DUMPSARENA

Developer Essentials for FileMaker 13

FileMaker FM0-308

Version Demo

Total Demo Questions: 13

Total Premium Questions: 132

Buy Premium PDF

https://dumpsarena.co

sales@dumpsarena.co

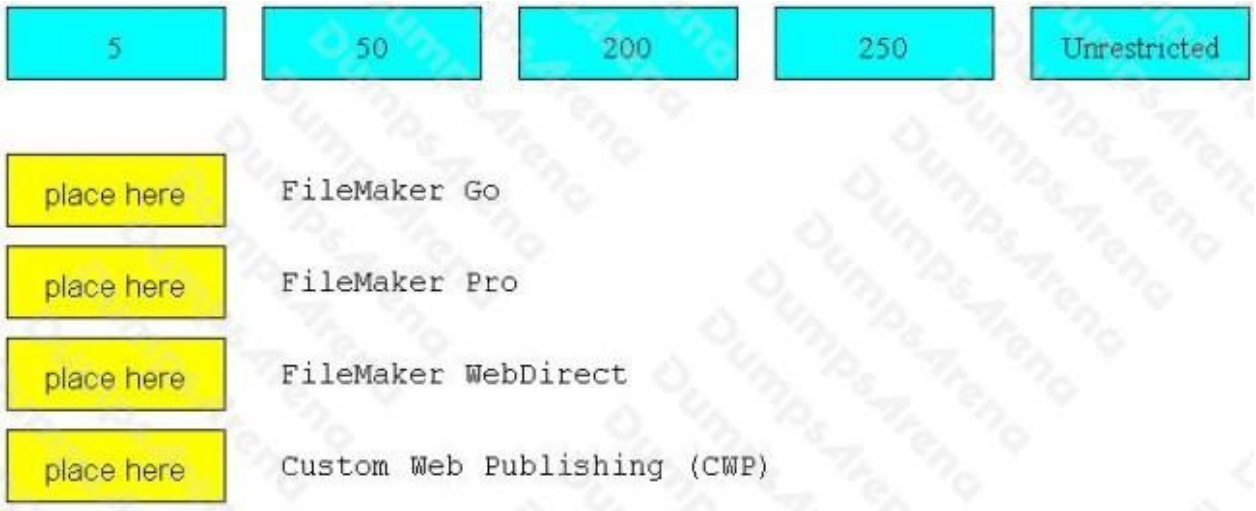
sales@dumpsarena.co
dumpsarena.co

QUESTION NO: 1 - (DRAG DROP)

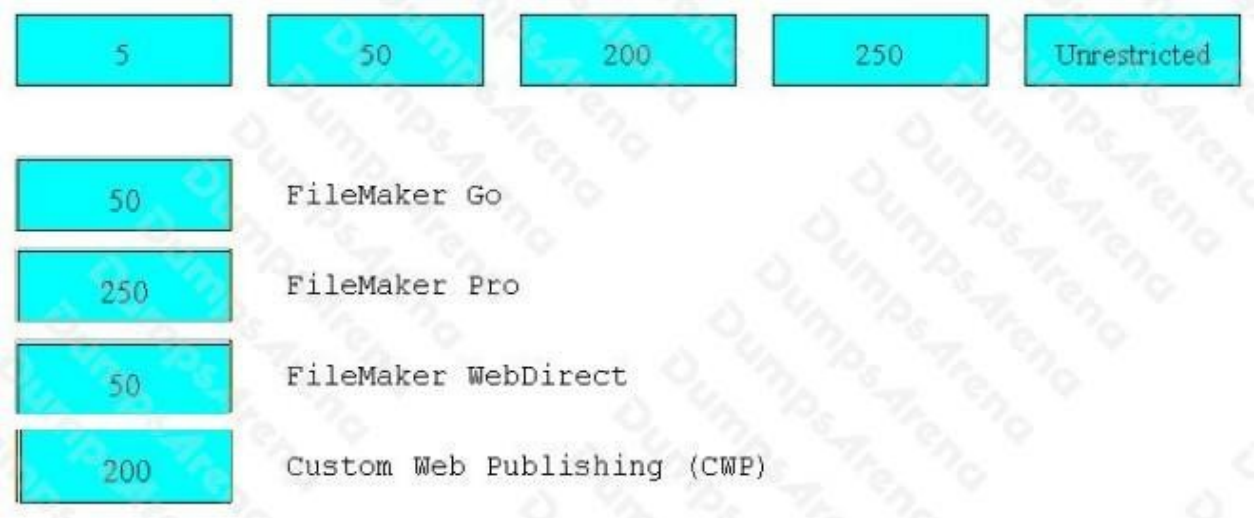
Match the FileMaker Server 13 hosting technology with maximum tested simultaneous connections according to the FileMaker Server 13 Technical Specifications.

