cycle (SDLC) framework**

during the construction of any software application.

a. Provide context to what the SDLC is, how it is used and by whom.

b. Explain what the SDLC is as a design process and distinguish it from a methodology.

c. Explain why and how the SDLC is valuable and essential to software development.

d. Provide an example of an SDLC scenario with reference to phases and methodology choice.

Every project to build a database includes a systematic approach which allows us to identify where resources are available to us and where we are in our design process.

It is used by project managers and the development team during the project.

The fives phases are: Planning, Analysis, Design, Building, and Maintenance.

SDLC also had different methodologies such as Agile, Waterfall, Lean… Why do we have different methodologies? Because there are different conditions in different industries. For example, if you are constructing a building, you cannot operate it until it’s been done. So you need to follow Waterfall to complete the construction. For software development, you probably want to add features every two weeks so we are building it as we are using it. That does not apply to construction or manufacturing. Therefore, we need different methods of delivery.

SDLC is a critical systems development life cycle for building every project from scratch to a complete project. Why is it critical? Because it allows us to know what are our concerns, where are we in the project, and how is it going to be used. For example, if we want to build a project, we need to know exactly what our stakeholders need, so that we don’t build the wrong thing. By having a systematic process, we go through iterations of the same phases and we can begin to predict the costs. In addition, we can predict situations well before we actually have one. We can dynamically reallocate resources. We are no longer affected by emergencies. We are better able to recover. When we follow a systematic process, we have a higher probability of building something that can provide value.

An example of an SDLC scenario: Software development uses Agile. Construction uses Waterfall. Manufacturing uses Lean.

**5. Explain the concept of high availability (‘HA’).**

a. Provide context to ‘high availability’ in relation to data management.

b. Explain what terms are used to measure the effectiveness of HA.

c. Explain why and how HA is valuable and essential to database operations.

d. Provide an example of a database being that is a prime candidate for HA.

High availability is an important part of any production database environment where uptime and application connection is critical to the mission and the revenue of organizations. By saying production, we mean that it’s live and customer facing data. Uptime is measured in a percentage. The target of uptime is 99.999%. We cannot achieve high availability without using synchronous transfer of data. When we use synchronous transfer, we have automatic failover. Using 2 phase commit (Diagram server A and server B and a Web 1. Write from web to server A. 2. Mirroring from A to B 3.Confirm received 4. Joint Commit) It slows down the ability to process transactions. Synchronous transfer means it has a conversation back and forth between the two servers. We do not accept the transaction if one server did not receive the communication, because we do not want to have two servers that are not identical. When we have synchronous servers, we can fail back and forth all day long. Therefore, we achieve automatic failover.

High availability is valuable and essential because we can avoid troubleshoot and poor customer experience. A high availability server allows customers to use the service at any time. Why is it important to invest money in high availability? Because having downtime for hours will cause a company to lose money. If a company makes 20million per hour, having 3 hours of downtime can cost 60 million dollars. Thus, it’s better to invest money to achieve high availability in the first place.

An example of a prime candidate for high availability is the drink sold in Starbucks. The company relies on selling drinks to gain revenue. Therefore, selling drinks needs to be the prime candidate for HA.

**6. Explain the concept of scalability.**

a. Provide context to scalability in relation to data management.

b. Explain what terms are used to measure the effectiveness of scalability.

c. Explain why and how scalability is valuable and essential to database operations.

d. Provide an example of a database being that is a prime candidate for scalability.

Scalability is the measurement of a an application or an infrastructure that is able to meet the dynamic demands of a user base. We measure in throughput which includes measuring the number of transactions per minute and the number of simultaneous connections. For example, for NewYork Times, we need to measure the number of simultaneous connections because customers just connect to the server. Amazon needs to measure the number of transactions per minute because the customers are sending transactions every second.

There are two forms of scalability - Scale up and Scale out. Scale up means we are adding more hardwares and capabilities to a single node. When we scale out, we are adding more nodes. When we are dealing with relational database, we only have two nodes so we can only do scale up. When we are dealing with NoSQL, we can applu scale out to it. The difference between scale up and scale out is that scale up is high maintenance, manual improvement. Scale out is low maintenance, no people involved, and dynamic growth.

Scalability is valuable and essential. Why? Because having scalability allows more people to connect to the applications. For example, if there are one million people want to connect to the applications but only 1000 got through, it will affect the revenue generated. If the application crashes due to not enough scalability, we now don’t even have availabiliy. Therefore, we must be able to scale to meet the demand of the business.

An example of a prime candidate for Scalability is the connection to server of a video game. A game needs to have scalability to allow millions of players to connect to it at the same time.

