1. **What is EJS in the context of web development?**
   A) A database management system
   B) A JavaScript library for client-side templating
   C) A templating engine used for rendering HTML pages
   D) A server-side framework
   Answer: C) A templating engine used for rendering HTML pages
   Explanation: EJS (Embedded JavaScript) is a templating engine used to generate HTML markup with plain JavaScript. It is primarily used for rendering dynamic HTML pages on the server-side, often in Node.js environments.
2. **What is the purpose of EJS partials in a web application?**
   A) To store large amounts of data in a single file
   B) To break down a large template into smaller reusable components
   C) To style HTML elements using CSS
   D) To handle routing in the application
   Answer: B) To break down a large template into smaller reusable components
   Explanation: EJS partials allow you to reuse common HTML code, like headers, footers, and sidebars, across multiple pages. This promotes code reusability and makes the application easier to maintain.
3. **Which function is used to include an EJS partial in another EJS template?**
   A) `include()`
   B) `render()`
   C) `partials()`
   D) `insert()`
   Answer: A) `include()`
   Explanation: The `include()` function in EJS is used to include one template (partial) into another template. It can be used to insert common UI components like headers and footers.
4. **How do you pass data to an EJS partial when including it in a template?**
   A) By using the `data` attribute
   B) By defining the data in the partial itself
   C) By passing an object of data in the `include()` function
   D) By using the `locals` property
   Answer: C) By passing an object of data in the `include()` function
   Explanation: When including an EJS partial, you can pass data to it by providing an object of data as a second argument to the `include()`

function. For example: `<%- include('partials/header', { title: 'Home' }) %>`.

5. What is the syntax to include an EJS partial in another template?
   A) `<%- include('partial/header') %>`
   B) `<% include('partial/header') %>`
   C) `<%- partial('header') %>`
   D) `<% partial('header') %>`
   Answer: A) `<%- include('partial/header') %>`
   Explanation: The correct syntax to include an EJS partial is `<%- include('partial/header') %>`. The `<%-` syntax ensures that the content is rendered (as opposed to escaping HTML).
6. Which of the following EJS syntax is used to include a header partial into a page template?
   A) `<%- include('header.ejs') %>`
   B) `<%- include(header.ejs) %>`
   C) `<%- include('header') %>`
   D) `<% include('header') %>`
   Answer: A) `<%- include('header.ejs') %>`
   Explanation: The correct syntax for including a partial in EJS is `<%- include('header.ejs') %>`. You can also use just the file name (without the `.ejs` extension) if it's in the same directory.
7. What type of EJS include syntax should be used to include HTML content from a partial?
   A) `<% include('header') %>`
   B) `<%- include('header') %>`
   C) `<% include('%header') %>`
   D) `<%- include('%header') %>`
   Answer: B) `<%- include('header') %>`
   Explanation: The `<%-` syntax is used for including raw HTML content without escaping it. This is typically used to include HTML files or EJS partials.
8. How can you create a partial for a footer in EJS?
   A) By writing a footer HTML block directly in the main template
   B) By creating a separate `.ejs` file for the footer and including it in the main template
   C) By using the `render()` method to include the footer
   D) By writing a JavaScript function that renders the footer

Answer: B) By creating a separate `.ejs` file for the footer and including it in the main template
Explanation: To create a footer partial in EJS, you should create a separate `.ejs` file for the footer and use the `include()` function to insert it into the main template.

9. Can EJS partials be used to include JavaScript code in a template?
   A) Yes, by including a `.js` file directly in the partial
   B) No, EJS is only for HTML rendering
   C) Yes, EJS supports embedding JavaScript directly within the partials
   D) Yes, JavaScript can be passed as data to the partials
   Answer: C) Yes, EJS supports embedding JavaScript directly within the partials
   Explanation: EJS partials can include JavaScript code inside the templates, allowing dynamic content generation, such as loops, conditionals, or rendering data passed from the server.
10. Which of the following is a valid example of how to include a sidebar partial in a template?
    A) `<%- include('sidebar') %>`
    B) `<%- sidebar() %>`
    C) `<% include('sidebar') %>`
    D) `<% include('sidebar.ejs') %>`
    Answer: A) `<%- include('sidebar') %>`
    Explanation: To include a sidebar partial, you use the syntax `<%- include('sidebar') %>`. The `<%-` syntax ensures that the content of the partial is rendered, not escaped.
11. What is the default file extension for EJS partials?
    A) `.html`
    B) `.ejs`
    C) `.js`
    D) `.tpl`
    Answer: B) `.ejs`
    Explanation: The default file extension for EJS templates and partials is `.ejs`. However, you can omit the extension when including a partial, and EJS will automatically look for a file with the `.ejs` extension.
12. How would you pass dynamic data (e.g., `user.name`) to a header partial in EJS?
    A) `<%- include('header', { user: user }) %>`
    B) `<%- include('header', user) %>`

C) `<% include('header', user.name) %>`

D) `<% include('header') %>`

Answer: A) `<%- include('header', { user: user }) %>`
Explanation: You pass dynamic data to an EJS partial by including it as an object. The object `{ user: user }` passes the `user` data to the header partial.

13. What is the result of including the following in an EJS template? `<%- include('footer') %>`

    A) It will escape the HTML content of the `footer.ejs` partial.

    B) It will render the content of the `footer.ejs` partial as raw HTML.

    C) It will include the raw code of `footer.ejs` without rendering.

    D) It will execute JavaScript from `footer.ejs` directly.

    Answer: B) It will render the content of the `footer.ejs` partial as raw HTML.
    Explanation: The `<%-` syntax is used in EJS to render content from the included partial as raw HTML, without escaping it.

14. Can you pass multiple variables to an EJS partial?
    A) No, you can only pass one variable at a time.
    B) Yes, by passing an object containing all the variables.
    C) Yes, but only if they are declared in the main template.
    D) No, EJS does not support passing multiple variables.
    Answer: B) Yes, by passing an object containing all the variables.
    Explanation: You can pass multiple variables to an EJS partial by passing an object that contains all the variables as properties. For example: `<%- include('header', { title: 'Home', user: currentUser }) %>`.

15. How do you create a sidebar partial that includes navigation links in EJS?
    A) By using a JavaScript function to generate links dynamically
    B) By creating an HTML structure for the sidebar and including it as a partial
    C) By using inline CSS to create the sidebar
    D) By including the sidebar template inside a JavaScript controller
    Answer: B) By creating an HTML structure for the sidebar and including it as a partial
    Explanation: A sidebar partial in EJS is typically an HTML structure containing navigation links, which can be dynamically generated or static, and is included in the main template using the `include()` function.

16. How do you pass dynamic data to a sidebar partial in EJS?
    A) `<%- include('sidebar') %>`

B) `<%- include('sidebar', { links: links }) %>`

C) `<%- include('sidebar', links) %>`

D) `<% include('sidebar', { links: links }) %>`

Answer: B) `<%- include('sidebar', { links: links }) %>`

Explanation: To pass dynamic data such as a list of links to a sidebar partial, you can pass it as an object (e.g., `{ links: links }`) in the second argument of the `include()` function.

17. What happens if you try to include a partial that doesn't exist in EJS?

A) It throws a 404 error

B) EJS renders an empty string

C) EJS throws an error

D) The application crashes

Answer: C) EJS throws an error

Explanation: If an EJS partial is not found, EJS will throw an error indicating that it could not locate the partial file.

18. How can you organize EJS partials for different UI components such as header, footer, and sidebar in your project?

A) Store all partials in a single directory named `partials`

B) Store each partial in a separate file in different directories

C) Combine all partials into a single file

D) Use a database to store all partials

Answer: A) Store all partials in a single directory named `partials`

Explanation: It's common to store all EJS partials in a directory named `partials` to maintain a clean project structure and keep reusable components organized.

19. How can you dynamically change the content of a footer partial in EJS based on the page?

A) By passing a dynamic variable to the footer partial from the main template

B) By changing the footer directly in JavaScript

C) By using the `locals` property in the response object

D) By hardcoding the footer content

Answer: A) By passing a dynamic variable to the footer partial from the main template

Explanation: You can dynamically change the content of a footer by passing different data (e.g., `footerText: 'About Us'`) to the footer partial from the main template.

20. Which of the following syntax is used to include a header partial and pass a `title` variable to it in EJS?

A) `<%- include('header', { title: 'Page Title' }) %>`

B) `<% include('header', { title: 'Page Title' }) %>`

C) `<%- include('header') %>`

D) `<%- include('header', 'Page Title') %>`

Answer: A) `<%- include('header', { title: 'Page Title' }) %>`

Explanation: You pass the `title` variable to the header partial by providing it in an object within the second argument of the `include()` function.

21. Which of the following can EJS partials contain?

    A) Only HTML content

    B) Only JavaScript code

    C) Both HTML and JavaScript code

    D) Only CSS code

    Answer: C) Both HTML and JavaScript code

    Explanation: EJS partials can contain both HTML and JavaScript. You can dynamically render content or logic using JavaScript inside EJS partials.

22. What is the default behavior of an EJS partial when you include it in a template without passing any data?

    A) It throws an error

    B) It renders the partial without any dynamic data

    C) It ignores the partial

    D) It waits for data to be passed

    Answer: B) It renders the partial without any dynamic data

    Explanation: If no data is passed, EJS will simply render the partial as static HTML. If the partial has dynamic code (like `<%= title %>`), it will render empty or undefined.

23. Which of the following is the correct method for including a footer partial and passing dynamic content in an Express.js app using EJS?

    A) `res.render('index', { footerContent: '© 2024 MySite' })`

    B) `<%- include('footer', { footerContent: '© 2024 MySite' }) %>`

    C) `res.send('index', { footerContent: '© 2024 MySite' })`

    D) `<%- include('footer') %>`

    Answer: A) `res.render('index', { footerContent: '© 2024 MySite' })`

    Explanation: In Express.js, you use `res.render()` to render an EJS template and pass data to it. This can include passing dynamic data like `footerContent` to the `footer` partial within the template.

24. **Which EJS function is used to include external JavaScript files into an EJS template?**
    A) `<%- include() %>`
    B) `<script src="file.js"></script>`
    C) `<%- include('file.js') %>`
    D) `<% include('file.js') %>`
    Answer: B) `<script src="file.js"></script>`
    Explanation: To include external JavaScript files in EJS, you simply use the regular HTML `<script src="file.js"></script>` tag within the EJS template.
25. **How do you prevent HTML content from being escaped in an EJS partial?**
    A) By using `<%= include('partial') %>`
    B) By using `<%- include('partial') %>`
    C) By using `<% include('partial') %>`
    D) By using `{{ include('partial') }}`
    Answer: B) By using `<%- include('partial') %>`
    Explanation: The `<%-` syntax renders the content of the partial as raw HTML, without escaping any special characters.
26. **What is the best practice for managing large and complex UI components in EJS?**
    A) Write all HTML in one large EJS file
    B) Break down the UI into smaller partials and include them in a main template
    C) Avoid using partials for large components
    D) Use only static HTML for large components
    Answer: B) Break down the UI into smaller partials and include them in a main template
    Explanation: Best practice involves breaking down complex UI components into smaller, reusable EJS partials and then including them in a main template. This keeps the code modular and easier to maintain.
27. **How do you include a partial dynamically based on certain conditions in an EJS template?**
    A) Using an `if` statement in the main template
    B) Using the `include()` function with a conditional expression
    C) Using a JavaScript switch case inside the partial
    D) EJS does not support dynamic partial inclusion
    Answer: B) Using the `include()` function with a conditional expression
    Explanation: You can conditionally include a partial in EJS by using

JavaScript logic within the template, such as: `<%- include(condition ? 'header' : 'footer') %>`.

