|  |  |
| --- | --- |
| 21 | Clustered indexes sort and store the data rows in the table based on their key values, whereas non-clustered indexes store a pointer to the data rows. |
| 22 | SQL Server Agent automates and schedules tasks, making database management more efficient. |  |  |
| 23 | SSIS offers powerful data integration capabilities but can be complex and resource-intensive. |  |  |
| 24 | SSMS provides a user-friendly interface for managing databases, running queries, and configuring settings. |  |  |
| 25 | Foreign key constraints ensure that the data between related tables remains consistent. |  |  |
| 26 | Stored procedures encapsulate T-SQL code, providing reusability, security, and performance benefits. |  |  |
| 27 | SSRS supports comprehensive reporting capabilities, enhancing decision-making processes. |  |  |
| 28 | Indexes improve query performance by allowing faster data retrieval but can slow down data modification operations. |  |  |
| 29 | Transactions ensure data consistency by treating a sequence of operations as a single unit of work. |  |  |
| 30 | A well-defined backup and restore strategy protects data from loss and ensures business continuity. |  |  |
| 36 | Encapsulation,Inheritance,Polymorphism,Absraction |  |  |
| 37 | Interfaces define method signatures without any implementation, allowing multiple inheritance and facilitating loose coupling. Abstract classes can contain method implementations, fields, and constructors, supporting single inheritance and providing a base for subclasses to extend and specialize functionality. |  |  |
| 38 | Interfaces in C# define only method signatures that classes must implement, promoting loose coupling and multiple interface inheritance. Abstract classes can include method implementations and fields, supporting single class inheritance and providing a base for subclasses to extend and specialize functionality. |  |  |
| 39 | Exception handling in C# allows developers to manage and respond to runtime errors gracefully, contributing to robust software by preventing crashes and maintaining application stability |  |  |
| 40 | Classes in C# define the blueprint or template for objects, while objects are instances of classes that encapsulate data and behavior, forming the foundation of Object-Oriented Programming by promoting code reuse and modularity. |  |  |
| 41 | Method overloading in C# involves defining multiple methods with the same name but different parameters within the same class, while method overriding occurs when a derived class provides a specific implementation of a method already defined in its base class. |  |  |
| 42 | Constructors in C# are special methods used to initialize objects when they are created, differing from regular methods by having the same name as the class and no return type; types include default constructors, parameterized constructors, and static constructors. |  |  |
| 43 | Properties in C# provide controlled access to fields by encapsulating them with getters and setters, while fields are variables that directly store data in a class or struct. |  |  |
| 44 | Value types in C# store data directly, whereas reference types store references to the location of the data in memory. |  |  |
| 45 | Inheritance in C# allows a class to inherit properties and methods from another class, promoting code reusability by enabling the reuse of existing code and extending functionality as needed. |  |  |
| 66 | A namespace in C# is a way to organize and group related classes and types. It provides a hierarchical naming structure to prevent naming conflicts and improve code organization. |  |  |
| 67 | Value types store their actual value directly, whereas reference types store a reference to the location of the value. Value types are stored on the stack, while reference types are stored on the managed heap. |  |  |
| 68 | The "using" statement in C# is used for automatic disposal of resources. It ensures that IDisposable objects are properly disposed of when they are no longer needed, improving memory management and preventing resource leaks. |  |  |
| 69 | The "string" type in C# is immutable, meaning its value cannot be changed once it is created. "StringBuilder" is mutable and provides better performance when manipulating large strings frequently. |  |  |
| 70 | Delegates in C# are type-safe function pointers that allow methods to be passed as parameters or stored in variables. They enable event handling, callbacks, and asynchronous programming. |  |  |
| 71 | The "finally" block in a try-catch-finally statement is used to specify code that should be executed regardless of whether an exception is thrown or not. It ensures cleanup tasks or resource releases are performed. |  |  |
| 72 | "IEnumerable" is used for querying in-memory collections, while "IQueryable" is used for querying external data sources such as databases. "IQueryable" provides deferred execution and supports complex query operations. |  |  |
| 73 | The "var" keyword in C# allows the compiler to infer the type of a variable based on its initialization value. It provides syntactic sugar to reduce code verbosity when the type is evident. |  |  |
