ACD300$^{Q\&As}$

Appian Certified Lead Developer

QUESTION 1

You are designing a process that is anticipated to be executed multiple times a day. This process retrieves data from an external system and then calls various utility processes as needed. The mam process will not use the results of the utility processes, and there are no user forms anywhere.

Which design choice should be used to start the utility processes and minimize the load on the execution engines?

A. Use the Start Process Smart Service to start the utility processes.

B. Start the utility processes via a subprocess synchronously.

C. Use Process Messaging lo star! the utility process.

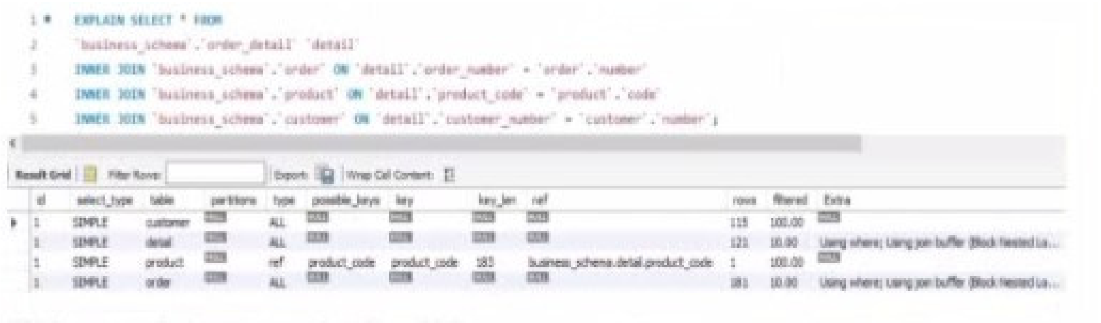
D. Start the utility processes via a subprocess asynchronously

Correct Answer: C

To design a process that is anticipated to be executed multiple times a day, that retrieves data from an external system and then calls various utility processes as needed, you should use Process Messaging to start the utility process and minimize the load on the execution engines. Process Messaging is a feature that allows you to send and receive messages between processes in Appian. By using Process Messaging, you can start the utility process asynchronously, which means that the main process does not have to wait for the utility process to finish before continuing. This way, you can improve the performance and scalability of your process design, and reduce the load on the execution engines. The other options are not as effective. Option A, using the Start Process Smart Service to start the utility processes, would also start the utility process asynchronously, but it would require more configuration and maintenance than Process Messaging. Option B, starting the utility processes via a subprocess synchronously, would start the utility process as a part of the main process flow, which means that the main process would have to wait for the utility process to finish before continuing. This would reduce the performance and scalability of your process design, and increase the load on the execution engines. Option D, starting the utility processes via a subprocess asynchronously, would also start the utility process as a part of the main process flow,but it would not wait for the utility process to finish before continuing. However, this option would still create more overhead than Process Messaging, as it would create more instances of processes in Appian.

QUESTION 2

Review the following resultof an explain statement: Which two conclusions can you draw from this?



```
EXPLAIN SELECT * FROM
`business_schema`.`order_detail` `detail`
INNER JOIN `business_schema`.`order` ON `detail`.`order_number` = `order`.`number`
INNER JOIN `business_schema`.`product` ON `detail`.`product_code` = `product`.`code`
INNER JOIN `business_schema`.`customer` ON `detail`.`customer_number` = `customer`.`number`;
```

id	select_type	table	partitions	type	possible_keys	key	key_len	ref	rows	filtered	Extra
1	SIMPLE	customer	NULL	ALL	NULL	NULL	NULL	NULL	115	100.00	NULL
1	SIMPLE	detail	NULL	ALL	NULL	NULL	NULL	NULL	121	10.00	Using where; Using join buffer (Block Nested Lo...
1	SIMPLE	product	NULL	ref	product_code	product_code	183	business_schema.detail.product_code	1	100.00	NULL
1	SIMPLE	order	NULL	ALL	NULL	NULL	NULL	NULL	181	10.00	Using where; Using join buffer (Block Nested Lo...