28. Which of the following can be used inside an EJS partial to handle loops?

    A) `forEach()`

    B) `for` loops

    C) `map()`

    D) All of the above

    Answer: D) All of the above

    Explanation: You can use regular JavaScript `for` loops, `forEach()` loops, or other array methods like `map()` inside EJS partials to dynamically generate content.

29. Can you reuse an EJS partial in different templates?

    A) No, EJS partials are template-specific

    B) Yes, as long as the partial is stored in the correct directory

    C) Yes, but only for the same type of data

    D) No, partials cannot be reused across templates

    Answer: B) Yes, as long as the partial is stored in the correct directory

    Explanation: EJS partials can be reused across different templates as long as they are stored in a consistent directory (e.g., `partials/`), and you include them properly.

30. Which of the following is the correct way to include a header partial and pass a `title` variable to it in an Express app with EJS?

    A) `res.render('index', { title: 'Home' })`

    B) `<%- include('header', { title: 'Home' }) %>`

    C) `<%- include('header') %>`

    D) `res.send('index')`

    Answer: A) `res.render('index', { title: 'Home' })`

    Explanation: In Express, you pass data to an EJS template (like `title`) through the `res.render()` method. The data passed will be available in the template, which can then be passed to partials using `include()`.

31. How do you include a sidebar partial and pass an array of links to it in EJS?

    A) `<%- include('sidebar', { links: ['Home', 'About', 'Contact'] }) %>`

    B) `<%- include('sidebar', links: ['Home', 'About', 'Contact']) %>`

    C) `<%- include('sidebar', ['Home', 'About', 'Contact']) %>`

D) `<%- include('sidebar', { links: 'Home, About, Contact' }) %>`

Answer: A) `<%- include('sidebar', { links: ['Home', 'About', 'Contact'] }) %>`

Explanation: You pass an array of links to a sidebar partial by including it within an object, like `{ links: ['Home', 'About', 'Contact'] }`. The sidebar partial can then iterate over the `links` array to dynamically generate navigation items.

32. What is the role of the `<%-` tag in EJS when including partials?

A) It is used for including static files

B) It escapes special characters in HTML content

C) It renders raw HTML without escaping characters

D) It runs JavaScript functions in partials

Answer: C) It renders raw HTML without escaping characters

Explanation: The `<%-` tag is used to render content in a partial without escaping special characters, allowing raw HTML to be inserted directly into the template.

33. Can you use JavaScript variables in EJS partials?

A) No, EJS does not support variables

B) Yes, JavaScript variables can be passed to partials from the main template

C) Yes, but only if the variable is declared inside the partial itself

D) No, only static content is allowed in EJS partials

Answer: B) Yes, JavaScript variables can be passed to partials from the main template

Explanation: You can pass JavaScript variables from the main template to an EJS partial. These variables can then be used inside the partial to dynamically render content.

34. How do you include a footer partial in an EJS template without passing any data?

A) `<%- include('footer') %>`

B) `<%- include('footer', {}) %>`

C) `<% include('footer') %>`

D) `<%- include('footer', '') %>`

Answer: A) `<%- include('footer') %>`

Explanation: If you don't need to pass any data, you can simply include the footer partial using `<%- include('footer') %>`. The partial will be rendered with its static content.

35. **What is a good practice when organizing EJS partials in a project with multiple layouts?**
A) Store all partials in the root directory for easy access
B) Use a `partials/` folder and separate the partials based on their role (e.g., `header.ejs`, `footer.ejs`, `sidebar.ejs`)
C) Place partials directly within the main template file
D) Avoid using partials for complex layouts
Answer: B) Use a `partials/` folder and separate the partials based on their role (e.g., `header.ejs`, `footer.ejs`, `sidebar.ejs`)
Explanation: It's best practice to organize partials into a dedicated `partials/` folder. This helps maintain a clean project structure and makes it easier to manage and locate specific partials.

36. **Which of the following is true about EJS partials and inheritance?**
A) EJS supports template inheritance directly like Django or Jinja2
B) EJS does not support inheritance, but you can use partials to reuse components
C) EJS supports inheritance through a custom `extends` keyword
D) EJS allows for nested partials but not inheritance
Answer: B) EJS does not support inheritance, but you can use partials to reuse components
Explanation: EJS does not have a built-in inheritance mechanism like Django or Jinja2. Instead, partials are used for code reuse, allowing for common UI elements (header, footer, etc.) to be shared across different templates.

37. **How can you modify the content of a partial dynamically when rendering an EJS template in Express?**
A) By passing dynamic data via `res.render()`
B) By using JavaScript inside the EJS partial itself
C) By modifying the partial file before each request
D) EJS does not allow dynamic content in partials
Answer: A) By passing dynamic data via `res.render()`
Explanation: Dynamic content is passed to EJS partials through the `res.render()` function in Express. The data is then available in the template and its partials.

38. **Which EJS tag is used to output content safely by escaping HTML characters?**
A) `<%= %>`
B) `<%- %>`
C) `<%# %>`

D) `<%= include('partial') %>`

Answer: A) `<%= %>`

Explanation: The `<%= %>` tag is used to output content in EJS and automatically escape any HTML characters, preventing XSS (Cross-Site Scripting) vulnerabilities by escaping special characters like <, >, and &.

39. How do you include a partial only if a condition is met in an EJS template?

A) Use the `if` statement inside the `include()` function

B) Use the `if` statement outside the `include()` function to wrap it

C) EJS does not support conditional partial inclusion

D) Use a custom JavaScript function to check the condition

Answer: B) Use the `if` statement outside the `include()` function to wrap it

Explanation: EJS allows you to use JavaScript logic, such as `if` statements, to conditionally include partials.

40. How can you include a partial from a different directory in EJS?

A) Provide the relative path to the partial file, like `../partials/header`

B) EJS only supports partials in the same directory

C) Use the `path` module to resolve the partial's path

D) EJS automatically searches all directories for partials

Answer: A) Provide the relative path to the partial file, like `../partials/header`

Explanation: EJS allows you to include partials from different directories by providing the correct relative path to the partial file, such as `../partials/header`.

41. What happens if you use the `<%- include('header') %>` syntax in EJS?

A) It escapes HTML content from the included file

B) It includes the header file but skips the JavaScript code

C) It renders the content of the included file without escaping it

D) It returns the file as plain text without rendering it

Answer: C) It renders the content of the included file without escaping it

Explanation: The `<%-` syntax ensures that the content from the included file is rendered as raw HTML, without escaping special characters.

42. Which of the following is the correct way to loop through an array and include a partial for each item in EJS?

A) `<% for (let item of items) { %> <%- include('itemPartial', { item }) %> <% } %>`

B) `<% include('itemPartial', { items }) %>`

C) `<%- for (let item of items) { include('itemPartial') } %>`

D) `<%- include('itemPartial') %>`

Answer: A) `<% for (let item of items) { %> <%- include('itemPartial', { item }) %> <% } %>`
Explanation: To loop through an array and include a partial for each item, you can use a `for` loop and the `include()` function. The `item` is passed to the partial in the second argument.

43. How do you ensure that an EJS partial renders dynamic content when the data changes?
A) Reload the partial file every time the page is refreshed
B) Use static data in the partials
C) Pass updated data to the partial from the main template during each render
D) Keep the partial static and don't use dynamic content
Answer: C) Pass updated data to the partial from the main template during each render
Explanation: Dynamic content in partials is rendered by passing updated data each time the template is rendered using `res.render()` in Express. This ensures the partial renders with the latest data.

44. What is the purpose of the `extends` keyword in some templating engines like Jinja2 or Django templates?
A) To import external JavaScript libraries
B) To create template inheritance for common layouts
C) To define reusable variables across templates
D) To render raw HTML content from partials
Answer: B) To create template inheritance for common layouts
Explanation: In templating engines like Jinja2 or Django, the `extends` keyword is used for template inheritance, allowing one template to inherit and extend another template's structure.

45. Which of the following methods helps in debugging missing partials in EJS?
A) Use the `console.log()` function to print the partial contents
B) Ensure the path to the partial is correct and check for typos
C) Avoid using partials altogether
D) Revert to static HTML for all partials
Answer: B) Ensure the path to the partial is correct and check for typos
Explanation: Missing or incorrect partials in EJS are often caused by incorrect file paths or typos in the `include()` statement. Ensure the file path is correct and matches the structure of the project.

46. What is a database?
A) A collection of data stored on physical paper

B) A collection of interrelated data stored electronically
C) A software used to manipulate images
D) A type of cloud storage
Answer: B) A collection of interrelated data stored electronically
Explanation: A database is an organized collection of data that is stored and accessed electronically. It enables easy management and retrieval of data.

47. Which of the following is an example of a relational database management system (RDBMS)?
A) Oracle
B) MongoDB
C) Hadoop
D) Cassandra
Answer: A) Oracle
Explanation: Oracle is a relational database management system (RDBMS) that uses tables to store and manage data. MongoDB, Hadoop, and Cassandra are examples of NoSQL databases.

48. What does SQL stand for?
A) Standard Query Language
B) Structured Query Language
C) Simple Query Language
D) Secure Query Language
Answer: B) Structured Query Language
Explanation: SQL stands for Structured Query Language, which is used to manage and manipulate relational databases. It is the standard language for querying and interacting with relational databases.

49. What is a primary key in a relational database?
A) A key that links tables together
B) A unique identifier for records in a table
C) A key used to create relationships between tables
D) A key that holds encrypted data
Answer: B) A unique identifier for records in a table
Explanation: A primary key uniquely identifies each record in a table, ensuring no two records have the same primary key value. It is used to maintain data integrity.

50. What is the purpose of a foreign key in a relational database?
A) It ensures data consistency
B) It links two tables together by referencing the primary key in another table
C) It stores large amounts of data

D) It prevents any data deletion
Answer: B) It links two tables together by referencing the primary key in another table
Explanation: A foreign key is used to establish a relationship between two tables by pointing to the primary key of another table, ensuring referential integrity.

51. **Which of the following is true about normalization?**
A) It aims to remove redundancy and improve data integrity
B) It reduces the complexity of queries
C) It creates more data relationships
D) It increases the size of the database
Answer: A) It aims to remove redundancy and improve data integrity
Explanation: Normalization is the process of organizing data to reduce redundancy and dependency. This makes data more consistent and easier to maintain.

52. **What does ACID stand for in database transactions?**
A) Atomicity, Consistency, Isolation, Durability
B) Atomicity, Connection, Integrity, Data
C) Association, Consistency, Isolation, Durability
D) Access, Consistency, Isolation, Data
Answer: A) Atomicity, Consistency, Isolation, Durability
Explanation: ACID properties ensure that database transactions are processed reliably. Atomicity ensures complete transactions, Consistency maintains data accuracy, Isolation ensures transaction independence, and Durability guarantees that data persists after a transaction.

53. **Which SQL clause is used to filter records in a query?**
A) ORDER BY
B) SELECT
C) WHERE
D) GROUP BY
Answer: C) WHERE
Explanation: The `WHERE` clause is used to filter records based on specified conditions in SQL queries. It is essential for querying data selectively.

54. **Which SQL function is used to calculate the sum of values in a column?**
A) AVG()
B) COUNT()
C) SUM()
D) MAX()
Answer: C) SUM()

Explanation: The `SUM()` function is used to calculate the total sum of values in a specific column in a table.

55. What is a "table" in a relational database?

A) A type of database query

B) A collection of related data organized in rows and columns

C) A method of data storage

D) A storage unit for large files

Answer: B) A collection of related data organized in rows and columns

Explanation: A table is a fundamental unit of data storage in a relational database. It organizes data in rows (records) and columns (fields), where each row represents a record and each column represents a field or attribute.