**7. Explain the concept and best use of super-type/sub-type (‘Enhanced ERD’).**

a. Provide context to type inheritance in relation to data management.

b. Explain what terms are used in the discovery/implementation of type inheritance.

c. Explain why and how type inheritance is valuable and essential to database design.

d. Provide an example of a database design with super-type/sub-type (before & after).

The basic concept between super-type(superclass) and sub-type(subclass) is called attribute inheritance. It means we can take a general object and specify deeper and gather only the relevant attributes.

Superclass: An entity type that includes one or more distinct sub-groupings of its occurrence.

Subclass: Distinct sub-grouping of occurrences of an entity type

Specialization and Generalization

Specialization is a process of maximizing differences between members of an entity by identifying their distinguishing characteristics

Generalization is a process of minimizing differences between entities by identifying their common characteristics.

Rules:

Superclass and subclass relationships are one-to-one. We borrow the key from the superclass and it becomes the key of the subclass.

Example: musician PK -> Instrument with musician’s PK

Superclass may contain overlapping or distinct subclasses

Example: A musician can play multiple instruments but they can also play only one instrument.

Not all members of superclass need to be a member of subclass

Constraints on Specialization and Generalization:

Participation joint: It determines whether every member in superclass must participate as a member of a subclass (Mandatory or optional)

Disjoint: It describes the relationship between members of the subclasses. Indicates whether a member of the superclass can be in one or more subclasses. (disjoint or nondisjoint)

Type inheritance is valuable because it simplifies the process for people who are writing reports based on the database and for the people who are building the application by eliminating null value in a table. It can also significantly improve the performance and the development time it takes to build a database.

**8. Explain what the instructor means by the phrase ‘organize or die’.**

a. Provide context to ‘organize or die’ in relation to human existence and evolution.

b. Explain what ‘organize or die’ means in relation to current economic systems.

c. Explain why and how ‘organize or die’ is valuable and essential to businesses.

d. Provide an example of a company having success following principles of ‘organize or die’.

INFO 330: Final Exam Study Guide

**9. Compare and contrast relational database systems with NoSQL systems as presented in lecture.**

a. Provide context to these systems in relation to data management.

b. Explain what the determining factors are in deciding which design is preferred.

c. Explain why and how each system is valuable and essential to database operations.

d. Provide an example of a database in context of a business scenario that would be best

implemented with each design.

**10. Explain the differences between OLTP and OLAP systems as presented in lecture.**

a. Provide context to these systems in relation to data management.

b. Explain what the determining factors are in deciding which design is preferred.

c. Explain why and how each system is valuable and essential to database operations.

d. Provide an example of a database in context of a business scenario that would be best

implemented with each design.

OLTP: Online Transactional Processing

OLAP: Online Analytical Processing

OLTP and OLAP systems are both prominent active and important in almost every single solitary company.

OLTP is for operational, immediate relevance. Short shelf life. It has millions of tiny transactions. Live Data Relational

OLAP is a strategic, future planning, long shelf life. It has a single uploads that are massive. Dead Data (historical) Dimensional.

When we are doing analysis or evaluating, we want to choose OLAP becasue it has big vision. It’s built for analyzing data and make important business decisions.

When we are trying to deliver a package, we prefer OLTP because it focuses on the immediate transactions and get it down.

We take the relational data from OLTP and transfer it to OLAP and the transferring process is called ETL (extract, transform, load). The analytical data is mostly made up of the relational data.

OLTP is valuable because it’s good for a start up company. It helps the company to figure out how to deliver a package? How to put a box on a truck.

OLAP is valuable because it allows a mature company to compete against other companies by analysing their data.

Amazon example: deliver a package is OLTP. Decide which department is earning money and should be consolidated and eliminated is OLAP.

11. Explain the purpose and steps to database maintenance as presented in lecture.

a. Provide context to maintenance in relation to data management.

b. Explain what the determining factors are in establishing a set of maintenance tasks.

c. Explain why and how maintenance and planning for any disaster is critical for business survival.

d. Provide an example of proper maintenance from lecture.

**12. Explain the purpose and best design choices to data visualization as presented in lecture.**

a. Provide context to data visualization in relation to data management.

b. Explain what the determining factors are in establishing a dashboard of data visualizations

c. Explain why and how data visualizations are critical for business survival.

d. Provide an example of a ‘good’ data visualization from lecture and explain why it is good.

Data visualization is critical to explaining large volumes of data to a wide range of people because we are visually dominant in nature. Evidence: We are bipedal (walk with two legs). We are binocular (We see color). We have rapid fire visual distraction meaning that if something moves, it gets our attention.

Data visualization is critical because it allows us to communicate massive amounts of data in a short amount of time.

For example, we can spend only three seconds to figure out which county has the most COVID-19 cases.

Example of good visualization: Health.org

Accurate

Honest

Interactive

Provoking