A. The request is good enough to support a high volume of data. but could demonstrate some limitations if the developer queries information related to the product

B. The worst join isthe one between the table order_detail and order.

C. The join between the tables order_detail, order and customerneeds to be tine-tuned due to indices.

D. The join between the tables 0rder_detail and productneeds to be fine-tuned due to Indices

E. The worst join is the one between the table order_detail and customer

Correct Answer: DE

D. The join between the tables order_detail and product needs to be fine-tuned due to Indices. This is correct because the result of the explain statement showsthat the join between these two tables has a high cost of 0.99, which indicates that it is inefficient and needs to be fine-tuned. One possible reason for the high cost is that there are no indices on the columns that are used for joining these two tables, which leads to a full table scan. Therefore, creating indices on these columns could improve the performance of this join. E. The worst join is the one between the table order_detail and customer. This is correct because the result of the explain statement shows that the join between these two tables has a very high cost of 1.00, which indicates that it is the worst join in terms of efficiency and needs to be fine-tuned. One possible reason for the high cost is that there are no indices on the columns that are used for joining these two tables, which leads to a full table scan. Therefore, creating indices on these columns could improve the performance of this join. The other options are incorrect for the following reasons:

A. The request is good enough to support a high volume of data, but could demonstrate some limitations if the developer queries information related to the product. This is incorrect because the request is not good enough to support a high volume of data, as it has two joins with very high costs that need to be fine- tuned. Moreover, querying information related to the product would not necessarily cause any limitations, as long as the join between order_detail and product is optimized.

B. The worst join is the one between the table order_detail and order. This is incorrect because the result of the explain statement shows that the join between these two tables has a low cost of 0.01, which indicates that it is efficient and does not need to be fine-tuned.

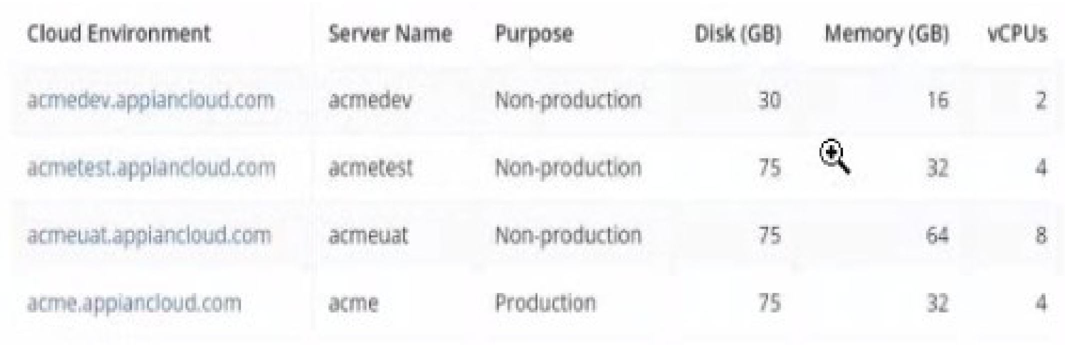
C. The join between the tables order_detail, order and customer needs to be fine- tuned due to indices. This is incorrect because there is no such join between three tables in the result of the explain statement. There are only two joins: one between order_detail and order, and another between order_detail and customer. Each of these joins needs to be fine-tuned separately due to indices.

QUESTION 3

You are on a protect with an application that has been deployed to Production and is live with users. The client wishes to increase the number of active users.

You need to conduct load testing to ensure Production can handle the increased usage

Review the specs for four environments in the following image.



