ACD300$^{Q\&As}$

Appian Certified Lead Developer

QUESTION 1

You are the lead developer for an Appian project, in a backlog refinement meeting You are presented with the following user story.

As a restaurant customer. I need to be able to place my food order online to avoid waiting in line for take out.\\'

Which two functional acceptance criteria would you consider \\'good\\'?

A. The user will click Save, and the order information will be saved in the ORDER table and have audit history

B. The user will receive an email notification when their order is completed.

C. The system mutt handle up to 500 unique orders per day

D. The user cannot submit the form without filling out all required fields.

Correct Answer: BD

Functional acceptance criteria are the conditions that a user story must satisfy to be accepted by the user or stakeholder. They should be specific, measurable, achievable, relevant, and testable. In this case, two functional acceptance criteria that would be considered `good\\' are: The user will receive an email notification when their order is completed. This is a specific, measurable, achievable, relevant, and testable criterion that describes a feature that the user needs to be informed of their order status. The user cannot submit the form without filling out all required fields. This is a specific, measurable, achievable, relevant, and testable criterion that describes afeature that the user needs to provide valid and complete information for their order. The other options are not as good. Option A, the user will click Save, and the order information will be saved in the ORDER table and have audit history, is not a functional acceptance criterion, but rather a technical implementation detail that is not relevant or visible to the user. Option C, the system must handle up to 500 unique orders per day, is not a functional acceptance criterion, but rather a non-functional requirement that describes a performance or quality attribute of the system.

QUESTION 2

You have created a Web API in Appian. with the following URL to call it: https://exampleappiancloud.com/suite/webapi/usef_managefnent/ users ?username=)=john.smith.

Which is the connect syntax forreferring to the user name parameter\\'

A. httpirequest.queryParameters users username

B. httpirequest usees username

C. httpirequest formData username

D. httpirequest queryParameters.username

Correct Answer: D

The correct syntax for referring to the username parameter in the Web API URL is httpirequest.queryParameters.username. This syntax allows you to access the value of the username parameter that is passed in the query string of the URL after the question mark (?). For example, if the URL ishttps://exampleappiancloud.com/suite/webapi/user_management/users?username=john. smith, then httpirequest.queryParameters.username will return john.smith. Verified References: Appian Documentation, section

"Web API".

