3.5 Architecture Database Design Overview

Robin Lamacraft 2017-04-10

SCOPE

The data held by HRE fits into several compartments:

- Client Component Internal Database Tables: These hold settings for the use of the Graphical User Interface (GUI), etc on the user's device. Typically, these are initialized by the installation, but may be edited by the user. They are copied from the Server when a Project database is opened and on the close of that Project, their current state is saved with the Project. These files are quite small in size and don't increase in size as the Project size grows

- Client Component Caches (some may be in a local database): These hold frequently used data for the GUI to reduce the interrogation of the database within the Server Component. These are initialized and stored on the Client Component and are updated by the Client. The user has no control over them, except for an ability to clear a cache or refresh a cache (like recently used values). At the opening of a Project some of these caches are copied from the Project database and on the close of that Project, their current state is saved with the Project. These files are quite small in size and don't increase in size as the Project size grows

- Server Component Internal Database Tables: These hold settings to manage actions within the Server Component. Typically, these are initialized by the installation, but may be edited by a Project administrator. These files are quite small in size and don't increase in size as the Project size grows

- Auxiliary Database Tables: These are tables that are initialized by the installation. In general, the user can add new records to define new alternatives but the user can't delete the installed records, but can suppress their visibility in the GUI. These tables have many uses and there are many of them. These files are quite small in size and slowly increase in size as the Project size grows. They are infrequently modified, but read more often

- Research Database Tables: These are the tables that hold the user-entered data. These grow at different rates as the Project size increases. It is the efficiency of updating and then compound retrieval from several inter-related tables that is of the most concern in the database design. Some of these tables can become extremely large in large projects

- Research External Files: These are normal files that are referenced by their file path from within HRE. They are often a collection of Image Files, etc. These can require large amounts of disk storage, but are seldom directly modified from with HRE

- Schema Definition Tables: These tables form a separate database within the Server. They define the structure of an HRE Project Database and other Auxiliary information. A parallel read only copy of this database is maintained in the Client Component to speed up the composition of some database queries. Only the installer and plugins may change these database tables to update the HRE Project Database Schema. These table records are marked with ownership, first version and last version usage to ease database upgrading.

Most of the database design discussion is about each table's content and the inter-table relationship within and between the Auxiliary Database Tables and the Research Database Tables.

COMMON DATABASE RECORD IDENTIFICATION AND STATUS FIELDS

To access and manage all records in all database tables, (with a small set of exceptions), all records will have the same group of record identification and record status fields:

- The HRE_ID field is at the core of this structure. It is a 64 bit (BIGINT/INT8) field that contains a unique persistent value that encodes the record table identity and the record instance within that table. That record instance value is maintained in all record rewrites as explained below
- Each database content or structural change is tagged with a unique increasing value Commit ID. Several SQL commands can constitute a Commit. A Commit should modify the current cohesive state of the database to a new cohesive state of the database.
- Each database common field set has 3 Commit ID fields; COMMIT_P, COMMIT_T, COMMIT_N, which are maintained in the following method:
    - On first creation of a record, allocate a HRE_ID, and if this is the first step in a Transaction also allocate the Current Commit ID. Create the new Record with COMMIT_P = 0, COMMIT_T = Current Commit ID, COMMIT_N =0
    - On update of an existing record, copy the Current Commit ID into the COMMIT_N field of the record that is being superseded, copy the COMMIT_T of the superseded record into the new record's COMMIT_T. Now replace the superseded COMMIT_T by its negative.
- This procedure leaves the superseded records within the database as a log of changes by Commit ID that can be purged later. (Useful for creating a change log.)