56. Which of the following is an example of a NoSQL database?

A) MySQL

B) MongoDB

C) PostgreSQL

D) Oracle

Answer: B) MongoDB

Explanation: MongoDB is a NoSQL database that stores data in a flexible, document-oriented format, unlike relational databases that use tables.

57. What is the purpose of the `JOIN` clause in SQL?

A) To combine data from two or more tables based on a related column

B) To delete data from a table

C) To group data based on a condition

D) To sort the result set

Answer: A) To combine data from two or more tables based on a related column

Explanation: The `JOIN` clause is used to retrieve data from multiple tables based on a related column, such as a foreign key and primary key.

58. What does the `GROUP BY` clause do in SQL?

A) It groups rows based on a column and applies aggregate functions

B) It orders the results alphabetically

C) It filters records based on specific conditions

D) It creates relationships between tables

Answer: A) It groups rows based on a column and applies aggregate functions

Explanation: The `GROUP BY` clause is used to group rows that have the same values in specified columns and allows aggregate functions like `COUNT()`, `SUM()`, `AVG()`, etc., to be applied.

59. **Which of the following is a characteristic of a "one-to-many" relationship?**
    A) A record in one table can relate to multiple records in another table
    B) A record in one table can relate to only one record in another table
    C) Both tables are identical
    D) There is no foreign key involved
    Answer: A) A record in one table can relate to multiple records in another table
    Explanation: In a one-to-many relationship, a single record in the "one" table can be associated with multiple records in the "many" table. This relationship is typically implemented using foreign keys.
60. **What does "referential integrity" mean in a database?**
    A) Ensures that foreign key values match primary key values in another table
    B) Ensures that primary keys are unique
    C) Ensures data consistency within a table
    D) Ensures that the database does not contain any redundant data
    Answer: A) Ensures that foreign key values match primary key values in another table
    Explanation: Referential integrity ensures that foreign keys in a table refer to valid primary key values in another table, preventing orphan records and maintaining data consistency across related tables.
61. **Which of the following is true about "denormalization"?**
    A) It increases data redundancy to improve query performance
    B) It reduces data redundancy to improve data integrity
    C) It organizes data into fewer tables
    D) It prevents data inconsistency
    Answer: A) It increases data redundancy to improve query performance
    Explanation: Denormalization involves intentionally introducing redundancy in a database to improve read performance by reducing the number of joins required in queries.
62. **Which of the following is NOT a database model?**
    A) Hierarchical model
    B) Relational model
    C) Graph model
    D) Computer model
    Answer: D) Computer model
    Explanation: Hierarchical, relational, and graph models are types of database models. "Computer model" is not a recognized database model.
63. **Which SQL command is used to remove a table from a database?**
    A) DELETE

B) REMOVE
C) DROP
D) ERASE
Answer: C) DROP
Explanation: The DROP statement is used to completely remove a table and its data from the database. The DELETE statement, on the other hand, removes data but keeps the table structure intact.

64. What is a "view" in a database?
A) A backup of the database
B) A stored procedure for data manipulation
C) A virtual table created from a query
D) A graphical representation of database schema
Answer: C) A virtual table created from a query
Explanation: A view is a virtual table that is generated by executing a query. It does not store data but presents data from one or more tables.

65. What is a "tuple" in a relational database?
A) A column in a table
B) A data type used for fields
C) A single record in a table
D) A relationship between tables
Answer: C) A single record in a table
Explanation: A tuple refers to a single row or record in a relational database table. Each tuple consists of a set of attributes (columns).

66. What is "data redundancy" in the context of a database?
A) Storing the same data in multiple places
B) Ensuring data consistency across tables
C) Normalizing data
D) Encrypting sensitive data
Answer: A) Storing the same data in multiple places
Explanation: Data redundancy occurs when the same data is stored in multiple locations within a database, which can lead to data inconsistency and inefficient storage.

67. What is a "transaction" in the context of a database?
A) A set of related operations that are treated as a single unit of work
B) A SQL query that modifies data
C) A procedure that ensures data security
D) A form of data retrieval
Answer: A) A set of related operations that are treated as a single unit of work
Explanation: A transaction is a sequence of operations that are executed

as a single unit of work. The transaction must either complete fully or not at all (atomicity), and it should leave the database in a consistent state.

68. **Which SQL clause is used to arrange the result set in a specified order?**
    A) ORDER BY
    B) GROUP BY
    C) SELECT
    D) FILTER
    Answer: A) ORDER BY
    Explanation: The `ORDER BY` clause in SQL is used to sort the result set by one or more columns, either in ascending or descending order.
69. **Which of the following is NOT a benefit of using databases?**
    A) Improved data security
    B) Faster data retrieval
    C) Data redundancy
    D) Data consistency
    Answer: C) Data redundancy
    Explanation: Databases aim to minimize data redundancy, which can lead to data inconsistency and inefficiencies. Using a database system ensures better data management and retrieval.
70. **What is the "data dictionary" in a database?**
    A) A physical storage unit for data
    B) A collection of metadata describing the structure of the database
    C) A backup system for database files
    D) A tool for optimizing query performance
    Answer: B) A collection of metadata describing the structure of the database
    Explanation: The data dictionary is a system catalog that contains metadata about the database schema, including information about tables, columns, data types, and relationships.
71. **Which of the following is a characteristic of SQL databases?**
    A) Schema-less structure
    B) Supports ACID properties
    C) Data is stored in key-value pairs
    D) Horizontal scalability
    Answer: B) Supports ACID properties
    Explanation: SQL databases, also known as relational databases, support ACID (Atomicity, Consistency, Isolation, Durability) properties, ensuring that database transactions are processed reliably.
72. **Which of the following is NOT a feature of NoSQL databases?**
    A) Flexible schema design

B) Horizontal scaling
C) SQL-based querying
D) Optimized for large datasets
Answer: C) SQL-based querying
Explanation: NoSQL databases typically do not use SQL for querying; instead, they use their own query languages or methods, depending on the type of NoSQL database (e.g., document-based, key-value).

73. Which of the following is a popular document-oriented NoSQL database?
A) MongoDB
B) Cassandra
C) Redis
D) Neo4j
Answer: A) MongoDB
Explanation: MongoDB is a document-oriented NoSQL database that stores data in BSON (Binary JSON) format, allowing flexible schema design and easy horizontal scaling.

74. Which SQL clause is used to filter records in a query?
A) SELECT
B) WHERE
C) GROUP BY
D) JOIN
Answer: B) WHERE
Explanation: The `WHERE` clause is used in SQL to filter records that meet a specified condition.

75. What is the main difference between SQL and NoSQL databases?
A) SQL databases are document-oriented, while NoSQL databases are relational
B) SQL databases use tables with fixed schemas, while NoSQL databases can store unstructured or semi-structured data
C) SQL databases are used for big data storage, while NoSQL is not
D) SQL databases do not support indexing, while NoSQL databases do
Answer: B) SQL databases use tables with fixed schemas, while NoSQL databases can store unstructured or semi-structured data
Explanation: SQL databases are structured and use tables with fixed schemas. NoSQL databases are more flexible, supporting unstructured or semi-structured data in various forms like documents, key-value pairs, or graphs.

76. What does the `JOIN` clause do in SQL?
A) Deletes records
B) Combines data from two or more tables based on a related column

C) Orders the result set
D) Updates data in a table
Answer: B) Combines data from two or more tables based on a related column
Explanation: The `JOIN` clause in SQL is used to combine rows from two or more tables based on a related column, typically a foreign key that links the tables.

77. **Which of the following is NOT a valid type of SQL join?**
A) INNER JOIN
B) LEFT JOIN
C) FULL JOIN
D) CONNEX JOIN
Answer: D) CONNEX JOIN
Explanation: "CONNEX JOIN" is not a valid SQL join type. The valid types are `INNER JOIN`, `LEFT JOIN`, `RIGHT JOIN`, and `FULL JOIN`.

78. **Which of the following is an example of a key-value store NoSQL database?**
A) MongoDB
B) Cassandra
C) Redis
D) Neo4j
Answer: C) Redis
Explanation: Redis is a key-value store NoSQL database that stores data in key-value pairs, providing fast data retrieval for applications.

79. **Which SQL command is used to remove data from a table?**
A) REMOVE
B) DELETE
C) DROP
D) TRUNCATE
Answer: B) DELETE
Explanation: The `DELETE` command in SQL is used to remove records from a table based on a condition. It removes specific rows while keeping the table structure intact.

80. **What is the purpose of the `GROUP BY` clause in SQL?**
A) It groups rows that have the same values into summary rows
B) It orders the result set
C) It updates data in the database
D) It filters records based on a condition
Answer: A) It groups rows that have the same values into summary rows
Explanation: The `GROUP BY` clause is used in SQL to group rows that have

the same values into summary rows, often used with aggregate functions like `COUNT()`, `SUM()`, and `AVG()`.

81. Which of the following is a key feature of NoSQL databases?
A) They are always ACID-compliant
B) They use SQL for querying
C) They support schema flexibility
D) They cannot scale horizontally
Answer: C) They support schema flexibility
Explanation: NoSQL databases offer schema flexibility, which allows them to store unstructured or semi-structured data, unlike SQL databases that require a fixed schema.

82. Which type of NoSQL database is optimized for querying data using graph structures?
A) Column-family store
B) Document store
C) Graph database
D) Key-value store
Answer: C) Graph database
Explanation: Graph databases, such as Neo4j, are optimized for storing and querying data in graph structures, making them ideal for handling relationships between entities.

83. Which SQL function is used to count the number of rows in a query result?
A) COUNT()
B) SUM()
C) AVG()
D) MAX()
Answer: A) COUNT()
Explanation: The `COUNT()` function in SQL is used to count the number of rows returned by a query or to count the number of non-NULL values in a column.

84. Which of the following is an example of a column-family NoSQL database?
A) MongoDB
B) Cassandra
C) Redis
D) Neo4j
Answer: B) Cassandra
Explanation: Cassandra is a column-family NoSQL database that stores data in columns rather than rows, optimizing it for querying large datasets efficiently.

85. **Which of the following is a benefit of using NoSQL databases?**
    A) Support for transactions
    B) Flexible data models that handle unstructured data
    C) Strong ACID properties
    D) Fixed schema design
    Answer: B) Flexible data models that handle unstructured data
    Explanation: NoSQL databases allow flexible data models that can store unstructured and semi-structured data, making them ideal for applications with rapidly changing or diverse data.
86. **What does the `HAVING` clause do in SQL?**
    A) Filters rows based on a condition
    B) Sorts the result set
    C) Filters groups created by `GROUP BY`
    D) Joins tables together
    Answer: C) Filters groups created by `GROUP BY`
    Explanation: The `HAVING` clause is used to filter groups after they have been created by the `GROUP BY` clause, often with aggregate functions like `SUM()` or `COUNT()`.
87. **Which of the following is an advantage of using SQL over NoSQL?**
    A) NoSQL databases can store unstructured data
    B) SQL databases have a fixed schema and structure
    C) SQL databases are more flexible in terms of scaling horizontally
    D) SQL databases use distributed architecture for high availability
    Answer: B) SQL databases have a fixed schema and structure
    Explanation: SQL databases use a well-defined schema that enforces a structured data format, which is advantageous for applications requiring consistent and organized data.
88. **In a relational database, what does the term "normalization" refer to?**
    A) Combining multiple tables into one
    B) Removing duplicate data and organizing data into separate tables
    C) Storing data in unstructured formats
    D) Encrypting sensitive data
    Answer: B) Removing duplicate data and organizing data into separate tables
    Explanation: Normalization is the process of organizing data into separate tables to reduce redundancy and improve data integrity.
89. **Which of the following is a feature of a "key-value" store NoSQL database?**
    A) Stores data as JSON documents
    B) Organizes data in a table format
    C) Stores data as key-value pairs

D) Optimized for complex queries and relationships
Answer: C) Stores data as key-value pairs
Explanation: Key-value stores, such as Redis and DynamoDB, store data as key-value pairs, where each key is unique and maps to a value.