Cloud Environment	Server Name	Purpose	Disk (GB)	Memory (GB)	vCPUs
acmedev.appiancloud.com	acmedev	Non-production	30	16	2
acmetest.appiancloud.com	acmetest	Non-production	75	32	4
acmeuat.appiancloud.com	acmeuat	Non-production	75	64	8
acme.appiancloud.com	acme	Production	75	32	4

Which environment should you use for load testing7

A. acmeuat

B. acmedev

C. acme

D. acmetest

Correct Answer: C

The best environment to use for load testing is acme, which is the production environment. This is because load testing should be performed on an environment that is as close as possible to the actual production environment, in terms of hardware, software, configuration, data, and user behavior. This way, the results of the load testing will be more realistic and reliable, and can help to identify and resolve any performance issues or bottlenecks before increasing the number of active users. Verified References: Appian Documentation, section "Load Testing".

QUESTION 4

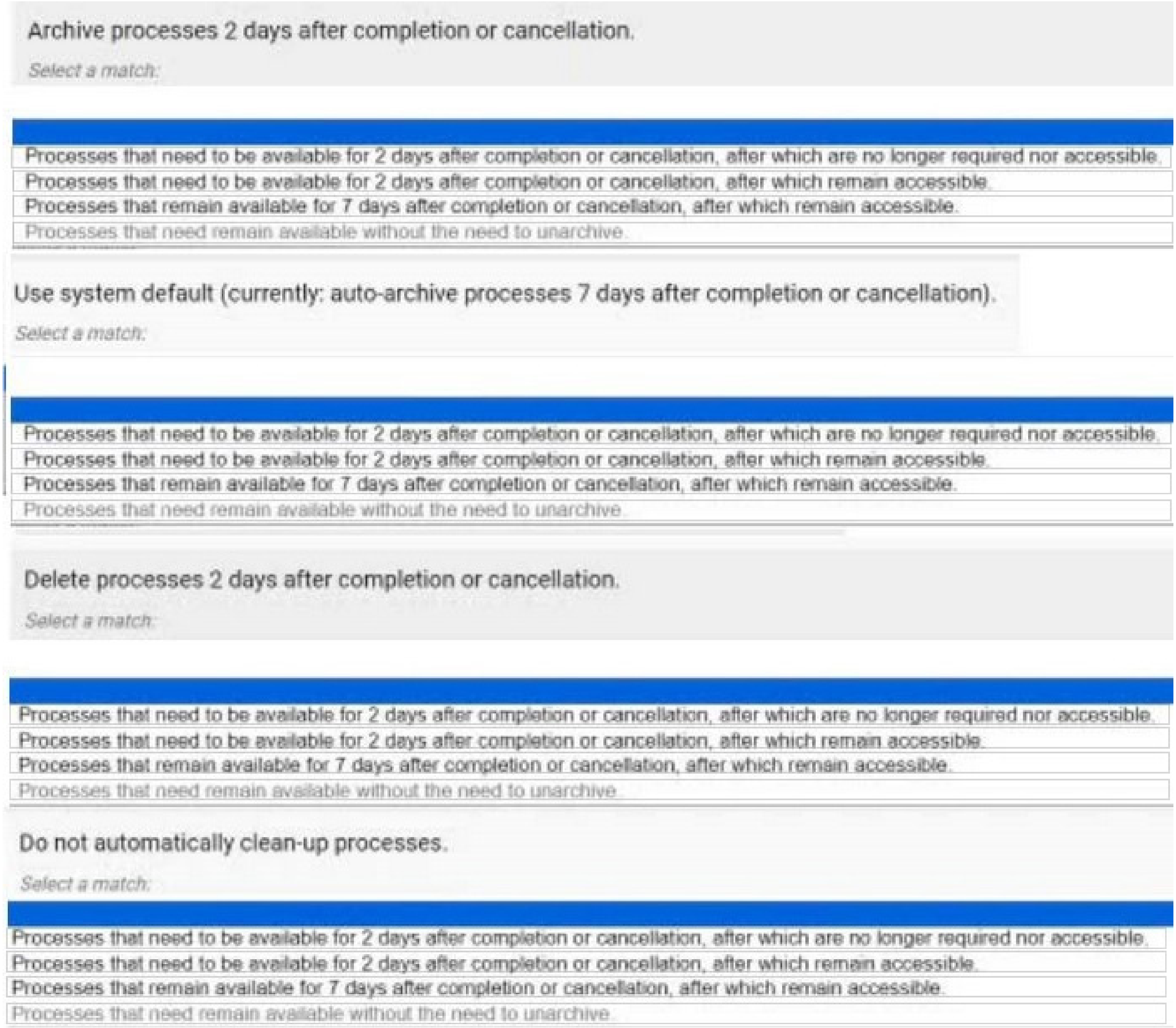
HOTSPOT

You are deciding the appropriate process model data management strategy.

For each requirement. match the appropriate strategies to implement. Each strategy will be used once.

Note: To change your responses, you may deselect your response by clicking the blank space at the top of the selection list.

Hot Area:



Archive processes 2 days after completion or cancellation.

Select a match:

