CALIBRATION DATABASE VERSION 1.0

SCHEMA

The schema is made of 10 tables, 4 materialized views, 9 triggers, 6 sequences, 2 procedures

TABLES:
- SITES: holds the initials of calibration sites names;
- MDT_HEAD: is the table of the calibrations, any entry is equivalent to 1 day of data taking; it references SITES table and is referenced from tube and rt table;
- MDT_TUBE: holds constants of tubes calibrations and references MDT_HEAD table (the primary key is given by (HEAD_ID, TUBE_ID, SITE_NAME));
- MDT_RT: holds information on the rt relation, but rt and tr are stored in (at least) one of the following tables, it references MDT_HEAD table;
- MDT_RT_CHEBY: holds rt, tr and the resolution parameterized as a sum of Chebytchev polynomials, it references MDT_RT table;
- MDT_RT_MAP_R, MDT_RT_MAP_S, MDT_RT_MAP_T: contain the rt, tr and the resolution parameterized as a map of points, they references MDT_RT table;
- BEFORE_VALIDATION_MDT_TUBE: this table is used to refresh materialized views, related to tubes calibration, at the end of insert in MDT_TUBE (i.e. before validation of constants in MDT_TUBE);
- BEFORE_VALIDATION_MDT_RT: this table is used to refresh materialized views, related to rt calibration, at the end of insert in MDT_RT (i.e. before validation of constants in MDT_RT);

MATERIALIZED VIEWS:
- VIEW_22_FILE_LAST10RUN: selects the first 22 fields of MDT_TUBE table and the root file from the matching MDT_HEAD table for the last 10 uprun into the MDT_TUBE table;
- VIEW_ALL_FILE_LASTINSERT: selects all fields of MDT_TUBE table and the root file from the matching MDT_HEAD table for the last insert time into the MDT_TUBE table;
- VIEW_ALL_FILE_LASTRUN: selects all fields of MDT_TUBE table and the root file from the matching MDT_HEAD table for the last uprun into the MDT_TUBE table;
- VIEW_ALLTUB_ROOTFILE_LAST10CAL: selects all fields of mdt_tube table and the root file from the matching mdt_head table for the last 10 calibrations.

TRIGGERS:
- MDT_HEAD_TRIGGER: is a before insert trigger, it inserts the next value of MDT_HEAD_SEQUENCE in MDT_HEAD.HEAD_ID and the insert time (DATE HH:MM:SS) in MDT_HEAD.INSERT_TIME;
- MDT_TUBE_TRIGGER: is a before insert trigger, it inserts the insert time (DATE HH:MM:SS) in MDT_TUBE.INSERT_TIME;
- MDT_RT_TRIGGER: is a before insert trigger, it inserts the next value of MDT_RT_SEQUENCE in MDT_RT.MDT_RT_ID and the insert time (DATE HH:MM:SS) in MDT_RT.INSERT_TIME;
- MDT_RT_CHEBY_TRIGGER: is a before insert trigger, it inserts the next value of MDT_RT_CHEBY_SEQUENCE in MDT_RT_CHEBY.MDT_RT_CHEBY_ID and the insert time (DATE HH:MM:SS) in MDT_RT_CHEBY.INSERT_TIME;
- MDT_RT_MAP_R_TRIGGER: is a before insert trigger, it inserts the next value of MDT_RT_MAP_R_SEQUENCE in MDT_RT_MAP_R.MDT_RT_MAP_R_ID and the insert time (DATE HH:MM:SS) in MDT_RT_MAP_R.INSERT_TIME;

- MDT_RT_MAP_S_TRIGGER: is a before insert trigger, it inserts the next value of MDT_RT_MAP_S_SEQUENCE in MDT_RT_MAP_S.MDT_RT_MAP_S_ID and the insert time (DATE HH:MM:SS) in MDT_RT_MAP_S.INSERT_TIME;
- MDT_RT_MDP_T_TRIGGER: is a before insert trigger, it inserts the next value of MDT_RT_MAP_T_SEQUENCE in MDT_RT_MAP_T.MDT_RT_MAP_T_ID and the insert time (DATE HH:MM:SS) in MDT_RT_MAP_T.INSERT_TIME;
- TUBE_MATVIEWS_REFRESH_TRIGGER: it is a after insert triggers that after an insert in BEFORE_VALIDATION_MDT_TUBE table will refresh materialized views related to TUBE calibration (i.e.: VIEW_22_FILE_LAST10RUN, VIEW_ALL_FILE_LASTINSERT, VIEW_ALL_FILE_LASTRUN and VIEW_ALLTUB_ROOTFILE_LAST10CAL);
- RT_MATVIEWS_REFRESH_TRIGGER: it is a after insert trigger that, after an insert in BEFORE_VALIDATION_MDT_RT table, refreshes all materialized views related to rt calibration (those views have not been written yet).