(Options may be used more than once.)

Select and Place:



ANSWER:



Explanation:

The correct matching reflects the separate connection limits FileMaker Server 13 was tested to support for each hosting technology. FileMaker Go is matched with 50 simultaneous connections. A FileMaker Go connection is a hosted-file session made from the mobile client, and this published tested limit is distinct from the desktop client limit. FileMaker Pro is matched with 250 simultaneous connections, representing the larger tested concurrent-session capacity for FileMaker Pro users connecting directly to hosted database files.

FileMaker WebDirect is matched with 50 simultaneous connections. WebDirect delivers FileMaker layouts through a supported web browser, so its tested capacity is listed independently rather than being combined with FileMaker Pro or mobile sessions. Custom Web Publishing (CWP) is matched with 200 simultaneous connections. CWP accesses hosted data through web technologies and the FileMaker Server web publishing engine, which has its own published tested connection capacity.

These figures are maximum *tested* simultaneous connections, not a promise that every deployment will reach the same practical throughput. Actual capacity can vary with server hardware, database design, scripting, network conditions, layout complexity, and the type of client activity. The important exam distinction is that the values identify the tested limits by access

method: direct FileMaker Pro access, FileMaker Go access, browser-based WebDirect access, and Custom Web Publishing access. FileMaker documentation continues to describe hosted access technologies and deployment planning in its FileMaker Server Help, while the product documentation archive is available from Claris Help Center Archive.

QUESTION NO: 2

Which two statements are true about multiple find requests in FileMaker Pro 13? (Choose two.)

A. Criteria entered in different fields within the same find request are treated as an AND search.

B. Separate find requests are treated as an OR search.

C. Separate find requests are treated as an AND search.

D. A find request can contain criteria for only one field.

E. All find requests must return at least one record for a find to succeed.

ANSWER: A B

Explanation:

Criteria entered in different fields within the same find request are treated as an AND search is correct. A record must satisfy every criterion in that one request to be included in the found set. For example, a request for State = CA and Status = Active finds only active records in California.

Separate find requests are treated as an OR search is also correct. FileMaker includes records that satisfy any one of the requests. For example, one request for State = CA and a second request for State = OR finds records from either state. See Claris FileMaker Pro Help: Finding records.

QUESTION NO: 3

Which two statements are true about the Perform Script on Server script step in FileMaker Pro 13? (Choose two.)

A. The script is executed in a separate FileMaker Server session.

B. A script parameter can be retrieved by the server-side script with Get (ScriptParameter).

C. The script runs on the client computer after FileMaker Server validates the script.

D. The script can run when the file is opened locally and is not hosted by FileMaker Server.

E. The script always uses the current client window and its active layout object.

ANSWER: A B

Explanation:

The script is executed in a separate FileMaker Server session is correct. Perform Script on Server sends the specified script to FileMaker Server, allowing work to be processed on the server rather than in the FileMaker Pro, FileMaker Go, or FileMaker WebDirect client session.

A script parameter can be retrieved by the server-side script with Get (ScriptParameter) is also correct. The calling script can provide a parameter, and the server-side script can obtain that parameter in the same way that other performed scripts obtain a script parameter. A result returned by the server-side script can subsequently be retrieved by the calling script with `Get ( ScriptResult )`. See Claris FileMaker Pro Help: Perform Script on Server.

QUESTION NO: 4

The following formula in a calculation field returns a Text result:

Let(tmp = “2 * 5“ Evaluate(Quote (tmp)))

What value will be returned?

A. 10

B. 2 * 5

C. “10”, quotes included

D. “2 * 5”, quotes included

ANSWER: B

Explanation:

`2 * 5` is returned as text. The variable `tmp` contains the text string `2 * 5`, rather than the result of multiplying two numbers. The `Quote` function then produces a properly quoted FileMaker expression representing that text—for example, it transforms the value into an expression equivalent to `"2 * 5"`. When `Evaluate` evaluates that expression, the quotation marks cause the expression to be interpreted as a text literal. Therefore, `Evaluate` returns the characters `2 * 5`; it does not perform the multiplication. Although the calculation field is specified to return Text, the essential behavior comes from the quoted expression passed to `Evaluate`: it evaluates to text and the quotation-mark characters are syntax used during evaluation, not characters included in the resulting value. FileMaker’s `Quote` documentation describes its role in enclosing text in quotation marks and escaping embedded quotation marks, while the `Evaluate` documentation explains that it evaluates text as a calculation expression. See Claris FileMaker Pro Help: Quote and Claris FileMaker Pro Help: Evaluate.

