Q1. Why sort component divide data into 1/3 of max core?

Ans:

Sort max score divide into 3 part

One part is buffer

Another part is data (to store the data)and

3rd part is awake(use memory for algorithm)

Q2. When does Abinito creates work directories where it stores the temp files? Does it create when the sort component uses a particular layout for the first time or it has to be created separately?

Ans: Which ever path is provided in layout option, will be used to crate intermediate files.

Tip 1:

In Abinitio we can read and write the same file in same phase when Access mode is "Append to File"

However this will fail when Access mode is set to "Delete and recreate file".

Tip 2:

Error handling:

make_error: customizes error before sending to error port. You can have any record format at error port.

output_for_error: if error occurs, it can be sent to output port with appropriate mapping.

With the help of these 2 functions you can have same dml for good and erroneous record. Component should be set to Never Abort

Tip 3:

Sometimes you create generic graph which has same transform for multiple input dml or custom function whose input parameter data type is not fixed and you want to check if specific field is present in input or not, then you have some options like

a) record_info on dml file

b) eval

c) recently introduced function **is_present()**

Tip 4:

Logging thorough Component:

You can call **write_to_log** function within transform and send any information to log port.

Whereas **write_to_log_file** can write directly to a file or stdout. Since 2nd parameter is of type string, you can use string_reptesentation function to write field or record of any type.

Example:

write_to_log_file("/tmp/log.txt", string _representation(in));

Q3.I have one xfr. I would like to identify how many graphs are using that xfr?

Ans:

air object uses /Projects/sample/xfr/abc.xfr -top-level

air object uses has been changed to air object referencing-files

For grep u can also use

grep -lR xfrname *

Here, L for list file name and R for to check in each sub directory

Q4. Broadcast versus Replicate component.

Ans:

Broadcast and replicate behavior varies with flow type and default flow type is different for both. It's always difficult to remember it unless you try all scenarios.

On a short note when you connect this component between same depth, default flow type for broadcast is all to all and it multiplies the data.

I would suggest to read help and simultaneously run trials.

Primary motive of broadcast is data parallelism and it duplicates data in each partition flow. As I said it behaves differently based on flow type and you need to run it to get correct understanding.

Q5. How to remove header and trailer from CSV file?

Ans:

Read by new line initially and remove header trailer then redefine.

To remove trailer, use dedup component with option key as null and keep last record. Finally retrieve all data records from dedup port.

Q6. How to do the count(distinct ID) In Abinitio?

Ans:

1) dedup and the count function in rollup

2) use expanded mode in rollup

Yes, dedup on cust id and count on null key.

And if you want to count only unique then “keep unique”. if you only to count at least once per id then “keep first”.

Q7. Kill vs m_kill ?

Ans: Kill is for unix process and m_kill for ab initio graph. kill needs process id, and m_kill needs rec file name which is easy to find out, and may be it would be doing something more in backend, like waiting for AB_TIMEOUT

Q8. What is the difference between .abintiorc and abinitiorc file?

Ans:

abinitiorc file is server file which is common to all server user and only admin can do changes to it.

.abinitiorc is user config file and has higher precedence over system config file abinitiorc.

Both contains lot of configuration variables like ab home, ab air root, host connection etc.

Tip 5:

We can use below command to change the version of an existing object in a tag.

Air tag change-version <tagname> object_name

We can pass -version and specific version number as well.

Tip 5:

We can use below command to change the version of an existing object in a tag.

Air tag change-version <tagname> object_name

We can pass -version and specific version number as well.

```
air tag change-version mytag  /Projects/lesson/dml/JoinCustomers.dml   -version 23
```

Q9. I have 100 records in a input file and I want to send first 20 to the output file. When I run the graph 2nd time it will take 21 st to 40 and so on

Ans1:

I/p file then reformat having two out port.

One port connect to a ROLLUP and then file this will hold the count.

2nd port to target file or table.

In reformat, use the count from that lookup file in select

Invocation_number > (lookup.count) and invocation_number < (lookup.count +20)

Ans2:

Use read multiple file component with read count option as 20 and skip count with sandbox parameter which will gets increased by 20 after every run.

Export global parameter in the end script.

Q10. How to add trailer record at every Third record of a file having 10 records?