SEQUENCES:
- MDT_HEAD_SEQUENCE: it is used to generate unique values for MDT_HEAD.HEAD_ID;
- MDT_RT_SEQUENCE: it is used to generate unique values for MDT_RT.MDT_RT_ID;
- MDT_RT_CHEBY_SEQUENCE: it is used to generate unique values for MDT_RT_CHEBY.MDT_RT_CHEBY_ID;
- MDT_RT_MAP_R_SEQUENCE: it is used to generate unique values for MDT_RT_MAP_R.MDT_RT_MAP_R_ID;
- MDT_RT_MAP_S_SEQUENCE: it is used to generate unique values for MDT_RT_MAP_S.MDT_RT_MAP_S_ID;
- MDT_RT_MAP_T_SEQUENCE: it is used to generate unique values for MDT_RT_MAP_T.MDT_RT_MAP_T_ID;

PROCEDURES:
- BEFORE_VALID_MDT_TUBE_PROC: inserts a row in the BEFORE_VALIDATION_MDT_TUBE table;
- BEFORE_VALID_MDT_RT_PROC: inserts a row in the BEFORE_VALIDATION_MDT_RT table;

PROCESSES AND DATA INSERT

- First a row is inserted into MDT_HEAD table with some information about data taking conditions and a unique identifier for the calibration in the site: the HEAD_ID value. This value is generated from a sequence (MDT_HEAD_SEQUENCE) and inserted in the HEAD_ID filed by the before insert trigger MDT_HEAD_TRIGGER any time that a new row is inserted in the table. The trigger also inserts date and time in the INSERT_TIME field.
- Once that the calibration is identified with the HEAD_ID value (and the SITE_NAME field), processes starts computing tube calibration constants that fill the MDT_TUBE table, always referencing MDT_HEAD table. INSERT_TIME is filled by the before insert trigger MDT_TUBE_TRIGGER. The CALIBFLAG field should be ok if the calibration has no problem, while the VALIDFLAG is 0 by default until validation process is not finished. Data are inserted in the MDT_TUBE table using the Oracle utility SQL*Loader with conventional path (to avoid to disable the triggers on the table).
- Before that validation process can start checking new data in the MDT_TUBE table, materialized views must be refreshed. So, when all rows have been loaded into MDT_TUBE, a row will be inserted in BEFORE_VALIDATION_MDT_TUBE table and the (after insert) trigger TUBE_MATVIEWS_REFRESH_TRIGGER refreshes all materialized views related to MDT_TUBE table.

- Now validation of TUBE constants can start; it reads from materialized views and updates the VALIDFLAG field in the MDT_TUBE table; once the flag is updated the Oracle Streams (configured with the rule that only rows with validflag different from 0 are replicated) may replicate all validated (successfully or not) rows of MDT_TUBE table (it is not necessary to replicate the views since they are only used by the validation process).
- At the same time, the calibration process do the same work in order to calculate rt calibration constants for any region and to write them in the MDT_RT table and in -at least- one of the following tables: MDT_RT_CHEBY, MDT_RT_MAP_R/S/T.
- To do this first information about calibration of rt are computed and written in the MDT_RT table, referencing MDT_HEAD.HEAD_ID; the MDT_RT.MDT_RT_ID value is always obtained using both the MDT_RT_TRIGGER and the MDT_RT_SEQUENCE and identifies the single RT calibration.
Now data about RT calibration must be computed and be written in or more of the table: MDT_RT_CHEBY, MDT_RT_MAP_R/S/T.
- About MDT_RT_CHEBY and MDT_RT_MAP_S, they must reference MDT_RT and SITES tables and their unique identifiers are obtained by suitable triggers and sequences, as for tables above.
- About MDT_RT_MAP_T, we suppose that it will not change so often, so it references MDT_RT table twice, i.e. with two values of MDT_RT_ID that identify the range of RT calibration for which the last row in the TIMES table row is valid (MDT_RT_ID_START, MDT_RT_ID_END), the MDT_RT_MAP_T_ID value is filled as usual with a sequence and a trigger.
-Then, the MDT_RT_MAP_R constants depend from constants in MDT_RT_MAP_T, so MDT_RT_MAP_R references both MDT_RT and MDT_RT_MAP_T. Before to insert a new row into MDT_RT_MAP_R, we must check if the last row in the TIMES table is still valid, if they are we just update the field MDT_RT_MAP_T.MDT_RT_ID_END with the last value of MDT_RT_ID, while the MDT_RT_MAP_T_ID doesn’t change and we insert the new row in the MDT_RT_MAP_R, referencing the MDT_RT_T_ID and the MDT_RT_ID; if the TIMES table is changed we insert the new row in the MDT_RT_MAP_T and obtain a new MDT_RT_MAP_T_ID as usual, this last value will be referenced in the MDT_RT_MAP_R table.
For any row in MDT_RT at least one row has to be written in one of other 4 rt tables, so a check will be implemented to this aim.
- When all rows of RT calibration have been inserted, a new row will be written in the BEFORE_VALIDATION_MDT_RT and the trigger RT_MATVIEWS_REFRESH_TRIGGER will refresh all materialized views related to rt calibration (those views have not been written yet).
- At last, Oracle STREAMS can start to propagate new rows in the tables of Cern Calibration database.