Oracle

Exam 1z0-531

Oracle Essbase 11 Essentials

Verson: Demo

[Total Questions: 10]

Question No : 1

Identify the true statement about the data cache.

A. Data cache contains compressed data blocks
B. Data cache contains uncompressed data blocks
C. Data cache contains compressed index entries
D. Data cache contains uncompressed index entries

Answer: B

Explanation: The Data Cache is a buffer in memory that holds uncompressed data blocks. Essbase allocates memory to the data cache during data load, calculation, and retrieval operations, as needed.

Note: there are four other caches as well, including the index cache.

Question No : 2

What are the three rules for Shared Members in ASO?

A. A Multiple Hierarchy Enabled dimension can have shared members.
B. When a hierarchy is tagged Multiple Hierarchies Enabled, it must be store.
C. The alternate hierarchy has shared members that refer to nonshared members of previous hierarchies in the outline.
D. The shared members roll up according to a different hierarchy from the nonshared members to which they refer.

Answer: A,C,D

Explanation: Shared member hierarchy is also an alternate hierarchy. All shared member refers to stored members of outline (C). In aggregate storage application only multiple hierarchies can have shared members. (A)

Stored hierarchy has only addition as consolidation operator. You can use the stored hierarchy type where aggregation is the only mathematical requirement. If you have some shared member in hierarchy then use multiple hierarchy.

Question No : 3

You have a reporting requirement to track and report the employee status for employees in your workforce ASO Essbase database. Employee status can change over time. One report requires employees down the rows and employee status across the columns.

What is the best solution to meet the all of the requirements?

A. Separate Employee status dimension
B. Text List
C. Smart List
D. Attribute dimension
E. Varying attribute dimension
F. Alternate hierarchy
G. User defined attribute

Answer: D

Explanation: D: An attribute dimension is a special type of dimension that is associated with a standard dimensions.
Use attribute dimensions to report and aggregate data based on characteristics of standard dimensions. In the Sample.Basic database, for example, the Product dimension is associated with the Ounces attribute dimension. Members of the Ounces attribute dimension categorize products based on their size in ounces.
Essbase does not allocate storage for attribute dimension member. Instead, it dynamically calculates the members when the user requests data associated with them.
Attribute dimensions are always sparse dimensions. And you can associate attribute dimensions only with sparse standard dimensions

Question No : 4

You need to calculate average units sold by the customer dimension within an ASO database. The member formula should calculate correctly regardless of level within the customer dimension. Identify the correct syntax for the member formula.

A. @AVG (SKIPBOTH, “Units_Sold”);
B. Avg(Customer.CurrentMember.Children, [Units_Sold])
C. Avg([Customer].Children, [Units_Sold]);
D. Avg(Customer.CurrentMember.Children, [Units_Sold]);

E. Avg(Customer.Children, [Units.Sold]);

Answer: B

Explanation: A custom rollup technique, custom rollup formulas, lets the cube builder define an MDX formula for each dimension level. Analysis Services uses this formula to determine the value of the dimension level's members. For example, you could use an AVERAGE function rather than a summation to determine all members in one dimension level. If you use the AVERAGE function, the MDX formula for a dimension called Customers would be Avg(Customers.CurrentMember.Children).

Note: The MultiDimensional eXpressions (MDX) language provides a specialized syntax for querying and manipulating the multidimensional data stored in OLAP cubes. While it is possible to translate some of these into traditional SQL, it would frequently require the synthesis of clumsy SQL expressions even for very simple MDX expressions. MDX has been embraced by a wide majority of OLAP vendors and has become the standard for OLAP systems.

Question No : 5

A calculation script is performed on a database for which Create Block on Equation is OFF. The command SET CREATEBLOCKONEQ ON is issued immediately before an equation in the script.

Which statement accurately describe when blocks will be created?

A. Blocks will be created ONLY when the equation assigns non-constant values to members of a sparse dimension
B. Blocks will be created ONLY when the equation assigns constant values to members of a sparse dimension
C. Blocks will be created when the equation assigns either constant or non-constant values to members of a sparse dimension.
D. No blocks will be created.

Answer: C

Explanation:
C: Blocks are always (whether or not CREATEBLOCKONEQ is ON or OFF) created when

a constant value is assigned to a member of a sparse dimension (for which a block does not exist). When SET CREATEBLOCKONEQ ON blocks will also be created when an non-constant value is assigned to a member of a sparse dimension (for which a block does not exist) in a new block.