Ans:

In output index of reformat use logic if next in sequence % 3 == 0 then output add trailer from in1.

Ans2:

You can use normalize generate length as 2 when next in sequence %3 ==0.

Q11. I have 100 records in input file in output i want 1 million records. What will be best approach?

Ans:

Use normalize with length as 10,000 fixed number.

Graph will be having below components to produce output:

Input file-> normalize -> output file

Tip 6:

Maximum max-core value allowed for system is calculated as below:

It's 2^31 - 1 for 32 bit around 2GB space

Q12. How we can pass jobid while running air sandbox run?

Ans:

U have to pass -AB_JOB_PREFIX for graph and not job id.

Ab_job_prefix is for graph level parameter and job_id is on plan level.

Q13. Can rollup remove duplicates? How?

Q14. What will be the result if you don't provide a key? (sort/dedup/rollup/lookup/join)

Q15. For Rollup, in what case can the output record count be more than the number of unique keys in input file? (show an example)

Q16. Differentiate between gather and fan-in flow.

Q17. How many priorities can be assigned at the max?

Ans: The priority setting's value is an integer, zero or greater, that is **up to 10 digits long**.

Q18. Length prefixed strings?

Q19. What is implicit and explicit reformat?

Q20. What do we receive in reject port of reformat?

Q21. if record is rejected from reformat, how can I find reason for that

Q22. I want to define default values for invalid date values in the date field of file, what should be done in transform of reformat component to achieve this.

Q23. Is it required to use sorted input record to use lookup function?

Q24. What all types of join can be used with lookup component? Which one type is not possible? Why?

Q25. What do you mean by spilling data on disk?

Q26. What is pipeline parallelism and which component breaks it?

Q27. If I use in-memory sort, will it give sorted output or records in order of input or random order?

Q28. When will you use output index against partition by expression.

Q29. Can you join 2 files having keys with different datatypes?

Q30. Which components can be used to reinterpret data?

Q31. What the difference is between redefine format and reformat.

Q32. What are multi stage transforms. Give examples of components which have multi stage transforms.

Q33. Explain run time behavior of Normalize.

Q34. List various functions in the normalize component.

Q35. What is the difference between using length variable and finished function in normalize.

Q36. Which parameter decides the count of How many times a finalize function will run.

Q37. List all the aggregation functions in rollup

Q38. What is the difference between rollup and scan?

Q39. What is difference between is_defined and first_defined.

Q40. I have 6 columns and 5 records in a file , I have to do transpose it , please tell possible ways of doing it.

Tip 7:

Working of First defined function is same as coalesce function of SQL statement.

E.g.

Coalesce(id,00) - if id is NULL then default value 00 will be assigned to it in SQL queries.

first_defined(lookup("lkp",in.id),00) -

If lookup output is null then first defined will assign default value 00 to it.

Tip 8:

Using lookup function you cannot achieve full outer join. Only left or right join can be achieved as we can't pull uncalled records from lookup file.

E.g.

Input file have below 3 records:

ID

1

2

3

Lookup file have below 5 records:

ID	Name
1	A
2.	B
3.	C
4	D
5	E

Suppose now you have used lookup function and then you can't pull all the records from lookup file in the output process.

As there is no ID is present in input file for number 4 & 5 hence their Name will not get pulled.

lookup("input file",in.ID).Name

Hence in Abinitio using lookup function we can't achieve full outer join functionality. To achieve this we need to use Join component.

Q41. I have 6 columns and 5 records in a file. I have to do transpose it , please tell possible ways of doing it.

Q42. How to convert 4-way multifile to 16-way-multilfile without using partition component?

Q43. You have one multi file and while operating it you come to know that one of its partition corrupted. So now how you will correct it? Provide at least three solutions.

Q44. You have a file and you have to extract header , trailer and body records. how to achieve it without rollup.

Q45. How to calculate second highest salary of each department? Provide 3 solutions.

Tip 9:

Information on .abi-unc files:

These are temporary files used by the checkout process. Under normal circumstances you should not see them. However, if the checkout failed or was interrupted, these files can be left behind. Delete them and try checking out again.

Details:

The checkout procedure includes two steps to ensure that a checkout does not leave your sandbox in a half-checked-out state:

1. It checks out the files, but gives them the .abi-unc suffix (for "uncommitted").

2. Once every file is successfully checked out, the .abi-unc files replace the real files. If an error occurs midway through checkout, any .abi-unc files that have been created might be left behind.