FIELD NAME	FIELD TYPE	NOTES
HRE_ID	INT8 NOT NULL	The unique HRE_ID of this record (see below).
OWNER_ID	INT8 NOT NULL	The (User)-ID of the owner of this record.
GROUP_ID	INT8 NOT NULL	The (Group)-ID that can access this Record
ORIGIN_ID	INT8 NOT NULL	The (Origin)-ID that stores the Record's Origin (import, etc)
VISIBLE_ID	INT	User visible ID
RIGHTS	INT	If current (User)-Id is not (Owner or Admin)-ID, then use 0 = no access, 1 = read, 2 = edit, 3 = delete
COMMIT_P	INT	Previous Commit Number that changed this record
COMMIT_T	INT	This Commit Number that changed this record
COMMIT_N	INT	Next Commit Number that changed this record

HRE_ID Properties

The HRE-ID is a 64 bit integer (BIGINT/INT8) database field. The lower 32 bits are used for the unique persistent never-reused instance ID. These will be allocated by the Business Rules BR_EntityLink module. The 2^31-1 (4,294,967,295) limit on the number of records in any one table is unlikely to be exceeded without a change to a much more powerful database engine. The instance value 0 (zero) within a table is reserved and does not represent a record.

The upper 32 bits are used to identify which table the record belongs to. This means that an HRE_ID field (not the record identity one) can hold any other HRE_ID. As an example, for a Citation Record where one end will reference the Source being cited but the other end can refer to any other object that can have a Citation attached to it. The range of values used to specify which table the record belongs to is divided into value sub-ranges where the objects, although distinct, behave very similarly to others in that group. There is plenty of scope for this classification within the remaining 32 bits. The classification will be discussed later.

INTERACTION BETWEEN 3 LAYERS OF IMPLEMENTATION

Without detailing all layers of the HRE implementation, let us concentrate on the following 3 layers:

- The Graphical User Interface (GUI) which the user interacts with
- The Business Rules Layer (BR) where all data interactions managed
- The Database Operations Layer where the database content is actually modified or retrieved from external storage.

Three Application Design Rules:

1. The updating of any HRE database table must be the responsibility of only one BR Module
2. There must be at least one direct or indirect way from the GUI to perform every database updating action within a BR Module
3. No other module at any level can modify the database table which is managed by another module.

NOTE: The above rules do not preclude the use of SQL JOINS to perform complex multi-table data retrieval operations.

BUSINESS RULE LAYER MODULES

The following list of Business Rule modules gives an overview of how the core structure of HRE is divided into collaborating services. The proposed database set for a module to manage is described in each BR detail file. (The #0.x suffix is the HRE implementation version where this BR module is first implemented.)

1. **BR_Accent** #0.2 - All aspects of Accent Definitions, Palette Construction and assignment of Palette Swatches to Flag States are applied to Fields that display data of a specified type

2. **BR_EncodedString** - All aspects of strings that have syntactical encoding of different forms. They may be sentence templates, filter scripts, name style output templates, report layout scripts, etc

3. **BR_EntityLink** – All aspects of allocation of HRE_ID values and tools to decode them for Node and Link entities

4. **BR_EventTask** #0.3 – All aspects of Event and Task data management and their relationship to other entities

5. **BR_ExternalFile** #0.5 – All aspects of Evidence stored as linked files to the database

6. **BR_FieldDefinition** – All aspects of data entry field properties, format, bounds, encoding type, validation rules and where data is to be stored

7. **BR_File** – All aspects of file handling for files stored on the Client or the Server component

8. **BR_Filter** #0.4 –All aspects of construction, editing, storage and execution of filter scripts

9. **BR_Flag** – All aspects of Internal and Custom Flag Definitions, Flag Value setting, editing and retrieval for all Node and Link Entity types that can have Flags

10. **BR_Grammar** – All aspects of the addition of grammatical hints in text value data for management of gender, prepositions, plurality, pronouns, etc for narrative reports

11. **BR_GuiElement** – All aspects of identification of GUI elements, for screen configuration, context sensitive help and selective use of multiple languages

12. **BR_GuiTranslation** – All aspects of the GUI language translation as it relates to BR_GuiElement

13. **BR_Help** – All aspects of translated context sensitive help as related to BR_GuiElement