Note: If this would be a select two alternative question, the alternatives would have to be reworded slightly differently.

Note #1:
The SET CREATEBLOCKONEQ command controls, within a calculation script, whether or not new blocks are created when a calculation formula assigns anything other than a constant to a member of a sparse dimension. SET CREATEBLOCKONEQ overrides the Create Block on Equation setting for the database.

Syntax: SET CREATEBLOCKONEQ ON|OFF;
ON: When a calculation formula assigns a non-constant value to a member of a sparse dimension for which a block does not exist, Analytic Services creates a new block.

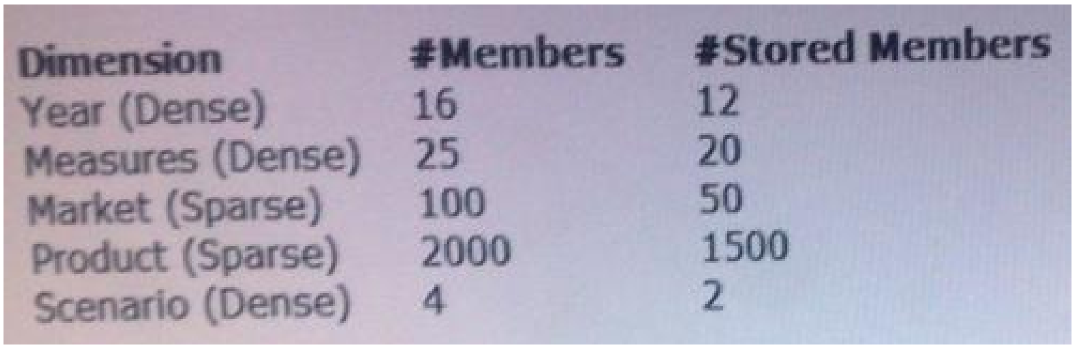
Note #2: The Create Blocks on Equation setting is a database property. The initial value for the Create Blocks on Equation setting is OFF; no new blocks are created when something other than a constant is assigned to a sparse dimension member. You can use Administration Services or MaxL to set the Create Blocks on Equation setting to ON at the database-level. For more information about enabling the Create Blocks on Equation property for a database, see MaxL documentation in the *Technical Reference* or Administration Services online help.
For more specific control, you can use the SET CREATEBLOCKONEQ calculation command within a calculation script to control creation of new blocks at the time the command is encountered in the script. Use of the SET CREATEBLOCKONEQ calculation command has the following characteristics:
* When Analytic Services encounters a SET CREATEBLOCKONEQ command within a calculation script, Analytic Services ignores the database-level setting.

* Where needed in the calculation script, you can use multiple SET CREATEBLOCKONEQ commands to define the Create Blocks on Equation setting value for the calculations that follow each command.
* The value set by the SET CREATEBLOCKONEQ command stays in affect until the next SET CREATEBLOCKONEQ command is processed or the calculation script is finished.

Reference: SET CREATEBLOCKONEQ

Question No : 6

Given the following, what is the declared block size?



Dimension	#Members	#Stored Members
Year (Dense)	16	12
Measures (Dense)	25	20
Market (Sparse)	100	50
Product (Sparse)	2000	1500
Scenario (Dense)	4	2

A. 1920 bytes
B. 480 bytes
C. 3840 bytes
D. 12450 bytes

Answer: C

Explanation: We need to multiple the stored (not the total) members of the dense dimensions (here Year: 12, Measures:20, and Scenario:2) with 8 to calculate the block size.
Block size: 12x20x2x8 = 3840

Note: Data block size is determined by the amount of data in particular combination of dense dimensions. For ex: when you change the dense or sparse configuration of one or more dimensions in the database, the data block size changes. Data block size is 8n bytes, where n is the number of cells that exist (ie. Stored, not total) for that combination of dense dimensions.Note: Optimal range is 8 to 100 kb

Question No : 7

The data block density for a particular BSO database is between 10% and 90%, and data values within the block do not consecutively repeat. Which type of compression would be most appropriate to use?

A. Bitmap
B. RLE
C. ZLIB
D. No compression required

Answer: A

Explanation: Bitmap is good for non-repeating data. It will use Bitmap or IVP (Index Value Pair).

