Question 1

Question Type: **MultipleChoice**

In what order are objects created to generate a RESTful Application Programming application?

A.

Database table 1

B.

Service binding Projection view 4

C.

Service definition 3

D.

Data model view 2

Options:

A- D A B C

B- B D C A

C- A D C B

D- C B A B

Answer:

C

Explanation:

The order in which objects are created to generate a RESTful Application Programming application is A, D, C, B. This means that the following steps are followed:

First, a database table is created to store the data for the application. A database table is a CDS DDIC-based view that defines a join or union of database tables. A database table has an SQL view attached and can be accessed by Open SQL or native SQL.

Second, a data model view is created to define a data model based on the database table or other CDS view entities. A data model view is a CDS view entity that can have associations, aggregations, filters, parameters, and annotations. A data model view can also define the behavior definition and implementation for the business object.

Third, a service definition is created to define the service interface for the application. A service definition is a CDS view entity that defines a projection on a data model view or another service definition. A service definition can also define service metadata, such as service name, version, description, and annotations.

Fourth, a service binding is created to define the service binding for the application. A service binding is a CDS view entity that defines a projection on a service definition. A service binding can also define the service protocol, such as OData V2, OData V4, or REST, and the service URL.

Question 2

Question Type: **MultipleChoice**

What are advantages of using a field symbol for internal table row access? Note: There are answers to this question.

Options:

A- The field symbol can be reused for other programs.

B- A MODIFY statement to write changed contents back to the table is not required.

C- The row content is copied to the field symbol instead to a work area

D- Using a field symbol is faster than using a work area.

Answer:

B, D

Explanation:

A field symbol is a pointer that allows direct access to a row of an internal table without copying it to a work area.Using a field symbol for internal table row access has some advantages over using a work area, such as12:

A MODIFY statement to write changed contents back to the table is not required: This is true. When you use a work area, you have to copy the row content from the internal table to the work area, modify it, and then copy it back to the internal table using the MODIFY statement. This can be costly in terms of performance and memory consumption. When you use a field symbol, you can modify the row content directly in the internal table without any copying.Therefore, you do not need the MODIFY statement12.

Using a field symbol is faster than using a work area: This is true. As explained above, using a field symbol avoids the overhead of copying data between the internal table and the work area. This can improve the performance of the loop considerably, especially for large internal tables.According to some benchmarks, using a field symbol can save 25--40% of the runtime compared to using a work area12.

You cannot do any of the following:

The field symbol can be reused for other programs: This is false. A field symbol is a local variable that is only visible within the scope of its declaration. It cannot be reused for other programs unless it is declared globally or passed as a parameter. Moreover, a field symbol must have the same type as the line type of the internal table that it accesses.Therefore, it cannot be used for any internal table with a different line type12.

The row content is copied to the field symbol instead to a work area: This is false. As explained above, using a field symbol does not copy the row content to the field symbol. Instead, the field symbol points to the memory address of the row in the internal table and allows direct access to it.Therefore, there is no copying involved when using a field symbol12.

Question 3

Question Type: **MultipleChoice**

In a RESTful Application Programming application, in which objects do you bind a CDS view to create a value help? Note: There are 3 correct answers to this question.

Options:

A- Data model view

B- Behavior definition

C- Metadata Extension

D- Service Definition

E- Projection View

Answer:

A, C, E

Explanation:

In a RESTful Application Programming (RAP) application, you can bind a CDS view to create a value help in the following objects:

Data model view: A data model view is a CDS view that defines the data structure and the associations of an entity in the RAP application. You can use the annotation @Consumption.valueHelpDefinition to bind a value help provider CDS view to an element of the data model view. The value help provider CDS view must contain the key fields of the value help entity and the fields that are displayed in the value help dialog.The value help annotation specifies the entity name, the element name, and optionally the additional binding conditions for the value help provider1.

Metadata Extension: A metadata extension is a CDS view that extends the metadata of another CDS view without changing its data structure. You can use the annotation @MetadataExtension.extendView to specify the target CDS view that you want to extend. You can then use the same annotation @Consumption.valueHelpDefinition to bind a value help provider CDS view to an element of the target CDS view.The metadata extension allows you to add value help definitions to existing CDS views without modifying them2.

Projection View: A projection view is a CDS view that defines the projection of another CDS view. You can use the annotation @AbapCatalog.sqlViewType: #PROJECTION to specify that the CDS view is a projection view. You can then use the same annotation @Consumption.valueHelpDefinition to bind a value help provider CDS view to an element of the projection view.The projection view allows you to add value help definitions to projected elements of another CDS view3.

You cannot bind a value help provider CDS view to a behavior definition or a service definition, because these objects do not define the data structure or the metadata of an entity in the RAP application.A behavior definition defines the behavior and the validation rules of an entity, such as the create, read, update, and delete (CRUD) operations, the draft handling, the authorization checks, and the side effects4.A service definition defines the service exposure and the service binding of an entity, such as the protocol, the version, the namespace, and the service name5.

Question 4

Question Type: **MultipleChoice**

What RESTful Application Programming object contains only the fields required for a particular app?

Options:

A- Database view

B- Metadata extension

C- Projection View

D- Data model view

Answer:

C

Explanation:

A projection view is a RESTful Application Programming object that contains only the fields required for a particular app. A projection view is a CDS view entity that defines a projection on an existing CDS view entity or CDS DDIC-based view. A projection view exposes a subset of the elements of the projected entity, which are relevant for a specific business service. A projection view can also define aliases, virtual elements, and annotations for the projected elements. A projection view is the top-most layer of a CDS data model and prepares data for a particular use case. A projection view can have different provider contracts depending on the type of service it supports, such as transactional query, analytical query, or transactional interface.