- Processes that need to be available for 2 days after completion or cancellation, after which are no longer required nor accessible.
- Processes that need to be available for 2 days after completion or cancellation, after which remain accessible.
- Processes that remain available for 7 days after completion or cancellation, after which remain accessible.
- Processes that need remain available without the need to unarchive.

Use system default (currently: auto-archive processes 7 days after completion or cancellation).

Select a match:

- Processes that need to be available for 2 days after completion or cancellation, after which are no longer required nor accessible.
- Processes that need to be available for 2 days after completion or cancellation, after which remain accessible.
- Processes that remain available for 7 days after completion or cancellation, after which remain accessible.
- Processes that need remain available without the need to unarchive.

Delete processes 2 days after completion or cancellation.

Select a match:

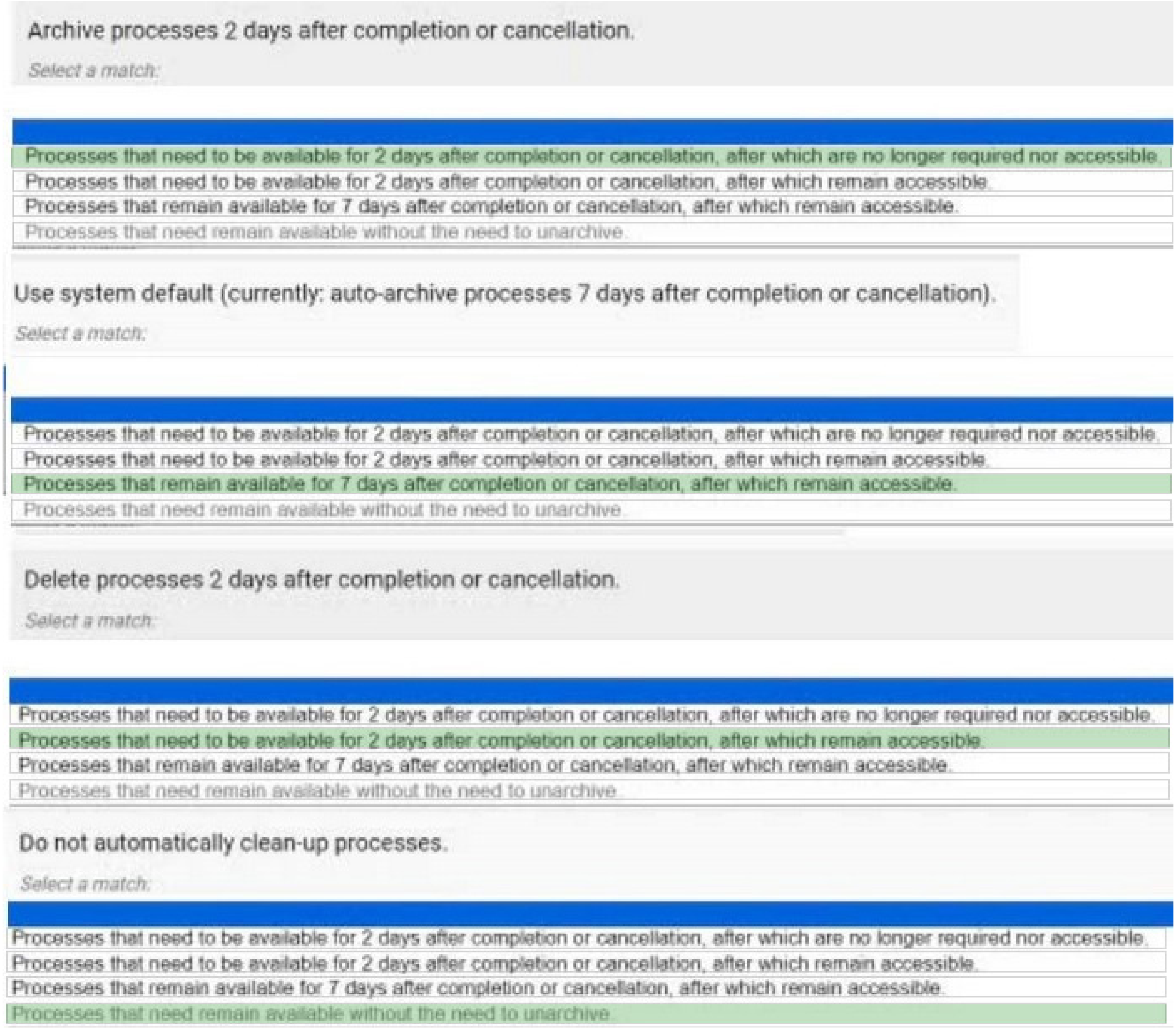
- Processes that need to be available for 2 days after completion or cancellation, after which are no longer required nor accessible.
- Processes that need to be available for 2 days after completion or cancellation, after which remain accessible.
- Processes that remain available for 7 days after completion or cancellation, after which remain accessible.
- Processes that need remain available without the need to unarchive.

