MGM AIC Project – Detail Requirements of the Milestones 1, 2, and 3

(**Note:** All TG's and SQL-Tables are defined in the AIC Design.xlsx. The system architecture is defined in the AIC System UML)

*** Terminology ***

Tech Stack Angular (front-end) + Node.js (or Nest, backend) + TypeScript (the only programming language for both Front & Back end) + Tabulator Trees (aka Treegrid) (https://tabulator.info/examples/5.5#tree) + PostgreSQL database (The only persistent storage for all TG's on Server) + Cloud Infrastructure Tools (**Note:** We have switched the TG Library from Syncfusion to the open-source Tabulator)

PWA Client Progressive Web App shall be 100% Responsive on all the 'Clients': Windows & Mac browsers (Chrome/Edge/Safari), Android browsers (Chrome/Firefox), and iOS browsers (Safari/Chrome).

System Capacity The whole 'System' shall run on a single Linux server on AWS, said single server shall pass the Stress Test of "Hosting 1M Product-TG's and 10M concurrent Clients each viewing 1-or-more TG".

AJAX When Client is viewing a TG, and said TG receives an update from Server, said TG's update on Client's screen should be performed in AJAX fashion (i.e. No webpage-refresh unless switching TG).

Browser Tab Each TG shall always vertically occupy the Client's browser Tab session (at any current browser window size), always enable Vertical & Horizontal default-scroll-bars, but no 'Pagination bar'.

Corrsp Abbreviation for the word 'Corresponding'.

TG A Treegrid (with its TG-ID in t-TG.TG). Each TG is represented by a corrsp TG-Row in Master-TG (See 'Master-TG' paragraph).

TG-Row All the TG-Row's are contained in the Master-TG. Each TG-Row represents a unique TG, with a unique TG Name-Cell and a unique TG-Cell (i.e. said TG's TG-ID).

Column A Column (i.e. Col) in any TG. Each Col (in a TG) corresponds to one of the fields in the Add/EditRow-dialog. (All the Dialogs in the system are Non-Modal, they can be moved/resized by Users)

TG/Row/Cell/Item A TG (of any TG Type) contains 1-or-more 'Rows (t-Row records)'. A Row contains Cells (t-Cell records) of corrsp Cols (t-Col records). A Cell contains 0-or-more 'Item(s) (t-Item records)'.

xyz-Column The Col with its Column Name-cell = xyz (in any TG).

xyz-Cell The Cell (in any Row) which corresponds to the xyz-Col. We may simply call 'xyz-Cell' as 'xyz'.

xyz-Item# The 'number(#) of Item(s)' in an xyz-Cell. In the Add/EditRow-dialog, users can Drag-n-drop each Item to 'Move' its position (i.e. t-Format.Item-Order) among all existing Items in said xyz-Cell.
e.g. Enclosed is a picture of '3 Items in a Cell' in any Row (contained in any TG). Said Cell can also be a Field in the 'Add/Edit/FormatXXXdialog', and in any said 'dialog', the user can 'Right-Click on any Existing Item or any Blank Field' to pop-up a '2nd-level Edit-Item dialog'. Please refer to Rule.#EditItem.

Data-Item:1 Data-Item:2 Data-Item:3

xyz-Value Locate the row (contained in a TG) with its Label-Cell=xyz, (if found) then, the 'Value-Cell of said row' is called the 'xyz-Value of said TG'.

Label-Cell* For each Row: If its Label-Cell has a * symbol, meaning the Value-Cell (of said Row) is a 'Mandatory Cell' which contains 1-or-more 'Non-NULL Item(s)' (except when it is in a Category-TG).
If a 'Mandatory Cell' is missing Data, said Cell's corrsp t-Format.Status ∋ ('∋' means said Cell contains) Missing Data (except when it is in a Category-TG).
If a 'Visible Row' has 1-or-many Cell(s) with corrsp t-Format.Status ∋ Missing Data, said Row's corrsp t-Format.Status ∋ Missing Data also (except when it is in a Category-TG).
If a TG has 1-or-many Visible Row(s) with corrsp t-Format.Status ∋ Missing Data, said TG's corrsp t-Format.Status ∋ Missing Data also.

Value-Cell Each Row that contains the Label-Cell: Said Label-Cell's matching 'Value Type-Cell (in All-Label)' is called the 'Value Type' of said Row, said Label-Cell's matching 'Value-Format-Cell (in All-Label)' is called the 'Value Format' of said Row, said Label-Cell's matching 'Value Default-Cell (in All-Label)' is called the 'Default Value' of said Row.
The system shall validate the Value-Cell (when it's being Added/Edited by the User) according to the 'Value-Type' of said Row, and system shall reject said Add/Edit if the User fails said Validation.
E.g. The system shall pop up a Validation-dialog (with 2 buttons: OK, Cancel) when the user Add/Edit a Social-ID (for Social account login), Email, Phone#, etc.