A database view is a CDS DDIC-based view that defines a join or union of database tables. A database view has an SQL view attached and can be accessed by Open SQL or native SQL. A database view can be used as a projected entity for a projection view, but it does not contain only the fields required for a particular app.

A metadata extension is a RESTful Application Programming object that defines additional annotations for a CDS view entity or a projection view. A metadata extension can be used to enhance the metadata of a CDS data model without changing the original definition. A metadata extension does not contain any fields, but only annotations.

A data model view is a CDS view entity that defines a data model based on database tables or other CDS view entities. A data model view can have associations, aggregations, filters, parameters, and annotations. A data model view can be used as a projected entity for a projection view, but it does not contain only the fields required for a particular app.

Question 5

Question Type: MultipleChoice

What are valid statements? Note: There are 2 correct answers to this question.

Options:

A- 'zcxl' is a dictionary structure, and 'paraml' and 'param2' are this structure.

B- 'paraml11 and 'param2' are predefined names.

C- The code creates an exception object and raises an exception.

D- 'previous' expects the reference to a previous exception

Answer:

C, D

Explanation:

The code snippet in the image is an example of using the RAISE EXCEPTION statement to raise a class-based exception and create a corresponding exception object.The code snippet also uses the EXPORTING addition to pass parameters to the instance constructor of the exception class12. Some of the valid statements about the code snippet are:

The code creates an exception object and raises an exception: This is true. The RAISE EXCEPTION statement raises the exception linked to the exception class zcxl and generates a corresponding exception object.The exception object contains the information about the exception, such as the message, the source position, and the previous exception12.

"previous" expects the reference to a previous exception: This is true. The previous parameter is a predefined parameter of the instance constructor of the exception class cx_root, which is the root class of all class-based exceptions. The previous parameter expects the reference to a previous exception object that was caught during exception handling.The previous parameter can be used to chain multiple exceptions and preserve the original cause of the exception12.

You cannot do any of the following:

"zcxl" is a dictionary structure, and "paraml" and "param2" are this structure: This is false. zcxl is not a dictionary structure, but a user-defined exception class that inherits from the predefined exception class cx_static_check. param1 and param2 are not components of this structure, but input parameters of the instance constructor of the exception class zcxl.The input parameters can be used to pass additional information to the exception object, such as the values that caused the exception12.

"paraml" and "param2" are predefined names: This is false. param1 and param2 are not predefined names, but user-defined names that can be chosen arbitrarily. However, they must match the names of the input parameters of the instance constructor of the exception class zcxl.The names of the input parameters can be declared in the interface of the exception class using the RAISING addition12.

Question 6

Question Type: **MultipleChoice**

In ABAP SQL, which of the following retrieves the association field_Airline-Name of a CDS view?

Options:

A- _Airline-Name

B- /_Airline Name

C- @_Airline-Name

D- '_Airline Name

Answer:

C

Explanation:

In ABAP SQL, the syntax to retrieve the association field of a CDS view is to use the @ sign followed by the association name and the field name, separated by a period sign (.). For example, to retrieve the association field _Airline-Name of a CDS view, the syntax is @_Airline.Name.This syntax allows the access to the fields of the target data source of the association without explicitly joining the data sources1. The other options are incorrect because they use the wrong symbols or formats to access the association field.

Question 7

Question Type: **MultipleChoice**

Which of the following ABAP SQL statements are valid? Note: There are 2 correct answers to this question.

Options:

A- SELECT FROM /dmo/connection FIELDS carrid O airpfrom,
MAX(distance) AS dist_max, MIN(distance) AS dist_min GROUP BY carrid, airpfrom INTO TABLE @DATA(It_hits)

B- SELECT FROM /dmo/connection FIELDS \/ O carrid, airpfrom,
MAX(distance) AS dist_max, MIN(distance) AS dist_min INTO TABLE @DATA(It_hits)

C- SELECT FROM /dmo/connection FIELDS V D MAX(distance) AS dist_max
MIN(distance) AS dist_min INTO TABLE @DATA(It_hits).

D- SELECT FROM /dmo/connection FIELDS r---i carrid, airpfrom u GROUP BY carrid, connid
INTO TABLE @DATA(It_hits).

Answer:

A, B

Explanation:

The following are the explanations for each ABAP SQL statement:

A: This statement is valid. It selects the fields carrid, airpfrom, and the aggregate functions MAX(distance) and MIN(distance) from the table /dmo/connection, and groups the results by carrid and airpfrom. The aggregate functions are aliased as dist_max and dist_min. The results are stored in an internal table named It_hits, which is created using the inline declaration operator @DATA.

B: This statement is valid. It is similar to statement A, except that it does not specify the GROUP BY clause. This means that the aggregate functions are applied to the entire table, and the results are stored in an internal table named lt_hits, which is created using the inline declaration operator @DATA.

C: This statement is invalid. It selects the aggregate functions MAX(distance) and MIN(distance) from the table /dmo/connection, but it does not specify any grouping or non-aggregate fields. This is not allowed in ABAP SQL, as the SELECT list must contain at least one non-aggregate field or a GROUP BY clause. The statement will cause a syntax error.

D: This statement is invalid. It selects the fields carrid and airpfrom from the table /dmo/connection, and groups the results by carrid and connid. However, the field connid is not included in the SELECT list, which is not allowed in ABAP SQL, as the GROUP BY clause must contain only fields that are also in the SELECT list. The statement will cause a syntax error.