QUESTION NO: 5

Which two statements are true about OnTimer triggers? (Choose two.)

A. They only run when the application is idle.

B. OnTimer triggers are configured in the File Options dialog.

C. Different OnTimer scripts can be installed in different windows.

D. Days and times can be specified when the OnTimer script activates.

ANSWER: A C

Explanation:

“They only run when the application is idle” is correct because an OnTimer script is not allowed to interrupt activity already being performed by FileMaker Pro. When its interval elapses, FileMaker waits until it is idle before running the installed timer script. This behavior makes an OnTimer trigger useful for background-style interface tasks, such as periodically refreshing displayed data or checking for a status change, without interrupting the user’s current operation or an active script.

“Different OnTimer scripts can be installed in different windows” is also correct. The *Install OnTimer Script* script step installs a timer script for the current window, and each window can have its own installed OnTimer script. A solution can therefore use one timer-driven behavior in one window and a distinct timer-driven behavior in another window. The timer remains associated with that window until it is replaced, uninstalled, or the relevant window context is closed. This window-specific behavior lets developers tailor timed actions to the purpose of each open interface. See Claris documentation for Install OnTimer Script and the script trigger overview.

QUESTION NO: 6

Which two statements are true about Touch keyboard types? (Choose two.)

A. The keyboard can be suppressed.

B. The setting has no impact on clients using FileMaker Pro.

C. The keyboard assigned to a field can be installed via a script.

D. The default keyboard for a field can be set under Field Options.

E. The default keyboard for new layout objects is Default for Data Type.

ANSWER: A B

Explanation:

The keyboard can be suppressed. is correct because a layout field's touch-keyboard setting can be configured so that FileMaker Go does not automatically display a touch keyboard when the user enters that field. This is useful when a solution supplies its own on-screen controls, such as a custom numeric keypad, barcode-entry interface, or button-driven data-entry workflow. The setting controls the keyboard presented by FileMaker Go for the active layout object; it does not change the field's underlying data type or validation behavior.

The setting has no impact on clients using FileMaker Pro. is also correct. Touch keyboard types are intended for FileMaker Go on iOS/iPadOS, where the on-screen keyboard can be tailored to the expected form of input, such as text, numbers, email addresses, URLs, or telephone numbers. FileMaker Pro uses the desktop operating system's standard keyboard input instead, so a touch-keyboard selection on a layout does not alter keyboard behavior for FileMaker Pro users. Claris documents touch-keyboard configuration as a FileMaker Go-focused layout feature; see Specifying touch keyboard types and the FileMaker Pro 13 Help archive.

QUESTION NO: 7

When writing a script that will edit a record in FileMaker Pro 13, which test will determine if a different user had the record locked?

A. Check to see if the value of Get (RecordOpenState) is not zero.

B. Set a script variable to the UUID of the record, and then use the formula PatternCount (Get (LockedRecordIDs) ; $myVar) > 0.

C. Attempt to modify a field with global storage, and then test to see if an error occurred.

D. Use the Open Record/Request script step, and then a test to see if an error occurred.

E. Attempt to set a script variable to a value from the record, and then test to see if an error occurred.

F. Include an On Error script step after the Set Field step but before the Commit Record/Request step

ANSWER: D

Explanation:

Use the Open Record/Request script step, and then a test to see if an error occurred. This is the appropriate test because FileMaker record locking is pessimistic: before a script edits a record, it can explicitly attempt to open that record for editing. If another user already has the record open and locked, the Open Record/Request step cannot obtain the lock. The script can then immediately evaluate `Get ( LastError )`; error 301 identifies the condition that the record is in use by another user. A typical implementation enables Set Error Capture, runs Open Record/Request, stores `Get ( LastError )` in a variable, and branches when the value is 301. When the open succeeds, the script can safely perform its Set Field operations and commit the record. Explicitly requesting the record lock before making edits provides a reliable, intentional point at which to detect the multi-user conflict and present a useful message or retry behavior. Claris documents the lock-request behavior of Open Record/Request and the returned status values through Get (LastError).