Note: **Bitmap compression**, the default. Essbase stores only non-missing values anduses a bitmapping scheme. A bitmap uses one bit for each cell in the data block, whether the cell value is missing or non-missing. When a data block is not compressed, Essbase uses 8 bytes to store every non-missing cell. In most cases, bitmap compression conserves disk space more efficiently. However, much depends on the configuration of the data.

Question No : 8

You need to tune a block storage option database for calculations on a 32bit Essbase server. Identify the two starting point tuning steps that you might take (assuming no direct I/O).

A. Set Index Cache the size of the index file
B. Set Data Cache to the size of 0.125 * pag file
C. Set Data File Cache to the size of .025 * data file
D. Reset sparse and dense dimensions to achieve a block size greater than 100KB

Answer: A,B

Explanation: Here are the best practices to tune a BSO database:
- ACR (Average clustering Rate) should be as close to 1 as possible, launch a dense restructure to de-fragment the database
- Block Size should be between 10 and 100KB (can be more on 64 bit systems)
- Data cache should be 12.5% of the pag files (B)
- Index cache should be the same size as the ind files (A)
- 1 Database per Application

Question No : 9

Within which two directories can you set the location and file size in ASO?

A. Default
B. Metadata
C. Log
D. Temp
E. Bin
F. Data

Answer: A,D

Explanation: For aggregate storage applications, Tablespace Manager controls data retrieval and storage, using tablespace definitions to manage data storage and work areas on the disk.

Tablespaces help optimize data file and work file storage and retrieval. Tablespaces define location definitions that map data artifacts, such as aggregate views and aggregations, to files. Each application directory contains directories for four tablespaces:
* default
* log
* metadata
* temp

For default and temp you can specify multiple locations and sizes, and you can define tablespace properties:
* Directory path locations
* Maximum disk space to be used at each location
* Maximum file size allowed within each location

Question No : 10

You are creating a MaxL script to log into the database, update a dimension, load data, and run a calculation. Identify the two true statements about creating this MaxL script.

A. A separate MaxL script is required for each step
B. The password must be hardcoded into the script when logging in.
C. IFERROR can be used in MaxL to handle errors after each statement, when triggered will skip to a subsequent statement
D. Variables for objects like server names, application names and database names can be used in a MaxL script to help with maintenance

Answer: C,D

Explanation: C: iferror instructs the MaxL Shell to respond to an error in the previous statement by skipping subsequent statements, up to a certain location in the script that is defined by a label name.
Goto forces the MaxL Shell to branch to a certain location in the script defined by a label name; goto is not dependent on the occurence of an error.
Syntaxiferror *LABELNAME*goto *LABELNAME*define label *LABELNAME*

D: In the MaxL Shell, you can use **variables** as placeholders for any data that is subject to change or that you refer to often; for example, the name of a computer, user names, and passwords. You can use variables in MaxL scripts as well as during interactive use of the shell. Using variables in MaxL scripts eliminates the need to create many customized scripts for each user, database, or host.
Variables can be environment variables (for example, $ARBORPATH, which references the directory Essbase is installed to), positional parameters (for example, $1, $2, etc.), or locally defined shell variables.
All variables must begin with a $ (dollar sign). Locally defined shell variables should be *set* without the dollar sign, but should be *referenced* with the dollar sign. Example:
set A = val_1;
echo $A;
val_1

Incorrect answer:
A MaxL cannot contain several steps.
Example:
login $1 $2;

import database sample.basic dimensions
from data_file 'C:\\data\\dimensions.txt'
using rules_file 'C:\\\\data\\rulesfile.rul'
on error append to 'C:\\\\logs\\dimbuild.log';

iferror 'dimbuildFailed';

```
import database sample.basic data from data_file
"$ARBORPATH\\app\\sample\\basic\\calcdat.txt"
on error abort;

define label 'dimbuildFailed';
exit;
```

B: It is recommend that you encrypt the MaxL scripts that includes user names and password, but it is not required.

Note:
MAXL is an script language that we could use to manipulate essbase, we could use it to

* create or modify essbase applications or database or even outline
* create or modify dimension (e.g. add new member to the dimension)
* importing data into database
* execute calculation scripts.
* ...many more , actually most of the functionality that we use the graphic admin console to do could be done using maxl scripts.

MAXL script is only simple text that we could edit or write using the simple notepad . although admin console do provide an more easy editor for editing MAXL scripts.

Reference: MaxL Shell Syntax Rules and Variables