QUESTION 3

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 4

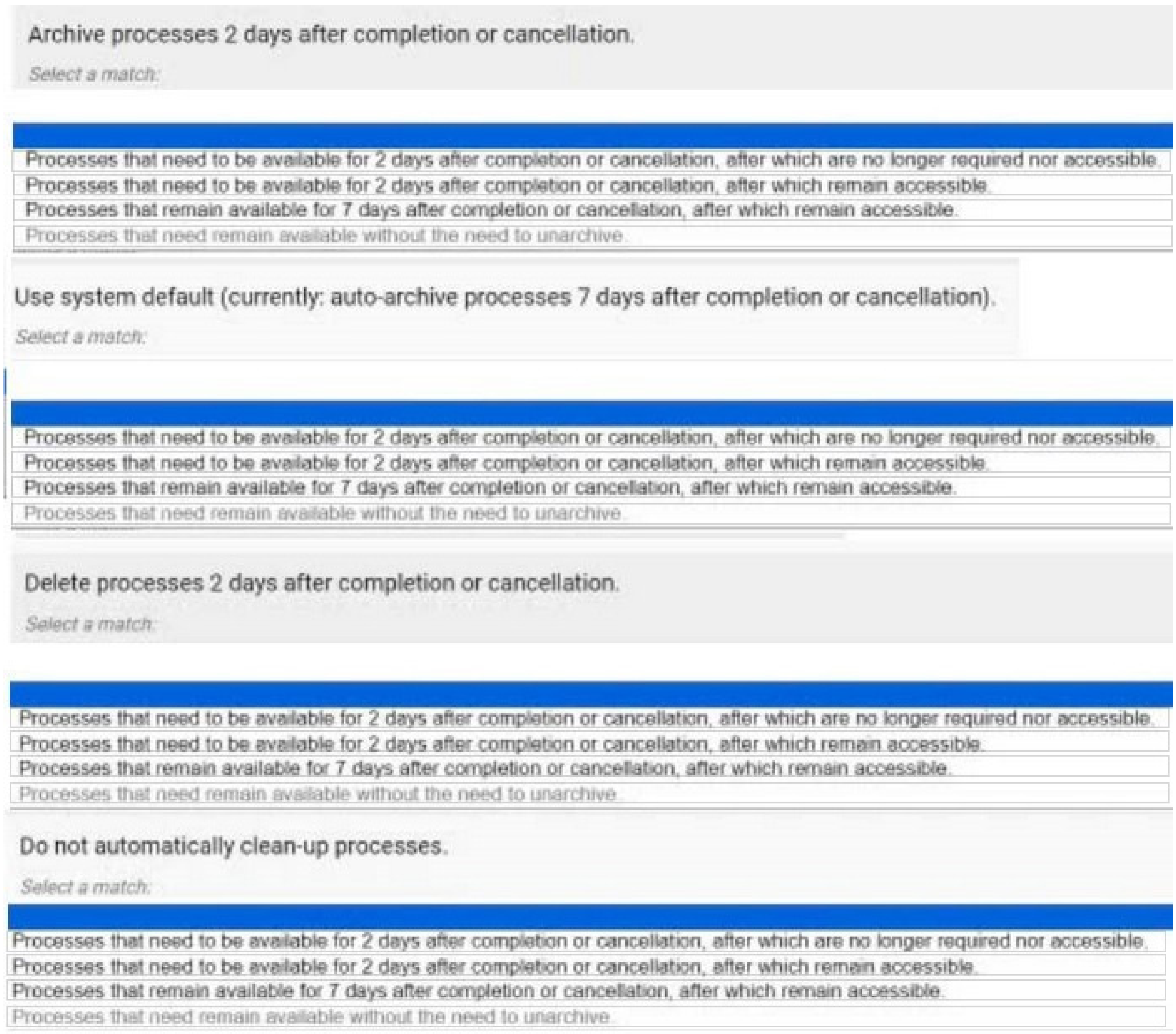
HOTSPOT

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:

