ECRIC – news from Tom Bacon about Monday's lecture

I won't be at the lecture on Monday due to the work swamp.

The plan is still to try and get into the data centre in two weeks time and do the next migration, go live the following week.

I edited down my long list of notes to ***a few short technical points*** and will email them in case any are worth mentioning.

ECRIC – notes from Tom Bacon for Monday's lecture

1) Approach to data migration.

Manual script based for import with no ETL tools used.

Extract, transform and load

A schema per migration separate from the main destination schema with the following:

– A load table per file to hold the data received and uploaded via Oracle SQL*Loader. Data types matching main schema tables where appropriate, e.g. some fields concatenated into destination *rawsource* fields. Text files exported from source system with unit and record separator characters to handle multi-line records. Tighter column typing will pick up field issues at the initial file load stage.

– Primary key mappings table to link load table primary key values to sequence generated primary key values that will be used in the main tables. Populate in bulk in advance to allocate a batch of main table primary key values to the migration.

– Views to map load tables to main schema lookup tables and join through the primary key mapping table to include main table primary IDs. '<*migration name*>_' columns for load table columns and remaining columns corresponding to main schema. Used to drive inserts into main tables. Multiple views on some load tables and union views to wrap up sub-classes for a single destination table insert.

Insert DML run to populate main tables from views. Defensive on primary IDs to prevent attempted duplicate inserts and limited by primary ID range where multiple migration batches are run and main table records from earlier batches may have been deleted.

Extra scripts to update main schema lookup tables from source databases where required.

2) The DML auditing approach needed to be speeded up for the bulk loading of data during migration.

3) Using Oracle Transparent Data Encryption to encrypt tablespaces rather than disk encryption.

4) Using Oracle Data Guard for replication (**redo** log shipping).

5) Need to watch out for character set translation issues when moving and uploading files.

6) Foreign keys should be indexed on child table columns unless the parent table records are not going to be updated or deleted, e.g. a lookup table where records are only ever inserted. Otherwise full scanning the child tables may make parent table updates and deletes very slow.

7) A correlated sub-query based update may not be the best approach. Running several updates to specific values each driven from a single sub-query may be suitable when setting a small number of values such as flag fields and may be much faster with large data volumes.

8) ***Ruby on Rails*** Active Record not using bind variables in SQL so limiting statement reuse. Can be an issue with batch processing if there are multiple sessions. Different Oracle settings may help.

9) A CLOB field was moved into a child table to reduce unnecessary CLOB processing by Active Record.

Character Large OBject

10) Don't forget about *nulls* and *three-valued logic*. If SQL query or DML execute counts aren't what you expect that might be the reason.

11) On Oracle 11g the default profile has a password life time of 180 days so user passwords will expire after this time unless the profile is updated. The default password lock time of 1 day is also better set to unlimited so locked out accounts have to be manually unlocked.

12) Deadlocks seen during de-duplication due to use of a view by ***Rails*** which included a few lookup table values. This led to (subexclusive) lookup table row locking in addition to the (exclusive) main table locking so that locking conflicts on the lookup tables were more likely.

13) It is possible to get invalid dates in Oracle date *datatype* fields from binary inserts (e.g. from Access linked-tables) or **to_date** function bugs which then cause errors when queried.

14) Schema integration complexity is coming from the changes in ICD classifications over time.

15) Don't forget the *rownum* pseudocolumn can be used in Oracle update DML to assign unique values to each row,

e.g. **update tab1 set col1 = rownum ;**

16) The size of the DML audit table needs watching. The rest of the main schema will grow during the migrations but the audit table is growing much more quickly. Archiving off and/or partitioning may not be suitable. Useful compression requires the 11g Advanced Compression licensed option. The table is not required in the analysis snapshots.

17) Analysis snapshots will be taken onto separate server(s), e.g. one daily, a few monthly and one yearly. Analysis users need scratch areas in their own schemas, and a common scratch area with quotas on suitable tablespaces. The approach is being discussed.