Do not automatically clean-up processes.

Select a match:

- Processes that need to be available for 2 days after completion or cancellation, after which are no longer required nor accessible.
- Processes that need to be available for 2 days after completion or cancellation, after which remain accessible.
- Processes that remain available for 7 days after completion or cancellation, after which remain accessible.
- Processes that need remain available without the need to unarchive.

Correct Answer:



Archive processes 2 days after completion or cancellation.

Select a match:

- [✓ Processes that need to be available for 2 days after completion or cancellation, after which are no longer required nor accessible.]
- Processes that need to be available for 2 days after completion or cancellation, after which remain accessible.
- Processes that remain available for 7 days after completion or cancellation, after which remain accessible.
- Processes that need remain available without the need to unarchive.

Use system default (currently: auto-archive processes 7 days after completion or cancellation).

Select a match:

- Processes that need to be available for 2 days after completion or cancellation, after which are no longer required nor accessible.
- Processes that need to be available for 2 days after completion or cancellation, after which remain accessible.
- [✓ Processes that remain available for 7 days after completion or cancellation, after which remain accessible.]
- Processes that need remain available without the need to unarchive.

Delete processes 2 days after completion or cancellation.

Select a match:

- Processes that need to be available for 2 days after completion or cancellation, after which are no longer required nor accessible.
- [✓ Processes that need to be available for 2 days after completion or cancellation, after which remain accessible.]
- Processes that remain available for 7 days after completion or cancellation, after which remain accessible.
- Processes that need remain available without the need to unarchive.

Do not automatically clean-up processes.

Select a match:

- Processes that need to be available for 2 days after completion or cancellation, after which are no longer required nor accessible.
- Processes that need to be available for 2 days after completion or cancellation, after which remain accessible.
- Processes that remain available for 7 days after completion or cancellation, after which remain accessible.
- [✓ Processes that need remain available without the need to unarchive.]