14. **BR_Historical Date** – All aspects of Historical Date and Historical Interval Definitions, Historical Date and Historical Interval value setting, editing and retrieval for Occasions and other uses of Historical Dates and Historical Intervals. Includes date and interval calculations, sorting and calendar conversion

15. **BR_Import** #0.2 – All aspects of controlling and performing import of data from external digital sources to become an HRE Project. Initially, this will control the importing of TMG Projects

16. **BR_Image** #0.5 – All aspects of managing and linking external image format files that can be embedded in reports, etc

17. **BR_InternalText** #0.5 – All aspects of storing large encoded texts as evidence within the HRE database and retrieving them for subsequent use, e.g. DNA test data, XML, etc

18. **BR_Keyboard** – All settings for keyboard management, particularly mapping keys for use in another language, also shortcut key combinations

19. **BR_Location** – All aspects of entry, editing, sorting, and relationships between Locations

20. **BR_Logging** – All controls over the time logging of actions, warnings and errors within HRE

21. **BR_Menu** – All aspects of construction, modification, actions from selection

22. **BR_Monitor** – All settings for monitor management, particularly screen size, multiple screens and user default font sizes and contrast

23. **BR_NameStyle** – All aspects of definition of Name Styles, their editing, their output templates and data entry using Name Styles

24. **BR_Notepad**– All aspects of Internal and Custom Notepad Definitions, Notepad Value setting, editing and retrieval for all Node and Link Entity types that can have Notepads

25. **BR_Notification** – All aspects of composing and managing the logging and displaying of warning messages, error messages

26. **BR_PanelConfig** – All aspects of selecting prior screen configurations, editing and saving screen configurations and applying them to the currently open parent screen

27. **BR_Pattern** – All aspects of defining new Research Data Types from one of the 3 available patterns; Item Entity, Biological Entity, and Group Entity. Acts like a tag but defines a new Research Data Type. Person is a Pattern of a Biological Entity. Place is Pattern of a Group Entity

28. **BR_Person** #0.2 – All aspects of entry, editing, sorting, and relationships between Persons

29. **BR_Place** #0.9? – All aspects of entry, editing, sorting, and relationships between Places. For use in one-place studies

30. **BR_Pointer** – All settings for the pointer, mouse, touch screen used to move the cursor around the monitor, usually about scroll rates, sensitivity and extra mouse buttons

31. **BR_Printer** – All settings for the selected HRE printer. usually about paper sizes, print quality etc

32. **BR_Project** – All aspects of Project management involving opening, copying, backup, restore, etc

33. **BR_RecentUse** – All aspects of creation, maintenance and retrieval of recently used value lists for selected data entry fields

34. **BR_Reminder** – All aspects of creation, maintenance and retrieval of the reminder content as related to BR_GuiElement

35. **BR_Report** #0.6 – All aspects of creation, maintenance, retrieval of report generation scripts and their execution

36. **BR_Setting** – All aspects of creation, maintenance, retrieval of the settings, both for the current project, current user device and current user preferences

37. **BR_Source** – All aspects of entry, editing and storage of Sources, Repositories and Citations

38. **BR_Subset** #0.5 – All aspects of creation, storing, using and operating on lists of HRE_IDs, to create new lists or to use those lists to control the content on the screen, within a calculation or as the focus list for a report. This is a more flexible version of the TMG Focus Group. It can apply to any list of HRE_ID from the same database table

39. **BR_Substitution** – All aspects of creation, editing, storing, and evaluation of syntax encoded variable substitution as is found in Name Style Output Templates, Sentences and Embedded Citations in Memos

40. **BR_Tag** – All aspects of creation, editing and use of the Tag Definitions

41. **BR_User** – All aspects of creation, editing and use of the User Identity and data access rights

42. **BR UserTranslation** - All aspects of User-created data translations, and their data storage and retrieval with possible linguist grammar hinting. The translated strings can also be encoded for content substitution as occurs in narrative sentence templates

43. **BR_Viewpoint** – All aspects of creation, editing and use of Viewpoint collections of screen layouts.