If you receive the error message "Cannot write project parameters file: /path/filename.abi-unc" during checkout and you see .abi-unc files, delete them and try checking out again. Uncommitted files are usually left behind because a checkout process was aborted.

Tip 10:

The job recovery file(.rec) working:

Co>Op monitors & records the state of jobs so that if a job fails, it can be restarted. This state info is stored in files associated with the job and enables the Co>Op to roll back the system to its initial state, or to its state as of the most recent completed checkpoint. Generally, if the application encounters a failure, all hosts and their respective files will be rolled back to their initial state or their state as of the most recent completed checkpoint; you recover the job simply by rerunning it.

Details:

An AI job is considered completed when the mp run command returns. This means that all the processes associated with the job — excluding commands u might have added in the script end — have completed. These include the process on the host system that executes the script, and all processes the job has started on remote computers. If any of these processes terminate abnormally, Co>Op terminates the entire job and cleans up as much as possible.

When an AI job runs, the Co>Op creates a file in the working directory on the host system with the name jobname.rec. This file contains a set of pointers to the log files on the host and on every computer associated with the job. The log files enable the Co>Op to roll back the system to its initial state or to its state as of the most recent checkpoint. If the job completes successfully, the recovery files are removed (they are also removed when a single-phase graph is rolled back).

If the app encounters a software failure (Ex, one of the processes signals an error or the operator aborts the app), all hosts & their respective files are rolled back to their initial state, as if the app had not run at all. The files return to the state they were in at the start, all temporary files & storage are deleted, & all processes r terminated. If the program contains checkpoint commands, the state restored is that of the most recent completed checkpoint.

When a job has been rolled back, u recover it simply by rerunning it. Of course, the cause of the original failure might repeat itself when you rerun the failed job. You will have to determine the cause of the failure by investigation or by debugging.

When a checkpointed application is rerun, the Co>Op performs a fast-forward replay of the successful phases. During this replay, no programs run & no data flows; that is, the phases r not actually repeated (although the monitoring system cannot detect the difference bet'n the replay & an actual execution). When the replayed phases are completed, the Co>Op runs the failed phase again.

Note that it might not always be possible for the Co>Op to restore the system to an earlier state. Ex, a failure could occur because a host or its native OS crashed. In this case, it is not possible to cleanly shut down flow or file operations, nor to roll back file operations performed in the current phase. In fact, it is likely that intermediate or temporary files will be left around.

To complete the cleanup and get the job running again, you must perform a manual rollback. You do this with the command m_rollback.

The syntax is:

m_rollback [-d] [-i] [-h] recoveryfile

Running m_rollback recoveryfile rolls the job back to its initial state or the last completed checkpoint. Using the -d option deletes the partially run job & the recovery file.

Q46. How to find out who is using your generic graph?

Q47. What is better join with database or unload and join in abinitio. Explain why?

Q48. Where you maintain external servers/applications connectivity information?

Q49. What all the things you will take into consideration when you introduce a phase break. (whats the reason lead you to use phase break?)

Q50. What type of parallelism you will break by using phase breaks in graph?

Q51. For any duplicate record entries only single should be present in output.

E.g.

Source_city	Destination_city	kms
Mumbai	Pune	200
Pune	Hyderabad	550
Hyderabad	Pune	550
Pune	Mumbai.	200

Output should be:

Mumbai Pune 200

Pune Hyderabad 550

Q52. How to pass the records to output only if input id is prime number.

Q53. We have 4 files. We have to unload the data from those 4 files. If for any file 1 record is rejected then that file need to reject? How you can achieve this?

Q54. In file we have data like below:

1,murali,3,4,5

In the output file We need the data like this

1,murali,3

1,murali,4

1,murali,5

How do you achieve this?

Q55. In an org, we have 2 files say file1, file2.

File1 have employee details who are currently working in that org & who are left from that org.

File2 have working employees.

We need to send the data In output file file1 working employees & in output file file2 we need to send the data who are left from the org.

How can u achieve this?

Q56. We have file file1. In that we have 10 records. How can we send all these records to reject port by using Filter By Expression component?