90. Which of the following SQL data types is used to store date and time values?
A) INT
B) DATE
C) VARCHAR
D) BOOLEAN
Answer: B) DATE
Explanation: The DATE data type is used in SQL to store date and time values in a standardized format.

91. Which of the following is a disadvantage of NoSQL databases?
A) They support a fixed schema
B) They are often not ACID-compliant
C) They do not support horizontal scalability
D) They do not support large datasets
Answer: B) They are often not ACID-compliant
Explanation: Many NoSQL databases sacrifice full ACID compliance to achieve better performance and scalability. However, some NoSQL databases support eventual consistency rather than strict consistency.

92. What is the purpose of the ALTER TABLE command in SQL?
A) To delete a table
B) To insert new data into a table
C) To modify the structure of an existing table
D) To drop a database
Answer: C) To modify the structure of an existing table
Explanation: The ALTER TABLE command is used to modify the structure of an existing table, such as adding, deleting, or modifying columns.

93. Which of the following is a document-based NoSQL database?
A) MongoDB
B) Redis
C) Cassandra
D) Neo4j
Answer: A) MongoDB
Explanation: MongoDB is a document-based NoSQL database that stores data in BSON format, which is similar to JSON but includes additional data types.

94. What does the DISTINCT keyword do in SQL?
    A) Sorts the result set
    B) Filters out duplicate rows from the result set
    C) Groups data by a specified column
    D) Updates the data in a table
    Answer: B) Filters out duplicate rows from the result set
    Explanation: The DISTINCT keyword is used to remove duplicate rows from the result set in a query, ensuring that only unique values are returned.
95. Which of the following NoSQL databases is best suited for storing large amounts of time-series data?
    A) Cassandra
    B) MongoDB
    C) InfluxDB
    D) Redis
    Answer: C) InfluxDB
    Explanation: InfluxDB is a time-series database optimized for storing and querying large amounts of time-stamped data, such as logs and sensor data.
96. Which of the following is an example of a column-family store NoSQL database?
    A) MongoDB
    B) HBase
    C) Cassandra
    D) Redis
    Answer: B) HBase
    Explanation: HBase is a column-family store NoSQL database that stores data in columns rather than rows, allowing for efficient querying of large datasets, often used for big data applications.
97. What is the difference between DELETE and TRUNCATE in SQL?
    A) DELETE removes all data from a table, while TRUNCATE removes only specific rows
    B) DELETE is a DDL operation, while TRUNCATE is a DML operation
    C) DELETE can be rolled back, but TRUNCATE cannot
    D) TRUNCATE removes the table structure, while DELETE keeps it
    Answer: C) DELETE can be rolled back, but TRUNCATE cannot
    Explanation: DELETE is a DML operation that removes rows from a table and can be rolled back, while TRUNCATE is a DDL operation that removes all rows and cannot be rolled back.

98. **What is a common use case for a graph database?**
A) Storing unstructured documents
B) Analyzing network connections like social networks or supply chains
C) Storing large sets of time-series data
D) Handling transactions with a high degree of consistency
Answer: B) Analyzing network connections like social networks or supply chains
Explanation: Graph databases like Neo4j are ideal for analyzing relationships and connections in datasets, such as social networks, recommendation systems, and supply chains.

99. **Which SQL keyword is used to combine the results of two queries?**
A) JOIN
B) UNION
C) SELECT
D) GROUP BY
Answer: B) UNION
Explanation: The `UNION` operator is used to combine the results of two or more SELECT queries into a single result set, removing duplicate rows.

100. **Which of the following databases is best suited for use cases requiring high-speed, low-latency data retrieval?**
A) MongoDB
B) PostgreSQL
C) Redis
D) MySQL
Answer: C) Redis
Explanation: Redis is an in-memory data store that provides extremely fast, low-latency data retrieval. It is often used for caching, real-time analytics, and session storage.

101. **Which type of NoSQL database is best for storing JSON-like documents?**
A) Key-value store
B) Column-family store
C) Document store
D) Graph database
Answer: C) Document store
Explanation: Document stores, like MongoDB, store data in JSON-like formats (BSON in MongoDB), making them ideal for applications that need to store complex, nested data.

102. **In SQL, what does the `IN` operator do?**
A) Filters results based on a range of values

B) Filters results based on specific values in a list
C) Joins two tables together
D) Groups records based on a column value
Answer: B) Filters results based on specific values in a list
Explanation: The `IN` operator allows you to filter records that match any value in a specified list, such as `SELECT * FROM students WHERE grade IN ('A', 'B');`.

103. What type of NoSQL database is best suited for storing highly relational data?
A) Key-value store
B) Column-family store
C) Graph database
D) Document store
Answer: C) Graph database
Explanation: Graph databases, such as Neo4j, are ideal for storing highly relational data where relationships are a core part of the model, such as social networks or recommendation engines.

104. What is the purpose of the `DISTINCT` keyword in SQL?
A) To count the number of unique records
B) To remove duplicates from the result set
C) To create new tables
D) To update multiple records at once
Answer: B) To remove duplicates from the result set
Explanation: The `DISTINCT` keyword is used in SQL to eliminate duplicate rows from the result set, ensuring that only unique rows are returned.

105. Which of the following is an example of a NoSQL database that uses a key-value store?
A) MongoDB
B) Cassandra
C) Redis
D) Neo4j
Answer: C) Redis
Explanation: Redis is a key-value store NoSQL database that allows data to be stored as key-value pairs. It is often used for caching and session storage.

106. Which SQL command is used to modify existing records in a table?
A) UPDATE
B) MODIFY
C) ALTER
D) CHANGE

Answer: A) UPDATE

Explanation: The `UPDATE` statement is used to modify existing records in a table. It requires a `SET` clause to specify the new values and a `WHERE` clause to identify the records to be updated.

107. **Which of the following is an example of an ACID property of a transaction in a database?**

A) Speed
B) Atomicity
C) Storage optimization
D) Compression
Answer: B) Atomicity

Explanation: Atomicity is one of the ACID properties of a database transaction, which ensures that all parts of a transaction are completed successfully or none are, guaranteeing data integrity.

108. **Which of the following is NOT a valid type of NoSQL database?**

A) Column-family store
B) Relational store
C) Key-value store
D) Graph database
Answer: B) Relational store

Explanation: "Relational store" is not a type of NoSQL database. Relational databases are separate from NoSQL databases and use tables with fixed schemas.

109. **Which of the following is the primary advantage of using a NoSQL database?**

A) Rigid schema structure
B) High performance for read-heavy applications
C) Simple query language like SQL
D) Support for complex transactions
Answer: B) High performance for read-heavy applications

Explanation: NoSQL databases are often optimized for performance in read-heavy or write-heavy applications and can scale horizontally to handle large datasets efficiently.

110. **Which of the following is true about SQL databases?**

A) They do not require a schema
B) They scale horizontally by adding more nodes
C) They use SQL for querying and support ACID transactions
D) They use a flexible, document-based structure
Answer: C) They use SQL for querying and support ACID transactions

Explanation: SQL databases (relational databases) use SQL for querying and support ACID properties, ensuring data consistency and reliability.

111. **Which SQL function would you use to find the maximum value in a column?**

A) MAX()
B) AVG()
C) COUNT()
D) SUM()

Answer: A) MAX()

Explanation: The `MAX()` function in SQL returns the highest value in a specified column.

112. **What is a NoSQL database's approach to scalability?**

A) Vertical scaling
B) Horizontal scaling
C) Limited scaling
D) No scaling

Answer: B) Horizontal scaling

Explanation: NoSQL databases typically scale horizontally, meaning they distribute data across multiple servers to handle increased load and traffic.

113. **Which SQL command is used to remove a table from a database?**

A) DELETE
B) DROP
C) REMOVE
D) TRUNCATE

Answer: B) DROP

Explanation: The `DROP` command is used to remove a table (and its data) from a database permanently.

114. **Which of the following NoSQL databases is best for high-volume read and write workloads with small, discrete pieces of data?**

A) Cassandra
B) MongoDB
C) Redis
D) Neo4j

Answer: C) Redis

Explanation: Redis is highly optimized for fast read and write operations, especially for small data items, and is commonly used for caching and session storage.

115. **What is the purpose of `LIMIT` in SQL?**

A) To specify the columns to return
B) To filter rows based on a condition

C) To restrict the number of rows returned

D) To perform a join operation

Answer: C) To restrict the number of rows returned

Explanation: The `LIMIT` clause is used in SQL to restrict the number of rows returned by a query, typically used for pagination or testing.

116. **Which of the following is a disadvantage of SQL databases compared to NoSQL databases?**

A) Support for horizontal scaling

B) Fixed schema requirements

C) Support for unstructured data

D) High availability

Answer: B) Fixed schema requirements

Explanation: SQL databases have a fixed schema that requires predefined structures for tables, whereas NoSQL databases offer more flexible schema designs that can handle unstructured or semi-structured data.

117. **What is a common use case for a key-value store?**

A) Managing large, complex data relationships

B) Storing user sessions or cache data

C) Storing documents with flexible schema

D) Handling transactions

Answer: B) Storing user sessions or cache data

Explanation: Key-value stores like Redis are ideal for use cases like session management, caching, and storing data in simple key-value pairs for fast access.

118. **Which of the following is a valid data type in SQL?**

A) JSON

B) DOCUMENT

C) TEXT

D) OBJECT

Answer: C) TEXT

Explanation: TEXT is a valid data type in SQL that is used to store large strings of text data.

119. **What is a common disadvantage of NoSQL databases?**

A) Lack of flexibility in data models

B) Complex querying capabilities

C) Lack of full ACID compliance

D) Fixed schema requirements

Answer: C) Lack of full ACID compliance

Explanation: Many NoSQL databases do not fully support ACID compliance

in favor of scalability and performance. This can result in eventual consistency instead of strong consistency.

120. **Which SQL keyword is used to modify the structure of a table?**

A) CHANGE

B) UPDATE

C) ALTER

D) MODIFY

Answer: C) ALTER

Explanation: The ALTER command in SQL is used to modify the structure of an existing table, such as adding, deleting, or modifying columns.

121. **What type of database is MongoDB?**

A) Relational database

B) Column-family store

C) Document store

D) Key-value store

Answer: C) Document store

Explanation: MongoDB is a NoSQL database that stores data in a document-oriented format (BSON, which is similar to JSON), making it flexible and scalable for a variety of applications.

122. **Which of the following is a characteristic of MongoDB's data model?**

A) Fixed schema

B) Data is stored in rows and columns

C) Flexible schema using JSON-like documents

D) Data is stored in tables

Answer: C) Flexible schema using JSON-like documents

Explanation: MongoDB uses a flexible schema for storing data in documents (BSON), allowing each document in a collection to have its own structure.

123. **What is the default port number for MongoDB?**

A) 27017

B) 3306

C) 5432

D) 8080

Answer: A) 27017

Explanation: The default port for MongoDB is 27017, which is used for client connections and communications with the database.

124. **What does the term "BSON" stand for in MongoDB?**

A) Binary JavaScript Object Notation

B) Basic Structured Object Notation

C) Binary Syntax Object Notation

D) Binary Stored Object Notation
Answer: A) Binary JavaScript Object Notation
Explanation: BSON stands for Binary JSON, which is a binary-encoded format that MongoDB uses to store data. It is similar to JSON but includes additional data types, such as binary data and ObjectId.