Requirement: Archive processes 2 days after completion or cancellation. Correct match: A. Processes that need to be available for 2 days after completion or cancellation, after which are no longer required nor accessible Exact explanation of correct match taken from Appian Documentation: This strategy is called "Archive after 2 days" and it is one of the options for process model data management in Appian. This strategy means that processes that complete or cancel will remain available for 2 days, after which they will be archived and no longer accessible. This strategy can help reduce the size of the process database and improve the performance of process reporting. Requirement: Use system default (currently auto-archive processes 7 days after completion or cancellation). Correct match: C. Processes that remain available for 7 days after completion or cancellation, after which are archived when accessed Exact explanation of correct match taken from Appian Documentation: This strategy is called "Use system default" and it is one of the options for process model data management in Appian. This strategy means that processes that complete or cancel will remain available for 7 days, after which they will be archived when accessed. This strategy can help balance the availability and performance of process data, as it allows processes to be archived on demand rather than on a fixed schedule. Requirement: Delete processes 2 days after completion or cancellation. Correct match: B. Processes that need to be available for 2 days after completion or cancellation, after which remain accessible Exact explanation of correct match taken from Appian Documentation: This strategy is called "Delete after 2 days" and it is one of the options for process model data management in Appian. This strategy means that processes that complete or cancel will remain available for 2 days, after which they will be deleted and no longer accessible. This strategy can help reduce the size of the process database and improve the performance of process reporting, but it also means that process data will be permanently lost. Requirement: Do not automatically clean-up processes. Correct match: D. Processes that need to remain available without the need to unarchive Exact explanation of correct match taken from Appian Documentation: This strategy is called "Do not automatically clean-up" and it is one of the options for process model data management in Appian. This strategy means that processes that complete or cancel will remain available indefinitely without being archived or deleted. This strategy can help ensure the availability and integrity of process data, but it also means that the process database will grow over time and affect the performance of process reporting.

QUESTION 5

As part of your implementation workflow, users need to retrieve data stored in a third-party Oracle database on an interface. You need to design a way to query this information.

How should you set up this connection and query the data?

A. Configure a Query DataBase node within the process model Then, type in the connection information, as well as a SQL query to execute and return the data in process variables.

B. Configure a limed utility process that queries data from the thirdparty database daily, and stores It in the Applan business database, Then use alqueryEntity eating the Applan data source to retrieve the data.

C. Configure an expression-backed record type, calling an API to retrieve the data from the third-party database. Then, use allqueryRecordType to retrieve the data.

D. in the Administration Console configure the third-party database as a `\\'New Data Source,\\'\\' Then, use alqueryEntity to retrieve the data.

Correct Answer: D

To meet the application requirement of allowing users to navigate throughout the application while maintaining complete visibility in the application structure, and easily navigate to previous locations, you should include a breadcrumbs pattern on applicable interfaces to show the organizational hierarchy. A breadcrumbs pattern is a user interface component that displays the current location of the user within the application, and provides links to the previous levels of the hierarchy. For example, if the user is viewing a product details page, the breadcrumbs pattern could show something like "Home > Products > Product Details". This way, the user can see where they are in the application, and easily go back to any previous level by clicking on the corresponding link. The other options are not as effective. Option A, using Tiles as Cards pattern on the home page to prominently display application choices, would provide a way for users to access different parts of the application from the home page, but it would not show the organizational hierarchy or allow users to navigate to previous locations. Option B, implementing an Activity History pattern to track an organization\\'s activity measures, would provide a way for users to see the recent actions performed by themselves or others in the application, but it would not show the organizational hierarchy or allow users to navigate to previous locations. Option C, implementing a drilldown report pattern to show detailed information about report data, would provide a way for users to explore different levels of data in a report, but it would not show the organizational hierarchy or allow users to navigate to previous locations.

ACD300 Study Guide   ACD300 Exam Questions   ACD300 Braindumps