Q57. I have a graph like the below (for example)

input file --> join ---->replicate ---->sort --->output

in the above, the layout of input file and join are set to multifile where as sort and output file are set to serial layout.

So, the above graph ran file. My question is, does replicate behaves as a gathers in this case, if yes please explain why? Because I was expecting a depth error here.

Q58. I have an ebcdic file from mainframe system. Its dml is something like

record

integer(4) id,

void(120) bill_dtl

end

At position 80-85 and 100-104 has some information (it is actauly amount and product in the bill) which I need to extract, how to do it.

Q59. Suppose Sachin and Sehwag playing the match. Sachin playing the first ball Then how to calculate total runs scored by Sachin in single over? Consider extra balls like wide and no-ball as well along with 1,3,5 runs.

Q60. I have customer details as below,

cust_card_no item amount

10001 pen 10

10001 copy 20

10001 pen 10

10001 copy 20

10001 copy 20

now my question is to find the distinct count of items per customer and along with total amount spent by a customer?

Q61. Suppose I have a scenario like below:

Input:

======

Id Col1 Col2

a 100 Null

a Null 200

b 300 Null

b Null 400

And I need output as:

Output:

=======

Id Col1 Col2

a 100 200

b 300 400

How to get only Not NUll Values?

Q62. How to achieve the below scenario in abinitio?

Input file has below contents: Ball Run

1 1

2 1

3 1

4 1

5 1

6 1

1 0 ,2 1 ,3 1 ,4 1 ,5 1 ,6 1

1 1 ,2 1 ,3 1 ,4 1 ,5 1 ,6 0

Required Output :

Over Runs

1 6

2 5

3 5

Q63. I have a record like below: avi,mahi,virat,mahi,avi,virat,mahi,virat,avi

Output should be: avi1 mahi1 virat1 mahi2 avi2 virat3 mahi3 virat4 avi3

Please suggest how can I achieve this.

Tip 11:

A component's layout specifies where the component runs (on what computer & in what directory) and the number of ways parallel the component runs.

The layout default lacks a directory path, so when this layout is used, the component runs in the directory specified by AB_WORK_DIR.

During execution, components might need to write temporary data to disk. In most cases these temporary files r maintained in .WORK subdirectory of a component's layout.

For database components with a Database:default layout, the directory specified by AB_DATA_DIR provides disk storage for temporary files.

Q64. What is a broken lock ? How to fix it?

Ans:

If u want to edit a file, u must obtain a lock on that file. This prevents multiple users from modifying the same file at same time. However, if another user wants to edit ur locked file, he/she must 1st break ur lock. The act of breaking lock creates a broken lock.

A broken lock is effectively a msg to the original lock owner that another user has broken the lock.

When u open the file with broken lock, the GDE displays a msg informing u that ur lock has been broken. At that point, u may need to reconcile your changes with those made by other users. Once you click OK to the message, the GDE "resets" (removes) the broken lock.

At the command line use air lock break to break locks and air lock reset to remove broken locks.

U cannot lock a file if u own a broken lock on that file. You or the EME administrator must first reset the lock. An EME administrator can reset a user username's lock on file object-path by running:

air lock reset -user username -object object-path

Q65. What is the difference between a phase and a checkpoint?

Ans:

The essential differences bet'n a phase & a checkpoint are their purpose & how the temporary files containing the data landed to disk are handled:

- Phases are used to break up a graph into blocks for performance tuning.

- Checkpoints are used for the purpose of recovery.

Details: Following descriptions clarify the differences bet'n phases & checkpoints:

- Phase: The primary purpose of phasing is performance tuning by managing resources. Phasing limits the number of simultaneous processes by breaking up a graph into different pieces, only one of which is running at any given time. One common use of phasing is to avoid deadlocks. The temporary files created by phasing are deleted at the end of the phase, regardless of whether the run was successful.

- Checkpoint: The main aim of checkpoints is to provide the means to restart a failed graph from some intermediate state. When a graph with checkpoints fails, the temporary files from the last successful checkpoint are retained so you can restart the graph from this point in the event of a failure. Only as each new checkpoint is completed successfully are the temporary files corresponding to the previous checkpoint deleted.

Q66. What is difference between fan-out and partition by round robin?