125. **Which of the following is used to query documents in MongoDB?**
A) SQL
B) MongoDB Query Language (MQL)
C) MongoDB Query Interface (MQI)
D) JavaScript
Answer: B) MongoDB Query Language (MQL)
Explanation: MongoDB uses its own query language (MQL), which allows for querying documents in collections using a variety of methods, including operators for filtering, sorting, and aggregation.

126. **Which of the following is true about MongoDB collections?**
A) A collection contains rows
B) A collection contains documents
C) A collection has a fixed schema
D) A collection is similar to a relational database table
Answer: B) A collection contains documents
Explanation: In MongoDB, a collection is a group of documents (which are similar to rows in relational databases). Collections do not enforce a fixed schema, meaning the structure of each document can differ.

127. **Which of the following is NOT a valid data type in MongoDB?**
A) String
B) Integer
C) ObjectId
D) DateTime
Answer: B) Integer
Explanation: MongoDB uses `int` for storing integer values, not `Integer`. It also supports `String`, `ObjectId`, `Date`, `Array`, `Boolean`, and other types.

128. **How do you create a new collection in MongoDB?**
A) `db.createCollection()`
B) `create.collection()`
C) `new db.collection()`
D) `db.addCollection()`
Answer: A) `db.createCollection()`
Explanation: The `db.createCollection()` command is used to explicitly

create a new collection in MongoDB. However, MongoDB also allows collections to be created automatically when you insert a document.

129. **Which of the following methods is used to insert a document in MongoDB?**

A) `insert()`

B) `add()`

C) `save()`

D) `insertOne()`

Answer: D) `insertOne()`

Explanation: `insertOne()` **is used to insert a single document into a collection in MongoDB. There is also** `insertMany()` **for inserting multiple documents at once.**

130. **What is the default storage format for data in MongoDB?**

A) XML
B) BSON
C) JSON
D) CSV
Answer: B) BSON
Explanation: MongoDB stores data in BSON format (Binary JSON), which is a binary-encoded format that allows for more efficient storage and faster access to complex data types, such as `ObjectId` **and binary data.**

131. **What is an ObjectId in MongoDB?**

A) A data type used to store user-generated IDs
B) A unique identifier for each document in a collection
C) A reference to another document in the database
D) A query operator
Answer: B) A unique identifier for each document in a collection
Explanation: An `ObjectId` **is a 12-byte identifier used to uniquely identify documents in a MongoDB collection. It is automatically generated if not provided when a document is inserted.**

132. **Which of the following operations is used to update an existing document in MongoDB?**

A) `update()`

B) `set()`

C) `modify()`

D) `save()`

Answer: A) `update()`

Explanation: The `update()` **method in MongoDB is used to modify existing**

documents. You can specify a filter to find documents and an update operation to modify the document fields. MongoDB also supports `updateOne()` and `updateMany()` for updating single or multiple documents.

133. **Which of the following operators is used in MongoDB to find documents where a field's value is greater than a specified value?**

A) `$gt`

B) `$lt`

C) `$eq`

D) `$ne`

Answer: A) `$gt`

Explanation: The `$gt` (greater than) operator is used in MongoDB to find documents where a field's value is greater than the specified value. Similarly, `$lt` is used for "less than", $eq for "equal to", and $ne for "not equal to".

134. **Which of the following aggregation stages is used to group documents based on a field value in MongoDB?**

A) `$match`

B) `$group`

C) `$sort`

D) `$project`

Answer: B) `$group`

Explanation: The `$group` stage in MongoDB's aggregation pipeline is used to group documents by a specific field and can perform various operations, like summing or averaging values, for each group.

135. **What is the purpose of the `$match` stage in MongoDB's aggregation pipeline?**

A) To sort documents

B) To filter documents based on conditions

C) To project certain fields in documents

D) To group documents by a field

Answer: B) To filter documents based on conditions

Explanation: The `$match` stage is used to filter documents that match specific conditions in MongoDB's aggregation pipeline, similar to a `find()` query.

136. **Which of the following commands is used to remove a document from a MongoDB collection?**

A) `delete()`

B) `remove()`

C) `deleteOne()`

D) `drop()`

Answer: C) `deleteOne()`

Explanation: The `deleteOne()` method is used to remove a single document from a collection that matches a filter. There is also `deleteMany()` to remove multiple documents.

137. **Which of the following is used to ensure a field in a MongoDB document is unique?**

A) Index

B) Key

C) Unique constraint

D) Field validator

Answer: C) Unique constraint

Explanation: MongoDB allows you to enforce uniqueness of a field using a unique index. This ensures that no two documents can have the same value for that specific field.

138. **Which of the following is true about MongoDB indexes?**

A) Indexes improve read performance but can slow down write operations

B) Indexes can only be created on numeric fields

C) Indexes are automatically created for all fields

D) Indexes are not supported in MongoDB

Answer: A) Indexes improve read performance but can slow down write operations

Explanation: Indexes in MongoDB speed up query performance, especially for large datasets, but they can slow down write operations because the index must be updated every time a document is inserted or updated.

139. **Which of the following is used to check the status of MongoDB's replication set?**

A) `rs.status()`

B) `db.replication()`

C) `rs.check()`

D) `db.status()`

Answer: A) `rs.status()`

Explanation: The `rs.status()` command is used to check the status of the replica set in MongoDB, providing information on the members, their states, and the synchronization process.

140. **Which MongoDB command is used to create a new database?**

A) `db.create()`

B) `use <database>`

C) `createDatabase()`

D) `new db()`

Answer: B) `use <database>`

Explanation: In MongoDB, you create a new database by issuing the use `<database>` command. MongoDB will create the database only when you insert data into it.

141. **What is sharding in MongoDB?**

A) Replicating data across multiple servers
B) Splitting data across multiple collections
C) Splitting data across multiple servers to improve scalability
D) Sorting data within a collection
Answer: C) Splitting data across multiple servers to improve scalability
Explanation: Sharding is the process of distributing data across multiple servers (called shards) in MongoDB, allowing horizontal scaling and improving performance for large datasets.

142. **Which of the following is a valid query operator in MongoDB?**

A) `LIKE`

B) `IN`

C) `BETWEEN`

D) `EQUALS`

Answer: B) `IN`

Explanation: MongoDB uses the `$in` operator to query for documents where a field's value matches any value in a specified array. There is no direct `LIKE`, `BETWEEN`, or `EQUALS` operator in MongoDB like in SQL.

143. **What is the purpose of the `$project` stage in MongoDB's aggregation pipeline?**

A) To filter documents
B) To group documents
C) To reshape documents and specify which fields to include or exclude
D) To sort documents
Answer: C) To reshape documents and specify which fields to include or exclude
Explanation: The `$project` stage is used to include, exclude, or reshape fields in documents during the aggregation process, allowing fine control over the data returned.

144. **Which of the following is true about MongoDB's write concern?**

A) It specifies how many replicas must confirm a write operation

B) It determines the structure of documents in a collection
C) It defines the security level of database operations
D) It controls the performance of read queries
Answer: A) It specifies how many replicas must confirm a write operation
Explanation: Write concern in MongoDB specifies the level of acknowledgment requested from MongoDB for write operations, determining how many replicas must confirm the write.

145. **How can you make a field in a MongoDB collection case-insensitive?**
A) Use `Collation`
B) Use `lowercase` indexes
C) Use `ignoreCase` query option
D) Use `Text Indexes`
Answer: A) Use `Collation`
Explanation: MongoDB supports case-insensitive queries using `Collation`, which allows specifying rules for string comparison, including case insensitivity.

146. **Which npm package is used to connect a Node.js application to MongoDB?**
A) `mongoose`
B) `mongodb`
C) `mongo-client`
D) `mongoose-client`
Answer: B) `mongodb`
Explanation: The official MongoDB Node.js driver package is `mongodb`, which allows direct interaction with MongoDB. While `mongoose` is a popular Object Data Modeling (ODM) library that works on top of `mongodb`, the core connection is made using the `mongodb` package.

147. **How do you install the `mongodb` package in a Node.js project?**
A) `npm install mongo`
B) `npm install mongodb`
C) `npm install mongodb-driver`
D) `npm install mongoose`
Answer: B) `npm install mongodb`
Explanation: The correct command to install the MongoDB Node.js driver is `npm install mongodb`. This allows your Node.js app to interact directly with a MongoDB database.

148. **Which method is used to connect to MongoDB using the `mongodb` package in Node.js?**
A) `MongoClient.connect()`
B) `Mongo.connect()`
C) `connectToDatabase()`
D) `db.connect()`
Answer: A) `MongoClient.connect()`
Explanation: To connect to MongoDB, you use the `MongoClient.connect()` method. This method establishes a connection to the database and returns a promise.

149. **What is the default port number for MongoDB?**
A) 27015
B) 27017
C) 27018
D) 8080
Answer: B) 27017
Explanation: The default port number for MongoDB is `27017`. It is used when setting up a connection to MongoDB unless configured otherwise.

150. **Which of the following is required for MongoDB connection in Node.js?**
A) Connection string
B) Database name
C) Username and password
D) All of the above
Answer: D) All of the above
Explanation: When connecting to MongoDB, the connection string is essential, which may include the username, password, and the database name. Authentication details might be required based on the MongoDB instance configuration.

151. **What is the correct format for a MongoDB connection string in Node.js?**
A) `mongodb://localhost:27017/mydatabase`
B) `mongodb://username:password@localhost/mydatabase`
C) `mongodb://localhost/mydatabase`
D) `mongodb://localhost/mydb?authSource=admin`
Answer: A) `mongodb://localhost:27017/mydatabase`
Explanation: A MongoDB connection string typically follows the format `mongodb://<host>:<port>/<database>`. The database name is optional but can be specified in the URL.

152. **What does the `useNewUrlParser` option do when connecting to MongoDB using the Node.js driver?**

A) Enables parsing of MongoDB's internal URL format
B) Enables the connection string to be parsed according to the new parser
C) Enables automatic fallback for the connection string
D) Forces MongoDB to use an older URL parser
Answer: B) Enables the connection string to be parsed according to the new parser
Explanation: The `useNewUrlParser` option tells the MongoDB driver to use the new connection string parser. It is recommended to set this to `true` to avoid deprecation warnings.

153. **Which method is used to close a MongoDB connection in Node.js?**

A) `client.close()`
B) `MongoClient.disconnect()`
C) `db.close()`
D) `client.stop()`
Answer: A) `client.close()`
Explanation: The `client.close()` method is used to close an open connection to MongoDB when you are done interacting with the database.

154. **What is mongoose in the context of MongoDB and Node.js?**

A) A MongoDB driver
B) A database schema library
C) A query builder
D) A MongoDB administration tool
Answer: B) A database schema library
Explanation: mongoose is an Object Data Modeling (ODM) library for MongoDB and Node.js. It provides a higher-level abstraction for working with MongoDB, including features like schemas, validation, and model-based data manipulation.

155. **Which method in Mongoose is used to define a schema for a collection?**

A) `Schema.define()`
B) `Schema.create()`
C) `new Schema()`
D) `mongoose.Schema()`
Answer: C) `new Schema()`
Explanation: In Mongoose, the `new Schema()` method is used to define the structure of documents in a collection. It allows setting field types, validation rules, and other schema options.

156. **Which of the following methods is used to create a new document in a MongoDB collection using Mongoose?**

A) `Model.create()`

B) `Model.insert()`

C) `Model.add()`

D) `Model.new()`

Answer: A) `Model.create()`

Explanation: `Model.create()` is used in Mongoose to insert a new document into the database. It can accept a document or an array of documents as input.

157. **What is the purpose of the `find()` method in Mongoose?**

A) To create a new document

B) To update an existing document

C) To query documents in a collection

D) To delete documents

Answer: C) To query documents in a collection

Explanation: The `find()` method in Mongoose is used to query the database and retrieve documents based on specified criteria. It returns an array of matching documents.