| 74 | Boxing is the process of converting a value type to the corresponding reference type (boxing it into an object). Unboxing is the reverse process of extracting the value type from the boxed object. |  |  |
| 75 | "readonly" variables can be assigned a value either at the time of declaration or within the constructor, and their value cannot be changed afterward. "const" variables are compile-time constants with a fixed value that cannot be modified. |  |  |
| 116 | The self keyword is used in instance methods to refer to the instance of the class that is being operated on.  It allows access to the attributes and methods of the class in object-oriented programming. |  |  |
| 117 | The list data type is an ordered collection of items that can be changed (mutable).  Lists are defined using square brackets, e.g., my\_list = [1, 2, 3]. |  |  |
| 118 | The result is "nohtyP". The slice notation [::-1] reverses the string. |  |  |
| 119 | The output will be False True. In Python, the integer 0 converts to False, and any non-zero number (like 3.14) converts to True. |  |  |
| 120 | The \_init\_ method is the constructor method in Python.  It is automatically called when an instance of the class is created and is used to initialize the instance variables |  |  |
| 121 | The pass statement is a null operation; it is used as a placeholder in loops, functions, classes, or conditionals where code will be added later. |  |  |
| 122 | for i in range(5) will raise a SyntaxError because it lacks a colon (:) at the end, which is required to indicate the start of the for-loop block. |  |  |
| 123 | The output will be [1, 2, 3, 4, 5]. Lists are mutable and are passed by reference. The variable y references the same list as x. Therefore, appending to y also changes x. |  |  |
| 124 | The str() function is used to convert an object to its string representation |  |  |
| 125 | The output will be 5. The function add has a default argument b set to 2. Calling add(3) passes 3 as a, and b remains the default value 2. Therefore, the function returns 3 + 2 = 5. |  |  |
| 146 | The <meta> tag is used to provide metadata about the HTML document, such as character set, page description, and keywords. |  |  |
| 147 | The DOCTYPE declaration defines the document type and version of HTML being used, which helps browsers to render the page correctly. |  |  |
| 148 | The <form> element creates an interactive form for user input, containing various input elements like text fields, checkboxes, and submit buttons. |  |  |
| 149 | The <head> element contains metadata, links to stylesheets, scripts, and other resources essential for the document is head section. |  |  |
| 150 | The <script> tag is used to embed or reference JavaScript code within an HTML document, enabling dynamic content and interactive features. |  |  |
| 151 | The <table> element structures data in a grid format using rows (<tr>) and cells (<td>), allowing for organized presentation of tabular data. |  |  |
| 152 | The href attribute specifies the URL of the page or resource that the hyperlink (<a> tag) points to. |  |  |
| 153 | The class attribute assigns a CSS class to an HTML element, enabling the application of specific styles or behaviors to elements with the same class. |  |  |
| 154 | The <section> element defines a thematic grouping of content, helping to organize the document into distinct, meaningful sections. |  |  |
| 155 | The <title> tag sets the title of the HTML document, which appears in the browser s title bar or tab, providing a brief description of the page content. |  |  |
| 176 | The flexbox layout in CSS3 is used to design a flexible and efficient layout structure for distributing space among items in a container. |  |  |
| 177 | The grid layout in CSS3 allows for two-dimensional layout control with rows and columns, whereas the flexbox layout is primarily one-dimensional. |  |  |
| 178 | Media queries in CSS3 are used to apply different styles based on the characteristics of the device, such as screen size and resolution. |  |  |
| 179 | The box-shadow property in CSS3 is used to add shadow effects to an element"s frame, enhancing its visual appearance. |  |  |
| 180 | The @keyframes rule in CSS3 defines the intermediate steps of a CSS animation, specifying how the animation progresses at various stages. |  |  |
| 181 | The transition property in CSS3 allows for smooth changes between property values, creating a more interactive and visually appealing user experience. |  |  |
| 182 | The border-radius property in CSS3 is used to create rounded corners on elements, contributing to a modern and softer design aesthetic. |  |  |
| 183 | The rgba color model in CSS3 specifies colors using red, green, blue, and alpha (opacity) values, allowing for semi-transparent elements. |  |  |
| 184 | The flex property in CSS3 flexbox layout specifies how a flex item will grow or shrink relative to the other items in the container. |  |  |
| 185 | The calc() function in CSS3 is used to perform calculations to determine CSS property values, allowing for more dynamic and responsive design adjustments. |  |  |
|  |  |  |  |