Ans:

1. Basically working of both components is similar but data skew can be present if you are using fan out as compared to PRR component.

Input: 1 2 3 4 5 6 7 8 9 0

Suppose u r working on 3 way partition then

Output in fan out can be:

Partition1:14580

Partition2:2679

Partition3:3

As data is flowed to all partition equally it resulted in data skew where as this will not happen through PRR.

2. Input layout and output layout of component is same then we can't use fan out where as PRR can be used.

Serial layout -> fanout -> serial layout component (not possible)

Serial layout -> PRR -> serial layout component(possible)

3. Fan out will work faster as compared to PRR as data is flowing on the fly via data stream.

Tip 12:

There are 3 different ways to checkin a file marked as Never Checkin:

at the command line,

from the EME MC and

from the GDE Checkin Wizard.

Details: Once a file is marked as Never Checkin in GDE Checkin Wizard, it is not available for checkin through wizard. The file is assigned the MIME type of ignore in the project files list.

To check in a file marked as Never Checkin:

- At the command line, run air project import on the file (say, dml/mydml.dml) with the -force and -files options (but not the -auto-add option):

```
air project import \
  /Projects/lesson -basedir /disk1/data/sand/lesson \
  -force \
  -files dml/mydml.dml
```

Alternatively, run air project set-type to change the MIME type from ignore to one appropriate for the file. Then check in only the file (not its project).

- From the EME MC, use the Project Files List dialog. Double-click Ignored and change the value to No. Then choose another MIME type if the current one is not appropriate. Finally, check in only the file (not its project).

- From the GDE Checkin Wizard, check in only the file, selecting the Force overwrite check box on the Advanced Options dialog.

Tip 13: Committing intermediate updates (api mode only) with UPDATE TABLE and OUTPUT TABLE

You can determine the rate at which records are committed by specifying values for the commitNumber and commitTable parameters. If no commitNumber is specified to UPDATE TABLE, no updates are committed to the database until all data is processed.

NOTE:

If you are running UPDATE TABLE in a parallel layout, and if you have not specified a commitNumber, the effect of not specifying the parameter will occur on a per-partition basis (commitNumber is always interpreted on a per-partition basis). For a given process, it will be true that no updates are committed until all its data is processed, which will happen at some time before the end of the graph phase in which the process is running. However, it could happen that one of the parallel update processes will finish processing its data, and commit its updates, before the other processes have finished processing their data.

Otherwise, the input data is committed whenever the number of rows processed since records were last committed equals the number specified for the commitNumber parameter.

The table specified by the commitTable parameter keeps track of the number of updates for the current job and ensures that rows are not updated again after they have once been processed and committed.

CAUTION!

Records cannot be rolled back once they have been committed.

In general, only one commit table is needed. It can be shared by multiple jobs and users, and it only needs to be created once. In many environments a single commit table can be sufficient for all users.

NOTE:

For databases that support page level locking, the commit table must use row level locking. For example, for DB2 for z/OS, the tablespace that the commit table is created in must have row level locking specified.

Creating the commit table

You create a commit table by running the following command:

```
m_db create_commit_table config_file

   -table commit_table_name

   -index index_name

   [ -drop ] [ -print_ddl ]
```

Q71. How do you know that, a file that is a multifile or a serial file?

Ans:

By using m_expand –n <file-name>

If the answer is 1 then it is serial file.

If the answer is >1 then it is multi file.

Q75. If we give NULL key in scan component what will be the output and in dedup with keep parameter as unique?

Ans:

In case of {} key in scan it will give all the records in output.

In case of dedup if you keep it as first it will give first record, if you keep it as last it will give last record and in case of unique only it will give no records.

Q79. What will happen if we have multi file lookup and we are calling that in serial layout component?

Q80. What will happen if we are using lookup function for multi lookup file?

Tip 17:

Use m_env –v to know which abinitio version you are using.

Example:

```
m_env -v
```

Output:

```
ab initio version 2.15.8.0
```

Here,

2.15 is release version

and 8 is update version

and 0 denotes patch level

Q81. Input file is multifile and then you have used dedup sorted with null key. How many records will be in output with keep first, last and unique only?

Q82. Input file is multifile and then you have used rollup with null key. How many records will be in output?

Q83. In input file there are 100 million records and you want to create 100 sub files with 1 million in each file. How you will achieve this?