158. **Which of the following is true about Mongoose's `save()` method?**

A) It is used to delete a document

B) It is used to save a new document or update an existing one

C) It is used to modify the database schema

D) It is used to create a new collection

Answer: B) It is used to save a new document or update an existing one

Explanation: The `save()` method in Mongoose is used to persist changes to a document. If the document is new, it is inserted into the database; otherwise, it updates the existing document.

159. **Which of the following methods in Mongoose allows updating multiple documents?**

A) `Model.update()`

B) `Model.updateMany()`

C) `Model.updateAll()`

D) `Model.updateDocuments()`

Answer: B) `Model.updateMany()`

Explanation: The `updateMany()` method in Mongoose is used to update multiple documents in a collection based on a filter condition.

160. **What is the purpose of `mongoose.connect()`?**
A) To insert data into the database
B) To connect to a MongoDB database
C) To close the database connection
D) To configure the Mongoose schema
Answer: B) To connect to a MongoDB database
Explanation: `mongoose.connect()` is used to establish a connection to the MongoDB database. It takes a connection string and configuration options as arguments.

161. **How do you define a required field in a Mongoose schema?**
A) `{ required: true }`
B) `{ isRequired: true }`
C) `{ mandatory: true }`
D) `{ fieldRequired: true }`
Answer: A) `{ required: true }`
Explanation: In Mongoose, you can define a field as required by setting the `required` property to `true` in the schema definition.

162. **Which of the following methods is used to delete a single document in Mongoose?**
A) `Model.delete()`
B) `Model.remove()`
C) `Model.deleteOne()`
D) `Model.drop()`
Answer: C) `Model.deleteOne()`
Explanation: The `deleteOne()` method in Mongoose is used to delete a single document based on the given filter. There is also a `deleteMany()` method for deleting multiple documents.

163. **What is the purpose of the `populate()` method in Mongoose?**
A) To create new fields in the schema
B) To fetch associated documents from referenced collections
C) To delete documents from the database
D) To modify a schema's structure
Answer: B) To fetch associated documents from referenced collections
Explanation: The `populate()` method in Mongoose is used to retrieve related documents from other collections, effectively "joining" the data based on references defined in the schema.

164. **Which of the following is the correct way to define an ObjectId reference in a Mongoose schema?**

A) `type: ObjectId, ref: 'ModelName'`

B) `type: 'ObjectId', ref: 'ModelName'`

C) `type: 'ID', ref: 'ModelName'`

D) `type: 'ObjectID', ref: 'ModelName'`

Answer: A) `type: ObjectId, ref: 'ModelName'`

Explanation: In Mongoose, to define a reference to another collection, you use `type: ObjectId` and `ref: 'ModelName'`. The `ref` points to the model being referenced.

165. **How can you handle errors when connecting to MongoDB using Mongoose in Node.js?**

A) Using the `try-catch` block

B) Using the `connect()` callback

C) Using the `connect().catch()` method

D) All of the above

Answer: D) All of the above

Explanation: You can handle connection errors in Mongoose using `try-catch`, `connect()` callback, or `connect().catch()` for promise-based error handling.

166. **Which MongoDB command is used to list all databases?**

A) `db.show()`

B) `db.list()`

C) `show dbs`

D) `list databases`

Answer: C) `show dbs`

Explanation: To list all databases in MongoDB, the `show dbs` command is used. This will return a list of all databases in the MongoDB server.

167. **Which option in `mongoose.connect()` ensures the application reconnects to MongoDB if the connection is lost?**

A) `autoReconnect: true`

B) `reconnectTries: 5`

C) `useUnifiedTopology: true`

D) `useNewUrlParser: true`

Answer: A) `autoReconnect: true`

Explanation: The `autoReconnect: true` option in Mongoose ensures that the application automatically tries to reconnect to MongoDB if the connection is lost. This option is used in older versions of Mongoose.

168. **What is the purpose of the `useUnifiedTopology` option in `mongoose.connect()`?**
A) To allow using the new MongoDB driver topology engine
B) To enable authentication mechanisms
C) To enable advanced query options
D) To enable auto-reconnect behavior
Answer: A) To allow using the new MongoDB driver topology engine
Explanation: The `useUnifiedTopology` option ensures that Mongoose uses the new MongoDB driver's unified topology engine, which improves connection management and provides better handling of replica sets and sharded clusters.

169. **How can you handle asynchronous operations in Node.js when connecting to MongoDB?**
A) Using callbacks
B) Using Promises
C) Using async/await
D) All of the above
Answer: D) All of the above
Explanation: In Node.js, you can handle asynchronous operations using callbacks, Promises, or async/await syntax. Mongoose and the MongoDB driver support all three approaches to handle asynchronous code.

170. **Which method in the MongoDB Node.js driver is used to get a reference to a specific collection?**
A) `db.getCollection()`
B) `db.collection()`
C) `client.db()`
D) `db.get()`
Answer: B) `db.collection()`
Explanation: The `db.collection()` method is used to get a reference to a specific collection in MongoDB. You can then perform operations like `find()`, `insertOne()`, `update()`, etc., on that collection.

171. **What is the function of the `findOne()` method in the MongoDB Node.js driver?**
A) To retrieve all documents from a collection
B) To retrieve a single document that matches the given filter
C) To insert a new document into a collection
D) To update a document
Answer: B) To retrieve a single document that matches the given filter

Explanation: The `findOne()` method retrieves a single document that matches the specified filter. If no document matches, it returns `null`.

172. **Which of the following options is used to configure the MongoDB connection in Mongoose?**

A) `options`

B) `config`

C) `connectionOptions`

D) `settings`

Answer: A) `options`

Explanation: Mongoose uses the `options` object to configure various aspects of the connection, such as enabling use of the new URL parser (`useNewUrlParser`), automatic reconnection, and other settings.

173. **What is the purpose of `mongoose.disconnect()`?**

A) To terminate a specific connection
B) To close all connections to MongoDB
C) To disconnect from a specific database
D) To clear the database
Answer: B) To close all connections to MongoDB
Explanation: The `mongoose.disconnect()` method is used to close the connection to MongoDB. This is typically used when you are done with database operations and want to cleanly close the connection.

174. **Which of the following is a correct way to define a default value for a field in Mongoose?**

A) `field: { default: 'value' }`

B) `field: { defaultValue: 'value' }`

C) `field: 'value'`

D) `field: { default: function() { return 'value'; } }`

Answer: A) `field: { default: 'value' }`
Explanation: In Mongoose, you can define default values for fields using the `default` keyword in the schema definition. For example, `field: { type: String, default: 'value' }`.

175. **Which of the following is true about Mongoose schemas?**

A) A schema defines the structure of the database itself
B) A schema is only used for data validation
C) A schema defines the structure of documents within a collection
D) A schema defines the indexes used by the database
Answer: C) A schema defines the structure of documents within a collection

Explanation: A Mongoose schema defines the structure of documents within a collection, including field types, validation rules, default values, and indexes.

176. **Which command is used to list all collections in a MongoDB database?**

A) `show collections`

B) `list collections`

C) `db.collections()`

D) `listDatabaseCollections()`

Answer: A) `show collections`

Explanation: The `show collections` command is used to list all collections in the current database in the MongoDB shell.

177. **What does the `db.createCollection()` method do in MongoDB?**

A) Creates a new database
B) Creates a new collection within the database
C) Creates a new document within a collection
D) Creates a new index for a collection
Answer: B) Creates a new collection within the database

Explanation: The `db.createCollection()` method is used to explicitly create a new collection within the MongoDB database. In most cases, collections are created automatically when you insert documents into them.

178. **Which of the following methods is used to find multiple documents in MongoDB?**

A) `find()`

B) `findOne()`

C) `get()`

D) `query()`

Answer: A) `find()`

Explanation: The `find()` method is used to retrieve multiple documents from a MongoDB collection that match a given query. It returns a cursor that can be iterated over to access the matching documents.

179. **What is the purpose of the `skip()` method in MongoDB queries?**

A) To exclude a field from the results
B) To limit the number of results returned
C) To skip a specified number of documents in the result set
D) To order the result set by a specific field
Answer: C) To skip a specified number of documents in the result set

Explanation: The `skip()` method is used to skip a specified number of documents in the query results. It is commonly used for pagination.

180. **Which of the following is used to limit the number of documents returned by a MongoDB query?**

A) `limit()`

B) `slice()`

C) `restrict()`

D) `count()`

Answer: A) `limit()`

Explanation: The `limit()` method is used to specify the maximum number of documents to return in a query. It is useful for pagination or limiting the result set.

181. **What does the `aggregate()` method in MongoDB do?**

A) Executes an aggregation query to compute aggregate values
B) Creates an index on a field
C) Aggregates multiple collections into one
D) Retrieves all documents in a collection
Answer: A) Executes an aggregation query to compute aggregate values
Explanation: The `aggregate()` method is used to perform complex queries on the MongoDB data, such as filtering, grouping, and performing calculations on data. It is often used for aggregating results, like summing, averaging, or counting documents.

182. **How can you set a custom index on a field in a Mongoose schema?**

A) `schema.addIndex('fieldName')`

B) `schema.index({ fieldName: 1 })`

C) `schema.index('fieldName', { type: 'unique' })`

D) `schema.addIndex({ fieldName: 1 })`

Answer: B) `schema.index({ fieldName: 1 })`

Explanation: To set a custom index in Mongoose, you use the `schema.index()` method. The argument `{ fieldName: 1 }` specifies the field and the order of the index (ascending, denoted by 1).

183. **How do you perform a case-insensitive search in MongoDB using Node.js?**

A) By using the `$text` operator

B) By using regular expressions with the `i` flag

C) By using `$match` with `collation`

D) All of the above
Answer: D) All of the above

Explanation: Case-insensitive searches can be performed in MongoDB using regular expressions with the `i` flag (`/pattern/i`), the `$text` operator (if a text index is defined), or by specifying a `collation` option in a query.

184. **What does the `populate()` method do in Mongoose?**
   A) It populates missing fields in a document
   B) It replaces references to other collections with actual documents
   C) It inserts default values into the database
   D) It populates the database with initial data
   Answer: B) It replaces references to other collections with actual documents
   Explanation: The `populate()` method in Mongoose is used to replace a field that contains a reference (an `ObjectId`) to another document with the actual document data from the referenced collection.

185. **Which of the following is the correct way to query documents in a MongoDB collection using Node.js?**
   A) `db.find({})`
   B) `db.query({})`
   C) `db.collection.find({})`
   D) `collection.find()`
   Answer: C) `db.collection.find({})`
   Explanation: In MongoDB's Node.js driver, the correct syntax to query a collection is `db.collection.find({})`, where `db.collection` refers to the collection you are querying and `{}` is the query filter.

186. **What is the default behavior of MongoDB's `find()` method?**
   A) It only returns a single document
   B) It returns all documents in the collection
   C) It returns a cursor to iterate over documents
   D) It performs a full-text search
   Answer: C) It returns a cursor to iterate over documents
   Explanation: The `find()` method in MongoDB returns a cursor, which allows you to iterate over the query results. You can use `.toArray()` to convert the cursor into an array or iterate with `.forEach()`.

187. **What is Mongoose in the context of Node.js and MongoDB?**
   A) A MongoDB GUI tool
   B) An ORM (Object-Relational Mapping) library
   C) An Object Data Modeling (ODM) library
   D) A MongoDB database engine

Answer: C) An Object Data Modeling (ODM) library

Explanation: Mongoose is an ODM library for MongoDB and Node.js. It provides a higher-level abstraction over the MongoDB native driver and helps model application data with schemas and models.