Regular-TG A TG (represented by a TG-Row in Master-TG) with its TG Type-Cell = 'Category | Product | Level-Set | Search-Set' and Row-Level = 1.

Version-TG Each TG has 0-or-more 'child Version-TG(s) (represented by TG-Row's with Row-Level=2 in Master-TG)' as the alternative Copy(s) of said parent TG, each Version-TG (represented by TG-Row) was assigned a 'TG-Version (:DDT of Versions)'. Each Version-TG contains mostly the 'Shared Rows' that are shared (via 'Share Row' menu entry) from its 'parent TG (in the Master-TG)'.

Green Menu Entry (to Edit) TG. The user or system Edits a TG on the server, once the update succeeds, the server shall push said update to all the Clients that are currently accessing said TG.

Format-Record The 'SQL DB Table t-Format' contains all the Format-Record's in the system, i.e. each 't-Format record' is a Format-Record.

TG-Format Format-Record of a TG.

Col-Format Format-Record of a Column (in TG).

Cols-Format Format-Records of all Columns (in TG).

Row-Format Format-Record of a Row (in TG). (The 'Format Local/Share-Row' menu entry shall update corrsp Row-Format).

Rows-Format Format-Records of certain Rows (in TG).

Cell-Format Format-Record of a Cell (in a Row in TG). (The 'Format Cell' menu entry shall update corrsp Cell-Format).

Cells-Format Format-Records of certain Cells (in certain Rows in TG).

Item-Format Format-Record of an Item in a Cell (in a Row in TG). (The 'Format Item' menu entry shall update corrsp Item-Format).

Brown Menu Entry (to Edit) the Format-Record(s). Once the edit succeeds, the server notifies all Clients 'currently affected by said Edit' to click (the blinking) Refresh (on Menu-Bar) to reflect the said update.

Categories-TG Each TG-Row in the Categories-TG represents a 'Category-TG' with 1-or-more 'parent Category-TG(s)' and 0-or-more 'child Category-TG(s)' inside the Categories-TG.

Category-TG Each 'Category (TG)' is listed in Categories-TG as a TG-Row. e.g. 'Fish' (Category-TG) is a descendant of 'Food' (Category-TG).

All-Category A subset of TG-Rows (in Master-TG) with Row-Level=1 & TG-Type=Category, each TG-Row (in All-Category) represents a Category-TG.

All-Product A subset of TG-Rows (in Master-TG) with Row-Level=1 & TG-Type=Product, each TG-Row (in All-Product) represents a Product-TG.

ab (:TG-ID) Column ab is of TG-ID Data-type, each corrsp Cell represents (an existing) TG-Row's (corrsp TG's) ID in Master-TG.

ab (:Row-ID) Column ab is of Row-ID Data-type, each corrsp Cell represents (an existing) Row's ID (i.e. t-Row.RowID) in any TG.

ab (:Col-ID) Column ab is of Col-ID Data-type, each corrsp Cell represents (an existing) unique Column in a TG.

ab (:Item-ID) Column ab is of Item-ID Data-type, each corrsp Cell represents the unique ID of an existing (Data-)Item (in a Cell, of a Row).

ab (:Text) Column ab is of Text (i.e. Rich Text, not plain text) Data-type, the system shall pop-up 'Rich-Text Editor' to let users edit ab-Cell. (Editor.js enables WYSIWYG editing, it outputs in JSON format !!)

ab (:ML-Text) Column ab is of Multilingual-Text Data-type, system pop-up 'ML-Text-Editor-dialog (Refer to 'ML-Text' sheet in AIC Design)' to let users edit ab-Cell (as key-pairs of "*Lang* (:DDT)"=>"*Rich Text…*").

ab (:URL) Column ab is of URL Data-type, the system shall pop-up 'URL Editor-dialog (with webpage backup)' to let users edit ab-Cell (as "Original/Backup URL"=>"*Corresponding URL*" key-value pairs).

ab (:DDT of xyz) Column ab is of DDT (i.e. Dropdown Tree) Data-type, with 'DDT-Cell=xyz as the Prarent Row and all its Descendant Rows in All-DDT' as its DDT-Source.