Q84. In graph you have two files one with 100 gb data and another with 100 mb data. Which component you will use to match the file data?

Q85. In graph you have two files one with 100 gb data and another with 100 mb data you using join component with in memory to joining file data. Which file you choose as driving input and why?

Tip 18:

Use below logic generate the sequence number for each record of multi file:

(Next_in_sequence-1)*number_of_partitions+this_partition()

Q86. How do you find the size of MFS file?

Ans: m_ls -ltr

Q87. Difference between PBRR and Interleave?

Q88. Explain Metaprogramming with example?

Q89. What is the difference between versioned and nonversioned objects?

Q90. What will happen if we use the reformat component with NEVER ABORT, Force_error, Force_abort?

Tip 20:

To convert string into character array or in character vector.

For eg.

Input string:

Kishor

Output:

Vector [K,i,s,h,o,r]

Solution:

string_split_no_empty(string_replace("kishor","","-"),"-")

Q92. What is AB_LOCAL expression where do you use it in ab-initio?

Answer:

ablocal_expr is a parameter of input table component of Ab Initio. ABLOCAL() is replaced by the contents of ablocal_expr. Which we can make use in parallel unloads.

There are two forms of AB_LOCAL() construct, one with no arguments and one with single argument as a table name(driving table). The use of AB_LOCAL() construct is in Some complex SQL statements contain grammar that is not recognized by the Ab Initio parser when unloading in parallel.

You can use the ABLOCAL() construct in this case to prevent the Input Table component from parsing the SQL (it will get passed through to the database). It also specifies which table to use for the parallel clause.

Q93. How to Create Surrogate Key using Ab Initio?

Answer:

There are many ways to create surrogate key but it depends on your business logic. Here you can try belows ways.

1. next_in_sequence() function in your transform.

2. Assign key values component

3. Write a stored proc to this and call this stor proc wherever u need.

4. If you writing data into multi file then you have to use below formula to write numbers sequentially:

(next_in_sequence()-1)*no_of_partition()+this_partition()

Q94. How to add header after every fifth record?

Q95. How to read a file having a header after 10 records?

Q96. How to develop CDC logic with only single join component?

Q98. What is the difference between $ and ${} substitution.

Answer:

${} substitution is similar to $ substitution except that the parameter must be preceded by curly brackets. If we talk about these in parameter definitions then - 1. If the interpretation is $ substitution then we can give the value as both $ substitution and ${} substitution. e.g. Parameter can be of name $AI_SERIAL or ${AI_SERIAL} 2. If the interpretation is ${} substitution then we can only give the value as ${} substitution parameter. e.g. Parameter can only be of name ${AI_SERIAL}

Tip 21:

If you want to create protected tag. In other words, no one should be able add, edit or delete tag then you can use air tag freeze command.

Even owner of the tag can’t unfreeze the tag so be careful while using this option.

Syntax:

air tag freeze <tagname>

Tip 22:

To read below date time string:

2019-08-30T12:32:52.006-05:00

We can use date time dml:

datetime("YYYY-MM-DDTHH24:MI:SS.NNN-ZO:NE")

Q100. Without opening a graph how should we know how many input files are using in that graph?

Ans:

Suppose you have graph with name ABC.mp. So to find the number of input datasets(IP files/IP tables..etc) using by the ABC.mp you can do the following:

```
 air project get-required-file /Projectname/mp/ABC.mp > ListofInputfiles.dat
```

/* This wil save list of input files in file ListofInputfiles.dat */

```
 cat ListofInputfiles.dat | wc -l
```

/* It will give the count of input datasets required by the graph */

Tip 23:

When to use sort within groups:

Use Sort within Groups only if records within a group are processed before being sorted on a minor key.

i.e. if processing "A" needs records to be sorted on field {"1"} and later in the flow processing "B" needs records to be sorted on field {"1", "2"}.

In this case before processing "2" and after processing "1" use sort within groups with major-key as {"1"} and minor key as {"2"}.

If records are not grouped to start with, use Sort with a multi-part key. There is no benefit to using Sort within Groups directly after using a Sort component.

Q101. Input file

A

B1

1a

1b

C

B2

2a

2b

Output:

B1

1a

1b

B2

2a

2b

Print 2 next records if found B in input file without using vector.