188. **How do you install Mongoose in a Node.js application?**

A) `npm install mongoose`

B) `npm install mongo`

C) `npm install mongoose-driver`

D) `npm install mongo-client`

Answer: A) `npm install mongoose`

Explanation: To install Mongoose in a Node.js application, you run `npm install mongoose` to add it as a dependency in the project.

189. **Which method in Mongoose is used to define a schema for a collection?**

A) `new Schema()`

B) `mongoose.Schema()`

C) `mongoose.create()`

D) `defineSchema()`

Answer: B) `mongoose.Schema()`

Explanation: In Mongoose, you use the `mongoose.Schema()` method to define the schema of documents in a collection. This includes specifying field types, validation, default values, and more.

190. **Which method is used to create a model from a schema in Mongoose?**

A) `mongoose.model()`

B) `Schema.model()`

C) `new Schema().createModel()`

D) `mongoose.create()`

Answer: A) `mongoose.model()`

Explanation: The `mongoose.model()` method is used to create a model from a schema. The model is then used to interact with the MongoDB collection.

191. **Which of the following is a valid data type for a Mongoose schema?**

A) `String`

B) `Number`

C) `Date`

D) All of the above

Answer: D) All of the above

Explanation: Mongoose supports various data types, including `String`, `Number`, `Date`, `Buffer`, `Boolean`, `Array`, `ObjectId`, and more.

192. **How do you define a required field in a Mongoose schema?**

A) `{ required: true }`

B) `{ mandatory: true }`

C) `{ isRequired: true }`

D) `{ must: true }`

Answer: A) `{ required: true }`

Explanation: In Mongoose, you can define a field as required by setting the `required` property to `true` in the schema definition.

193. **What is the purpose of `mongoose.connect()`?**

A) To create a schema

B) To establish a connection to a MongoDB database

C) To create a new document in a collection

D) To disconnect from the database

Answer: B) To establish a connection to a MongoDB database

Explanation: `mongoose.connect()` is used to connect the Node.js application to a MongoDB database. It takes the connection string and optional configurations.

194. **Which of the following methods is used to insert a new document into a collection in Mongoose?**

A) `Model.insertOne()`

B) `Model.save()`

C) `Model.create()`

D) `Model.add()`

Answer: C) `Model.create()`

Explanation: The `Model.create()` method is used to insert new documents into a collection in MongoDB. You can pass an object or an array of objects to this method.

195. **Which method is used to find a single document in a Mongoose model?**

A) `Model.findOne()`

B) `Model.find()`

C) `Model.getOne()`

D) `Model.retrieve()`

Answer: A) `Model.findOne()`

Explanation: `Model.findOne()` is used to retrieve a single document that matches the specified filter from the collection.

196. **What does the `Model.find()` method return in Mongoose?**

A) A single document
B) A cursor to the documents
C) An array of documents
D) A promise
Answer: C) An array of documents
Explanation: `Model.find()` returns an array of documents that match the given query filter. If no filter is provided, it returns all documents in the collection.

197. **What is the purpose of the `Model.updateOne()` method in Mongoose?**

A) To update multiple documents
B) To find a document by its ID
C) To update a single document based on the given filter
D) To delete a document
Answer: C) To update a single document based on the given filter
Explanation: `Model.updateOne()` is used to update a single document that matches the given filter. If no document is found, no update occurs.

198. **Which method is used to update multiple documents in Mongoose?**

A) `Model.updateMany()`
B) `Model.updateAll()`
C) `Model.modifyMany()`
D) `Model.batchUpdate()`
Answer: A) `Model.updateMany()`
Explanation: `Model.updateMany()` is used to update multiple documents that match a given filter. This is useful for bulk updates.

199. **What is the purpose of the `Model.remove()` method in Mongoose?**

A) To remove all documents in a collection
B) To remove documents based on a filter
C) To delete the collection itself
D) To clear a document's fields
Answer: B) To remove documents based on a filter
Explanation: `Model.remove()` is used to delete documents from the database that match the specified filter.

200. **Which of the following is a valid way to define a reference between two Mongoose models?**

A) `ref: 'ModelName'`
B) `type: 'ModelName'`
C) `foreignKey: 'ModelName'`

D) `link: 'ModelName'`

Answer: A) `ref: 'ModelName'`

Explanation: In Mongoose, you define a reference between two models using `ref: 'ModelName'` in the schema definition. This creates a relationship between the current model and the referenced model.

201. **How do you define a field as a unique key in Mongoose?**

A) `unique: true`

B) `key: 'unique'`

C) `field: { unique: true }`

D) `field: 'unique'`

Answer: A) `unique: true`

Explanation: In Mongoose, to define a field as a unique key, you set the `unique: true` property on the schema field.

202. **Which of the following is true about Mongoose model instance methods?**

A) They are static methods that operate on models

B) They are called on model instances (documents)

C) They are called before the document is saved

D) They are used to query the database directly

Answer: B) They are called on model instances (documents)

Explanation: Instance methods are defined on model instances (documents) and are used to perform operations or manipulations on individual documents.

203. **Which of the following is used to handle validation in Mongoose?**

A) `validate()`

B) `validateSync()`

C) `validateField()`

D) `Model.valid()`

Answer: A) `validate()`

Explanation: The `validate()` method is used to perform validation on a document before saving it to the database. You can define custom validation rules in the schema.

204. **How do you handle asynchronous validation in Mongoose?**

A) By returning a promise in the validator

B) By using the async keyword in the validator

C) By using a callback function in the validator

D) All of the above

Answer: D) All of the above

Explanation: Mongoose supports asynchronous validation by returning a promise, using the async keyword, or using a callback function in the validator.

205. **What is the purpose of the `pre()` middleware in Mongoose?**

A) To execute code before a document is saved
B) To validate the document before saving
C) To remove a document from the collection
D) To transform data after saving
Answer: A) To execute code before a document is saved
Explanation: The `pre()` middleware in Mongoose is used to execute functions before certain actions (e.g., `save()`, `update()`) are performed on a document.

206. **Which of the following is true about Mongoose's `populate()` method?**

A) It is used to query for documents in multiple collections
B) It is used to replace references in a document with actual document data
C) It creates new references in the database
D) It is used to update documents in related collections
Answer: B) It is used to replace references in a document with actual document data
Explanation: The `populate()` method in Mongoose is used to populate references (i.e., `ObjectId` fields) with actual data from the referenced collection. This is similar to SQL joins.

207. **Which method in Mongoose is used to define a default value for a field?**

A) `default()`
B) `field.default()`
C) `field: { default: 'value' }`
D) `field: { defaultValue: 'value' }`
Answer: C) `field: { default: 'value' }`
Explanation: In Mongoose, you can define default values for fields in the schema by using `default: 'value'`.

208. **What does the `Model.countDocuments()` method do?**

A) Counts the number of documents in a collection
B) Counts the documents that match a specific filter
C) Counts the number of collections in a database
D) Counts the number of fields in a document
Answer: B) Counts the documents that match a specific filter
Explanation: `Model.countDocuments()` is used to count the number of documents that match a given query filter in a collection.

209. **Which method is used to paginate results in Mongoose?**

A) `skip()` and `limit()`

B) `find()` and `sort()`

C) `next()` and `previous()`

D) `limit()` and `orderBy()`

Answer: A) `skip()` and `limit()`

Explanation: Pagination in Mongoose is commonly handled by using `skip()` to skip over a set number of documents and `limit()` to specify the number of documents to return.

210. **Which of the following methods allows you to chain multiple queries in Mongoose?**

A) `exec()`

B) `then()`

C) `chain()`

D) `select()`

Answer: A) `exec()`

Explanation: The `exec()` method is used to execute queries in Mongoose, and it allows you to chain multiple query methods (like `find()`, `select()`, etc.) before actually executing the query.

211. **Which of the following methods is used to remove a single document in Mongoose?**

A) `Model.deleteOne()`

B) `Model.removeOne()`

C) `Model.delete()`

D) `Model.deleteMany()`

Answer: A) `Model.deleteOne()`

Explanation: The `Model.deleteOne()` method is used to remove a single document that matches a specified filter. If no document is found, no removal occurs.

212. **How do you define an array of subdocuments in Mongoose?**

A) `field: [{ type: String }]`

B) `field: [String]`

C) `field: { type: [String] }`

D) `field: Array[String]`

Answer: A) `field: [{ type: String }]`

Explanation: In Mongoose, you define an array of subdocuments by

specifying an array with the `type` of the subdocument. For example, `field: [{ type: String }]` defines an array of strings.

213. **What does the `virtual()` method in Mongoose do?**

A) Creates a virtual field that can be used in queries

B) Creates a computed field that doesn't get stored in the database

C) Defines a field with dynamic values

D) Creates a relationship between models

Answer: B) Creates a computed field that doesn't get stored in the database

Explanation: The `virtual()` method in Mongoose allows you to define fields that are not stored in the database but are calculated dynamically. These fields are computed based on other fields.

214. **Which Mongoose method is used to count documents that match a query?**

A) `Model.count()`

B) `Model.countDocuments()`

C) `Model.size()`

D) `Model.getCount()`

Answer: B) `Model.countDocuments()`

Explanation: The `Model.countDocuments()` method is used to count the number of documents that match a given query filter in the collection.

215. **How can you set a custom index in Mongoose?**

A) `schema.index({ fieldName: 1 })`

B) `Model.setIndex({ fieldName: 1 })`

C) `schema.createIndex({ fieldName: 1 })`

D) `Model.index({ fieldName: 1 })`

Answer: A) `schema.index({ fieldName: 1 })`

Explanation: To create a custom index in Mongoose, you use `schema.index({ fieldName: 1 })`. This creates an index on the specified field to optimize queries.

216. **What does the `mongoose.Schema.Types.ObjectId` represent?**

A) A string-based identifier for documents

B) A reference to another document in the database

C) A primary key for documents

D) A unique identifier for the database itself

Answer: B) A reference to another document in the database

Explanation: `mongoose.Schema.Types.ObjectId` is used to represent a

reference to another document in MongoDB. It is commonly used to create relationships between documents in different collections.

217. **Which method would you use to retrieve the first document that matches a filter in Mongoose?**

A) `Model.findOne()`

B) `Model.findFirst()`

C) `Model.getFirst()`

D) `Model.find()`

Answer: A) `Model.findOne()`

Explanation: The `Model.findOne()` method retrieves the first document that matches a specified filter. It returns a single document or `null` if no document is found.

218. **How do you handle errors in Mongoose queries?**

A) By using `try-catch` blocks

B) By passing a callback function with error handling

C) By using `.catch()` on the promise

D) All of the above

Answer: D) All of the above

Explanation: Errors in Mongoose queries can be handled using `try-catch` blocks, passing a callback with error handling, or using `.catch()` when working with promises.

219. **What is the purpose of `mongoose.Schema.Types.Mixed`?**

A) To define a field that can contain any data type

B) To create a custom index

C) To create an array of subdocuments

D) To define a reference to another model

Answer: A) To define a field that can contain any data type

Explanation: `mongoose.Schema.Types.Mixed` is used to define a field that can store any type of data, such as an object or an array of mixed types. It is flexible and allows for dynamic data.

220. **What is the purpose of `Model.findById()` in Mongoose?**

A) To find a document by a unique identifier

B) To find multiple documents by their IDs

C) To find documents by their references

D) To find documents by a custom field

Answer: A) To find a document by a unique identifier

Explanation: `Model.findById()` is used to find a document by its unique _id field. This is a shortcut for `Model.findOne({ _id: id })`.