QUESTION NO: 8

Which statement is true about the Filter Portal Records feature in FileMaker Pro 13?

A. The portal filter formula must refer to at least one field in the immediately related table.

B. The feature requires one or more additional relationships to be added to the Relationships Graph.

C. The portal filter formula may not refer to any globally stored fields, summary fields, or unstored calculation fields.

D. There can be one unique filter formula applied for each portal state (Normal State, In Focus, Hover, or Pressed).

E. Aggregate functions (e.g., Sum, Count) based on the portal's relationship will not be affected by the portal's filter formula.

ANSWER: E

Explanation:

Aggregate functions (e.g., Sum, Count) based on the portal's relationship will not be affected by the portal's filter formula. A portal filter controls which records from an already related record set are displayed in that specific portal; it does not alter the underlying relationship, its join criteria, or the complete set of records that the relationship makes available. Consequently, an aggregate calculation such as `Sum ( RelatedTable::Amount )`, or a Count calculation based on related records, continues to evaluate the full related set defined by the relationship rather than only the subset currently visible after portal filtering. This distinction is important when a layout presents a filtered portal alongside totals: the portal can show, for example, only records matching a user-entered condition, while a relationship-based aggregate still reports the total for all related records. If a solution needs a total that reflects the portal filter, the calculation must explicitly apply equivalent criteria rather than relying on the portal's display filter. Claris documents that portal filtering affects the records displayed in the portal and does not change the relationship itself. See Claris FileMaker Pro Help: Working with portal filters and Claris FileMaker Pro Help: Summary fields.

QUESTION NO: 9

What is the maximum number of recursive calls that can be made utilizing a custom function in FileMaker Pro 13 Advanced?

A. 10,000

B. 32,000

C. 50,000

D. 64.000

ANSWER: C

Explanation:

50,000 is correct. In FileMaker Pro 13 Advanced, a custom function can call itself recursively, allowing a calculation to repeat by evaluating progressively changed arguments until it reaches its terminating condition. FileMaker Pro 13 Advanced imposes a maximum recursion depth of 50,000 calls for such custom-function recursion. This limit is important when designing custom functions that process lists, parse text, traverse hierarchical data, or perform iterative calculations: the function must include a reliable stopping condition and should be designed so that its expected number of calls remains within the supported limit. Recursive custom functions are created and managed through the Custom Functions area of the calculation tools available in FileMaker Pro Advanced, rather than in FileMaker Pro alone. FileMaker's calculation documentation describes custom functions as reusable calculations and supports their use in expressions throughout a solution. For background on custom functions in the FileMaker 13 documentation, see the FileMaker Pro 13 function reference. Claris's current documentation also discusses recursion limits and the handling of recursive calculations in SetRecursion.

QUESTION NO: 10

A new solution is created with lwo tables, Client and Contracts. The only modification made on the Relationships Graph is to establish a relationship between them which is set to delete related records in Contracts when a record in Client is deleted.



What happens when the Client table occurrence is deleted from the Relationships Graph?

A. Since it is the only occurrence of the Client table on the graph, it cannot be deleted.

B. Any records in the Contracts table that were related to matching Client records are deleted.

C. The formula in the calculation field Name Full (defined as Name First & " " & Name Last) will be broken.

D. Any records in the Contracts table that were related to matching Client records will have values in the Client ID field deleted.

ANSWER: C

Explanation:

The formula in the calculation field Name Full (defined as Name First & " " & Name Last) will be broken. FileMaker calculation expressions can contain references that are resolved in the context of a table occurrence. In this case, the calculation depends on fields accessed through the Client table occurrence. Removing that occurrence from the Relationships Graph removes the occurrence used by the calculation's field references, so FileMaker can no longer resolve those references correctly. The calculation definition is consequently left with broken references and must be repaired by updating it to use an available table occurrence or otherwise restoring the required context.

Deleting a table occurrence is a schema-level change to the Relationships Graph. It removes the occurrence and its graph relationships, rather than deleting the underlying base table or performing a record deletion. Therefore, the delete-related-records relationship setting is not triggered: that setting applies when a Client *record* is deleted, not when its table occurrence is removed from the graph. Claris documents both the role of table occurrences in relationship graphs and the effect of deleting them in the FileMaker Pro Help: Working with the relationships graph and Deleting table occurrences.

QUESTION NO: 11

Which two statements are true about the Status Toolbar when using FileMaker Pro 13? (Choose two.)