(a) DDT-Source A Row (representing all its 'Descendent Rows' in the TG) which is assigned to the DDT Source field in an Add/EditCol-dialog (of 'DDT' Data-type).

ab (:Color) Column ab is of Color (RGB Hex Color Code) Data-type, the system shall pop-up a Color Palette when editing corrsp Cells.

ab (:Formula) Column ab is of Formula Data-type, system parse/execute 'Excel-like Formulas (Refer to MS Excel Formula & Func) – but replacing Excel-Row/Col with TG-Row/Col, or Msg: 'Error: Invalid Formula'.

ab (:User-ID) Column ab is of User-ID (i.e. RowID of a Row in Profiles-TG with its Row-Type='User') Data-type.

Show-Set A set of 1-or-many Visible Row(s) in a TG. (See 'Show-Set' paragraph)

Row-Level The 'Hierarchical-Level (:Int, ≥0)' of a Row in any TG.

Hierarchy-Cell The Cell (of a row) corresponds to the 'Hierarchy-Col (i.e. the Column that is showing the Row-Level Hierarchy)'.

Level-Set The Row-Format settings for each of the existing 'Row-Level(s)' of Current-TG. (See 'Level-Set' paragraph)

Col/RowLock Column-Edit Lock and Row-Edit Lock. (See 'Locks' paragraph)

Lock-Period Length (N sec) a user may hold each acquired Row/ColLock before it is Force-released (individually) by the server: When its Lock-Period ends, or, when the Timer lost connection with the server.

Threshold Server tolerates temporary disconnection from each Client for up to the 'Threshold (in seconds) in said User' before the server shall take action against said Client.
The 'Threshold' of each user is stored (and adjusted by the system based on the current network condition) as a Row in said user's User-Profile.

Recycle Bin 'Recycle Bin (RB)' contains all the Rows deleted from any of the TG's. (The system only manages one 'Global RB' on the server.)

(per) ACL (according to the user's) ACL (Access Control Level) Rank

CRUD C: Create, R: Read, U: Update (Edit), D: Delete (an Item/Cell/Row/Col/TG)

Underlined (placeholder of) Data Variables or Column Names;

Dimmed A Menu entry, Button, Toggle, Checkbox, Field (in dialog) is 'Dimmed' i.e. 'Grayed-out and Read-only': Whenever the system determines said object to be currently 'inactive' to the Current-User.

PC/Mac vs Mobile 'Mouse-hover over / (Single Left-)Click / Right-Click' on the PC/Mac – is equal to – 'Single-press / Double-press / Long-press' on the Mobile. By default, we refer to the PC/Mac UI in this doc.

Rule.#Topic The general 'Rule #Topic' applies whenever/wherever fits the described situation in the doc.

Ex.#Topic The 'Example #Topic', usually when this example was referred to more than once in the doc.

(RowX's) Parent(s) ⇔ All direct ancestor row(s) of RowX (in a TG).

(RowX's) Ancestors ⇔ All direct & indirect ancestor rows of RowX (in a TG).

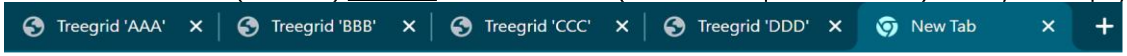
(RowX's) Child(ren) ⇔ All direct descendant row(s) of RowX (in a TG).

(RowX's) Descendants ⇔ All direct & indirect descendant rows of RowX (in a TG).

*** Milestone 1 ***

TG-Tab-Bar & Menu-Bar

On top of the webpage, there is a 'TG-Tab-Bar' with 1-or-many TG-Tab(s) (similar to Browser-Tabs), User clicks '+' (on TG-Tab-Bar) to add a new TG-Tab, and each TG-Tab contains 0-or-1 TG (selected from All-TG).
Each TG-Tab shows the (truncated) TG-Name of the Current-TG (as in the example shown below). The system displays the (default) content in each new TG-Tab according to the setting in User-Profile.



User clicks any existing TG-Tab (on the webpage) to switch to the TG (in TG-Tab) as 'Current-TG': The Current-TG shall vertically occupy (any current size of) the screen area below its Menu-Bar.
TG-Tabs allows users to 'Copy/Move/Share Rows' between 2 TG's – The use of TG-Tabs simplifies the implementation of these features by performing them within the same webpage.