221. **Which of the following methods is used to remove all documents that match a filter in Mongoose?**

A) `Model.removeMany()`

B) `Model.deleteMany()`

C) `Model.removeAll()`

D) `Model.clear()`

Answer: B) `Model.deleteMany()`

Explanation: The `Model.deleteMany()` **method is used to delete all documents that match a given filter. It is useful for bulk deletion.**

222. **How do you define a custom validation function in Mongoose?**

A) By using the `validate` **option in the schema definition**

B) By using the `customValidator()` **function**

C) By using `schema.validate()`

D) By creating a `Validator` **model**

Answer: A) By using the `validate` **option in the schema definition**

Explanation: You can define custom validation functions in Mongoose by using the `validate` **option in the schema. You can define either synchronous or asynchronous validation logic.**

223. **What is the default behavior of the** `findOneAndUpdate()` **method in Mongoose?**

A) It updates the first document found and returns the updated document

B) It updates all matching documents and returns the updated documents

C) It updates the first document found and returns the original document

D) It updates the entire collection and returns a success message

Answer: C) It updates the first document found and returns the original document

Explanation: By default, `findOneAndUpdate()` **updates the first document that matches the filter and returns the original document before the update. You can change this behavior by using the new option to return the updated document instead.**

224. **How do you enable automatic timestamps for documents in Mongoose?**

A) `{ timestamps: true }`

B) `{ autoTimestamps: true }`

C) `{ createdAt: true, updatedAt: true }`

D) `{ timestamps: 'true' }`

Answer: A) `{ timestamps: true }`

Explanation: To enable automatic creation of `createdAt` **and** `updatedAt`

fields for documents, you set `{ timestamps: true }` in the schema definition.

225. **What is the purpose of the `next()` function in Mongoose middleware?**
A) To pass control to the next middleware function
B) To return the result of the current middleware
C) To stop the middleware chain
D) To log errors in the middleware
Answer: A) To pass control to the next middleware function
Explanation: The `next()` function in Mongoose middleware is used to pass control to the next middleware function in the stack. If no middleware is defined, it allows the operation (e.g., save, update) to continue.

226. **Which of the following is used to define a field that is an enum in Mongoose?**
A) `enum: ['value1', 'value2']`
B) `type: 'enum', values: ['value1', 'value2']`
C) `enumField: { type: String, values: ['value1', 'value2'] }`
D) `enum: ['value1', 'value2'], type: String`
Answer: C) `enumField: { type: String, values: ['value1', 'value2'] }`
Explanation: To define an enum field in Mongoose, you specify the field type as `String` and use the `values` option to define the allowed set of values for the field.

227. **What does the `Model.findOneAndDelete()` method do in Mongoose?**
A) Finds and deletes a document based on a filter
B) Finds and updates a document based on a filter
C) Finds and returns the first document that matches the filter
D) Deletes a document without a filter
Answer: A) Finds and deletes a document based on a filter
Explanation: `Model.findOneAndDelete()` finds a document that matches the specified filter and deletes it. It returns the deleted document.

228. **Which of the following methods is used to update a document in Mongoose without validating it?**
A) `Model.updateOne()`
B) `Model.updateMany()`
C) `Model.update()`
D) `Model.findOneAndUpdate()`
Answer: C) `Model.update()`
Explanation: The `Model.update()` method in Mongoose performs an

update on documents without triggering validation. However, `updateOne()` and `updateMany()` will trigger validation unless explicitly disabled.

229. **Which of the following is the correct method to define a unique field in Mongoose?**

A) `field: { unique: true }`

B) `field: { index: true, unique: true }`

C) `field: { uniqueField: true }`

D) `field: { isUnique: true }`

Answer: A) `field: { unique: true }`

Explanation: In Mongoose, you can define a unique field by using the `unique: true` option in the schema. This ensures that no two documents in the collection can have the same value for this field.

230. **What is the default behavior of the `findOne()` method in Mongoose if no documents match the query?**

A) Throws an error

B) Returns `null`

C) Returns an empty array

D) Returns a new document

Answer: B) Returns `null`

Explanation: The `findOne()` method returns the first document that matches the filter. If no documents match, it returns `null`.

231. **What is the main difference between `Model.find()` and `Model.findOne()` in Mongoose?**

A) `find()` returns a single document, `findOne()` returns multiple documents

B) `findOne()` returns a single document, `find()` returns multiple documents

C) `find()` performs faster than `findOne()`

D) `findOne()` is used for inserting data, `find()` is used for querying

Answer: B) `findOne()` returns a single document, `find()` returns multiple documents

Explanation: `findOne()` retrieves the first document that matches the filter, whereas `find()` returns an array of all matching documents.

232. **In Mongoose, how would you specify the schema type for an array of numbers?**

A) `type: [Number]`

B) `type: { array: Number }`

C) `type: Array[Number]`

D) `type: Array(Number)`

Answer: A) `type: [Number]`

Explanation: In Mongoose, you define an array of numbers by specifying `type: [Number]` in the schema. This indicates that the field will be an array containing `Number` type elements.

233. **What is the purpose of `mongoose.Schema.Types.Date` in Mongoose schemas?**

A) To store a timestamp

B) To define a custom data type

C) To define a date or time value

D) To store an array of dates

Answer: C) To define a date or time value

Explanation: `mongoose.Schema.Types.Date` is used to define a field that holds a date or time value. This can be used to store birthdates, event times, etc.

234. **Which of the following methods in Mongoose is used to update a document's values?**

A) `updateOne()`

B) `findOneAndUpdate()`

C) `updateMany()`

D) All of the above

Answer: D) All of the above

Explanation: All the methods listed—`updateOne()`, `findOneAndUpdate()`, and `updateMany()`—are used for updating documents in Mongoose. The difference is in how they operate: `updateOne()` updates the first document that matches, `findOneAndUpdate()` also returns the updated document, and `updateMany()` updates multiple documents.

235. **Which of the following methods in Mongoose is used to remove a document from a collection?**

A) `Model.deleteOne()`

B) `Model.remove()`

C) `Model.removeAll()`

D) `Model.deleteMany()`

Answer: A) `Model.deleteOne()`

Explanation: `Model.deleteOne()` is used to remove a single document

from the collection based on a filter. `Model.remove()` is an older method and is now considered deprecated.

236. **How do you define a reference to another document (relationship) in Mongoose?**

   A) `type: Object`

   B) `type: Schema.Types.ObjectId`

   C) `type: [Schema.Types.ObjectId]`

   D) `type: Ref`

   Answer: B) `type: Schema.Types.ObjectId`

   Explanation: In Mongoose, references to other documents are defined using `Schema.Types.ObjectId`. This is typically used in conjunction with `populate()` to retrieve related documents.

237. **Which method is used to execute a Mongoose query and return a promise?**

   A) `exec()`

   B) `then()`

   C) `done()`

   D) `callback()`

   Answer: A) `exec()`

   Explanation: The `exec()` method is used to execute a query and return a promise. This allows you to use `then()` and `catch()` for handling asynchronous operations in Mongoose queries.

238. **What is the purpose of the `mongoose.Schema.Types.Mixed` type?**

   A) To store arbitrary types of data

   B) To define a reference to another schema

   C) To define a field that is required

   D) To define a date-time field

   Answer: A) To store arbitrary types of data

   Explanation: `mongoose.Schema.Types.Mixed` allows you to store arbitrary data types within a field, meaning that the field can contain anything like an object, array, or any data type.

239. **In Mongoose, how do you create an index for a field?**

   A) `schema.index({ fieldName: 1 })`

   B) `schema.createIndex({ fieldName: 1 })`

   C) `Model.index({ fieldName: 1 })`

   D) `Model.createIndex({ fieldName: 1 })`

   Answer: A) `schema.index({ fieldName: 1 })`

Explanation: The `schema.index({ fieldName: 1 })` method is used to define an index on a field to optimize query performance. The 1 indicates ascending order for the index.

240. Which of the following is the correct way to define a custom validation function for a Mongoose field?

A) `field: { validate: customValidator }`

B) `field: { customValidate: customValidator }`

C) `field: { validate: [customValidator] }`

D) `field: { validate: function() { ... } }`

Answer: A) `field: { validate: customValidator }`

Explanation: To define a custom validator in Mongoose, you can use the `validate` property and specify a custom function or validator. The function can perform validation logic and return a boolean or a promise.

241. Which Mongoose method is used to create a document from a model?

A) `Model.create()`

B) `Model.createDocument()`

C) `Model.save()`

D) `Model.insert()`

Answer: A) `Model.create()`

Explanation: `Model.create()` is used to instantiate a document and insert it into the database. It accepts an object with the data to be inserted.

242. What happens if you don't call `next()` in a Mongoose middleware function?

A) The middleware chain continues as usual

B) The document will be saved even if validation fails

C) The middleware chain will be blocked, and the operation will not continue

D) An error will be thrown

Answer: C) The middleware chain will be blocked, and the operation will not continue

Explanation: If `next()` is not called within a middleware function, the operation (such as saving a document) will not continue, effectively blocking further middleware or database operations.

243. Which of the following methods is used to return a promise from Mongoose queries?

A) `then()`

B) `exec()`

C) `finally()`

D) `next()`

Answer: B) `exec()`

Explanation: The `exec()` method is used to execute Mongoose queries and returns a promise. This allows for easier asynchronous handling using `.then()` and `.catch()`.

244. What does the `validate()` method do in Mongoose?

A) It performs validation on a single document

B) It updates a document in the database

C) It validates a query

D) It creates a new schema

Answer: A) It performs validation on a single document

Explanation: The `validate()` method in Mongoose runs all the validations defined in the schema for a single document. If validation fails, it returns an error.

245. How do you update a document's fields while returning the updated document in Mongoose?

A) `findOneAndUpdate()` with `{ new: true }`

B) `findOneAndUpdate()` with `{ returnUpdated: true }`

C) `update()` with `{ new: true }`

D) `updateOne()` with `{ returnUpdated: true }`

Answer: A) `findOneAndUpdate()` with `{ new: true }`

Explanation: The `{ new: true }` option in `findOneAndUpdate()` ensures that the updated document is returned rather than the original one.

246. Which method is used to populate a referenced document in Mongoose?

A) `populate()`

B) `ref()`

C) `populateReference()`

D) `include()`

Answer: A) `populate()`

Explanation: The `populate()` method in Mongoose is used to replace the ObjectId references in documents with the actual referenced documents from other collections.

247. In Mongoose, how do you specify a default value for a field in the schema?

A) `field: { default: value }`

B) `field: { value: default }`

C) `field: { initialValue: value }`

D) `field: value`

Answer: A) `field: { default: value }`

Explanation: To set a default value for a field in Mongoose, you use `default: value` inside the schema definition. This value will be applied if the field is not provided during document creation.

248. **How can you prevent saving an empty string as a value for a field in Mongoose?**

A) `required: true`

B) `minlength: 1`

C) `trim: true`

D) `validate: value => value.length > 0`

Answer: B) `minlength: 1`

Explanation: By setting `minlength: 1`, you prevent saving an empty string (`""`) because it enforces that the string must have at least one character.

249. **What will be the result of calling `Model.findOne()` on an empty MongoDB collection?**

A) `null` is returned

B) An empty array is returned

C) An empty document is returned

D) An error is thrown

Answer: A) `null` is returned

Explanation: When you call `Model.findOne()` on a collection that has no matching documents, it returns `null`. It does not throw an error or return an empty array.

250. **Which of the following Mongoose features allows you to automatically populate a field with referenced documents without explicitly calling `populate()`?**

A) `autoPopulate`

B) `populateOnFind`

C) `populate` with `{ auto: true }`

D) `refPath`

Answer: A) `autoPopulate`

Explanation: The `autoPopulate` feature in Mongoose can be enabled through a plugin or schema option to automatically populate a referenced field when a query is executed, eliminating the need to manually call `populate()`.