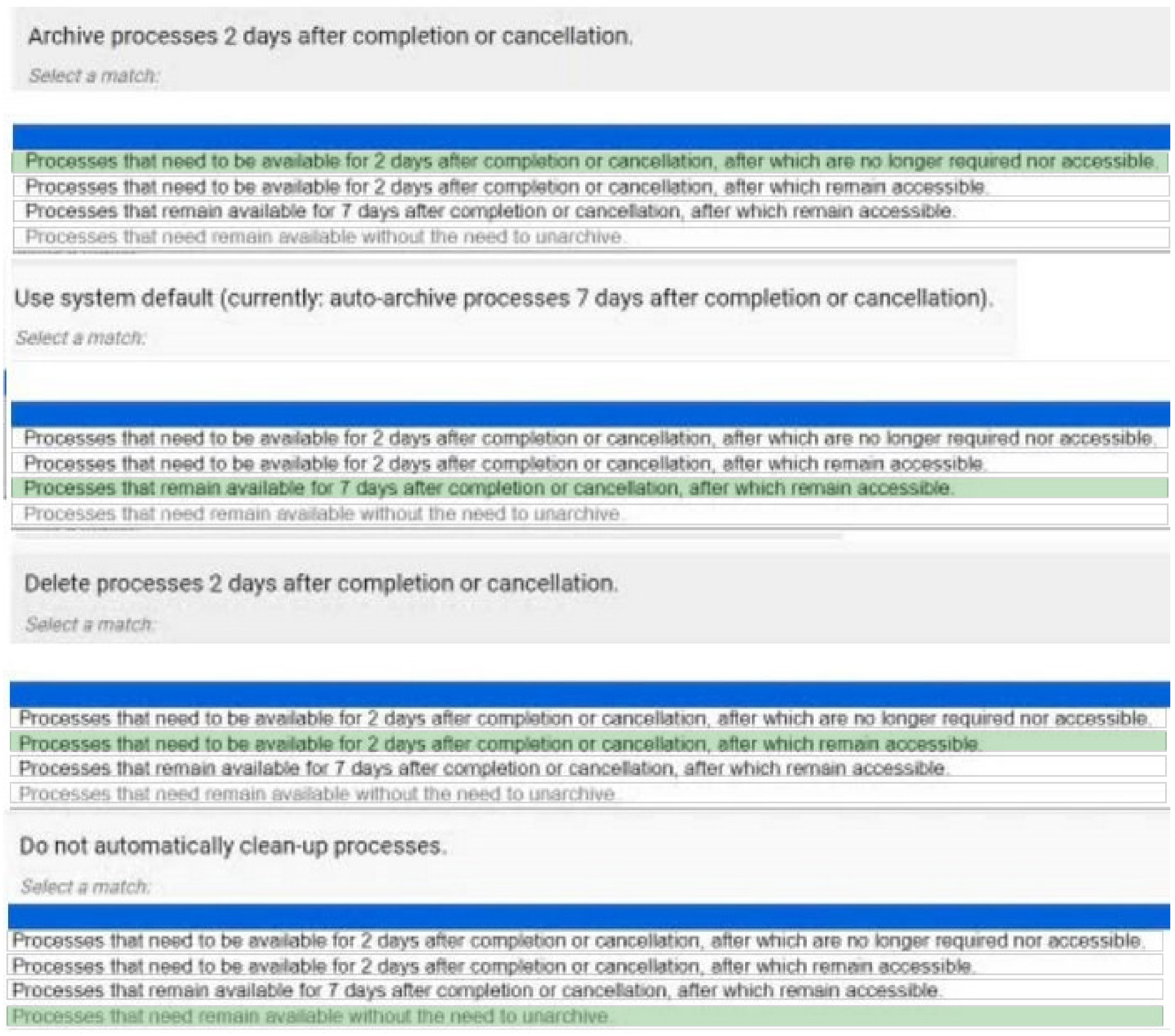
- Processes that need to be available for 2 days after completion or cancellation, after which are no longer required nor accessible.
- Processes that need to be available for 2 days after completion or cancellation, after which remain accessible.
- Processes that remain available for 7 days after completion 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.

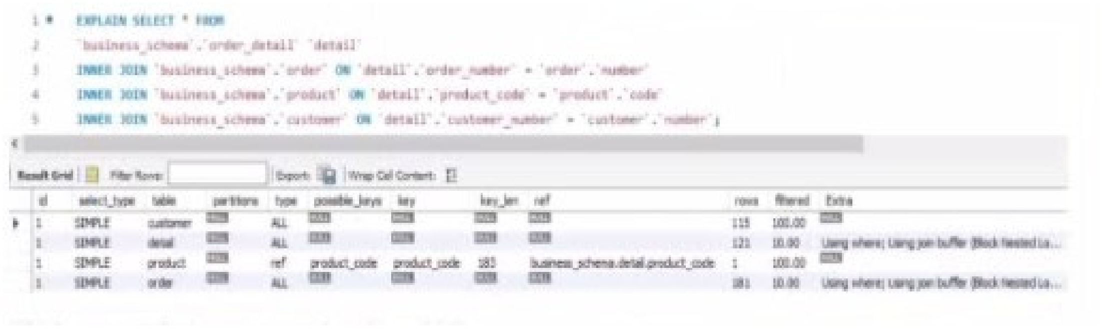
Select a match:

- 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

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`;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

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.