Q102. Input record contains value as

Atul

Output should be

A

At

Atu

Atul

Q103. Input record contains value as

Kishor

Output should be

K

Ki

Kis

Kish

Kisho

Kishor

Kisho

Kish

Kis

Ki

K

Q104. Input contains more than 1 million records. But you need print last 200 records of it. I don't want to read full file.

Q105. Explain difference between the "truncate" and "delete" commands?

Ans1:

Truncate: It is a DDL command, used to delete table/clusters. Since it is a DDL command hence it is auto commit and Rollback can't be performed. It is faster than delete.

Delete: It is DML command, generally used to delete a record, clusters or tables. Rollback command can be performed , in order to retrieve the earlier deleted things. To make deleted things permanently, "commit" command should be used.

Ans2:

Truncate belongs to DDL command where as DELETE belongs to DML command. Rollback can't be performed incase of Truncate stmt where as Rollback can be performed in Delete stmt. WHERE clause cannot be used in Truncate where as WHERE clause can be used in DELETE stmt.

Ans3:

When u use truncate all the space allotted for the records along with records gets deleted & if u use delete only the records are deleted & the space allocated for the records remains unaltered.

Q106. Explain the differences between api & utility mode?

Ans1:

When u load the data into a table in Utility mode all the constraints are disabled & then data is loaded which leads to faster access.

During API Mode constraints will be enabled so that the access will be slow.

Ans2:

API & UTILITY are two possible interfaces to connect to the db's to perform certain user specific tasks. These interfaces allow the user to access or use certain functions (provided by the db vendor) to perform operation on db's. The functionality of each of these interfaces depends on db's.

API has more flexibility but often considered as a slower process as compared to UTILITY mode. Well the trade off is their performance and usage.

Ans3:

API mode is more diagnostic purpose for retrieving data from db means like selecting particular fields like retrieving the data relatively according to the constraints. Whenever we select API we will get the option of ablocal expression (ab_exp) every time where ever record is passing it will look for the query we write in that ablocal expression block then it will send. That is why it's slow. Where as utility is like retrieving the data fields without any constraints or any conditions that is why it is fast.

Ans4:

In API mode data processing(load/update/insert/delete) is slow however other process can access the database tables during the update.

Compared to above Utility mode processing(load/update/insert/delete) goes faster as it handles data records in large chunks however during that time no other process can access db table i.e. the process running in Utility mode locks the table/owns exclusive ownership of that db instance.

In cross functional & largely distributed organizations API mode is recommended considering the performance aspect over Utility mode.

For one time loads/initialization of huge volume data in tables Utility mode can be used.

Q107. Explain what is .air-lock file?

Q108. What is the difference between .sandbox.pset and .project.pset?

Q109. What is need of .project-start.ksh and .project-end.ksh?

Q110. What are the various types of sandboxes?

Q111. What is ICFF? What are types of it?

Q112. Can blocked compressed ICFF be used with multiple/different keys?

Q113. Explain lookup template?

Q114. Explain working of lookup load function?

Q115. How we can make N number (assume 150) copies of a file in GDE without using replicate?

Q116. In which scenario, .rec files will not get created even if graph fails? How to use in Abinitio graph?

Q117. If a DML is changed, how to find out the impacted objects using a command?

Q118. How to find corrupted record in multifile?

Q119. How to recover a control file that is removed using rm ?

Q120. How to find the schema either Star or Snowflake Schema in our project?

Tip 24:

If you have mistakenly deleted any object from EME but now you want that object again. Then you use air object rollback command as below:

air object rollback <rpath> version_number <eme-server-version-of-graph>

Please check help for more details.

Tip 25:

Difference between max core and max memory.

Max core : component max-core parameter determines the maximum amount of memory the component will consume per partition before it spills to disk.

Max memory : Enables the collection of memory usage data for components.

Q121. Can we merge two tag into single tag?

Q122. What is use of -exact option in tag command?

Q123. How to check in multiple objects using import command?

Q124. How to export single object using export command?

Q125. How to check content of .sav file?

Q126. How to check differences between two versions of same object using air command?

Q127. How to change comment of checked in object?

Q128. Explain use of update-to-current option of air tag duplicate command.

Q129. How to remove secondary objects of already existed tag?