A. Record navigation via the Status Toolbar can be prevented by installing a Custom Menu.

B. The name and/or function of a status toolbar button can be changed by installing a Custom Menu.

C. The ability to add or remove buttons on the Status Toolbar can be prevented by installing a Custom Menu.

D. Users whose privilege set has Available menu commands set to Minimum cannot customize the Status Toolbar

E. The initial settings of the Status Toolbar are stored with the database file and shared with all users of the hosted file.

F. Customization of the Status Toolbar can be prevented by running a script which contains the Show/Hide Status Area [Lock; Show] script step.

ANSWER: A B

Explanation:

Record navigation via the Status Toolbar can be prevented by installing a Custom Menu. FileMaker custom menu sets let a developer control the availability of FileMaker menu commands. Status Toolbar controls are tied to FileMaker commands, including commands used to move among records. By omitting or disabling the relevant record-navigation commands in the installed custom menu set, a developer can prevent users from navigating records through those Status

Toolbar controls. This is useful when navigation must occur only through scripted buttons, a controlled interface, or a constrained workflow.

The name and/or function of a status toolbar button can be changed by installing a Custom Menu. A custom menu item can replace a standard command with a script, thereby changing what the associated command does when invoked from the Status Toolbar. Custom menus also support customized menu-item naming, allowing the interface presented to users to reflect the solution's terminology and workflow. These capabilities allow a FileMaker developer to tailor standard interface behavior without modifying FileMaker Pro itself. See Claris documentation on custom menus and customizing the Status Toolbar.

QUESTION NO: 12

Which statement is true about enabling database encryption in FileMaker Pro 13 Advanced?

A. Database encryption will secure network traffic when using peer-to-peer sharing.

B. External container data will automatically be set to use Secure Storage when encrypting the database.

C. To prevent repeated encryption password entry, two unhosted files must have both the same encryption password and Shared ID.

D. If an encrypted file is hosted on FileMaker Server 13, every user who opens the file will be required to enter the encryption password.

ANSWER: C

Explanation:

To prevent repeated encryption password entry, two unhosted files must have both the same encryption password and Shared ID. FileMaker encryption protects the database file at rest and requires an encryption password when an encrypted file is opened directly. A Shared ID is used to associate encrypted files that belong to the same solution. When the files also use the identical encryption password, FileMaker Pro can use the password already supplied for one file when it opens another related, unhosted encrypted file in that Shared ID group. This is particularly important for multi-file solutions, where opening a primary file may cause FileMaker Pro to open related files, such as a data or utility file. Configuring both values consistently provides a smoother opening experience while retaining encryption for each file. The Shared ID alone is not sufficient: the encryption password must match as well. Conversely, matching passwords without the same Shared ID does not establish the intended shared encrypted-file relationship. FileMaker's encryption workflow and its requirements for opening encrypted files are documented in the FileMaker Pro 13 Help: Encrypting database files and the FileMaker Pro 13 Help: Opening encrypted database files.

QUESTION NO: 13

Which three statements are true about FileMaker WebDirect? (Choose three.)

A. It is possible to create a custom FileMaker WebDirect Home Page.

B. It is possible to create a URL that links directly to a FileMaker WebDirect solution, bypassing the home page.

C. The appearance of most FileMaker WebDirect pages can be altered using Cascading Style Sheets (CSS]

D. It is possible to limit the list of databases visible on the FileMaker WebDirect Home Page to just the specific databases the current user may access.

E. It is possible to display databases from multiple instances of FileMaker Server 13 on the default FileMaker WebDirect Home Page if the servers have been configured to work with a directory service.

ANSWER: A B D

Explanation:

It is possible to create a custom FileMaker WebDirect Home Page because FileMaker Server's WebDirect deployment includes a default launch page that can be customized to provide a branded entry point or organization-specific navigation for hosted solutions. It is possible to create a URL that links directly to a FileMaker WebDirect solution, bypassing the home page, by using a WebDirect URL that identifies the hosted database and, when needed, a startup script or layout. This supports distributing direct links to users rather than requiring them to choose a file from the launch page.

It is also possible to limit the list of databases visible on the FileMaker WebDirect Home Page to just the specific databases the current user may access. WebDirect authenticates users against the account credentials configured in each hosted file, and the launch experience can show only files for which the authenticated user has access. This lets a single FileMaker Server deployment host solutions for different user groups while presenting an appropriate, permission-aware list of available databases. See the FileMaker Server 13 Help and the Claris FileMaker WebDirect Guide.