Right below the TG-Tab-Bar, there is a 'Menu-Bar (if Screen-width ⩾ MenuBar-Min-Width in User-Profile)' or 'Navicon-toggled-Menu (if Screen-width < MenuBar-Min-Width in User-Profile)' inside each TG-Tab.
The 'Blue' section of Menu-Bar: Their current values (per Current-User & TG-Main) are always stored in Cookies, said values are 'Restored' when 'matching User opens matching TG-Main' next time. (Ref. Persistent)

| Refresh | All Cat | Compare ☑ | User | Lang | TG | Version ☑ | Col's | View | Expand: N ⇳ | Sort ☑ | Filter ☑ | Level-Set ☑ | Show-Set ☑ | Timer ☑: Countdown & Locks Wait Status... (scrolling text) |
Tooltips for Menu-Bar: (Tooltip: Session-ID) (Tooltip: TG Name & TG Version) (Tooltip: 'Lock Timer: Countdown & Locks Wait Status – Full Details')

User clicks User on Menu-Bar to pop-up UserLogin-dialog with these buttons: Log-in, User-Profile, Log-out, Register, Cancel;
Log-in: Log in (to be the Current-User) using Email(optional)/Mobile#(optional) or one of the 'Previously validated' Social-ID(s) (currently in his User-Profile) with Captcha (to validate he is human).
The button is 'Dimmed' if already logged in. Once the user login successfully, the system shall open the designated TG (in the Current-Tab) as defined in his User-Profile.
User-Profile: The system adds a new TG-Tab for opening User-Profile (as TG) of Current-User, he can Edit (per ACL) rows in User-Profile. Button 'Dimmed' if not currently logged in. (See User-Profile paragraph)
Log-out: Log out of the Current-User. Button 'Dimmed' if not currently logged in. (**Note:** Please refer to Passport.js, the open-source library supporting 500 Strategies of Social Logins, including all that we need)
Forgot Password (optional): Help users Reset Passwords via their Email/Mobile#(SMS)/Social-ID with Captcha.
Register: Register a new User with his Email(optional)/Mobile#(optional) or his existing Social-ID with Captcha. Button 'Dimmed' if currently logged in.
Cancel: Discard the current edit and close the dialog. (**Note**: The system shall support all the Social-Logins that are listed under Social-ID in All-DDT of the 'Data' Sheet in 'AIC Design' Excel doc)
When the user Mouse-hover over User: The system pop-up a 'Tooltip (under Menu-Bar)' to display the current Session-ID.
Session-ID: The unique Browser-Tab(not TG-Tab)-Session-ID of Current-User (logged in) or Anonymous Viewer (not logged in), i.e. the 'Client' Session. (e.g. 'User:ABC.Tab:5', 'Anonymous:234.Tab:2', etc).
The Refresh button shall 'Blink' when the TG-Main has received an update from Server, the Current-User clicks the blinking button to show the 'updated' TG-Main on the screen (in AJAC fashion, no webpage refresh).
User clicks All Cat: The system shall add a new All-Cat-Tab (a special TG-Tab). (See All-Cat-Tab paragraph)
User set 'Compare ☑' Checkbox=On: System shall switch the Current-TG (i.e. a Product-TG in All-Cat-Tab) to 'Product-Comparison-Mode'. Off (default): Back to normal. (See Product-Comparison paragraph)
'Compare ☑' is 'Dimmed' by default, it is Only Enabled when the said Product-TG is currently opened inside an All-Cat-Tab.
When Mouse-hover over TG or Version: The system pop-up a 'Tooltip (under Menu-Bar)' to display the Current-TG's (i.e. its corrsp TG-Row's) TG Name-Cell and TG Version-Cell.
Rule.#New-TG: When the system creates a new TG with its corrsp (TG-Row in Master-TG) with Row-Level=1, said TG-Row shall have its TG-Version=Default.
A newly created TG shall always have its corrsp (TG-Row in Master-TG with) TG-Owner=Current-User (:User-ID; i.e. himself) and TG-Status=Private (DDT of Statuses).
Only users with the Editor ACL (or above) may remove said 'Private (Item)' from said 'TG-Status-Cell (Cell)', once said Item is removed, said TG is no longer Private and it shall be Visible to all other users.
When the Current-TG has its TG-Version=Default, the 'Checkbox of Version (on the Menu-Bar)' is 'Dimmed'.
User clicks (per ACL) Version to pop-up TGVersion-dialog with 3 buttons: Open, Save As, Close:
Open: When the user clicks Open, the system pop-up OpenVersion-dialog (with 1 button: Close) to display the TG-Version-List (of Current-TG) and lets user open (via 'Open TG' menu entry) an existing TG (e.g. ABC) in said list, then, the system adds a TG-Tab for opening said TG ABC, also, the system shows the TG ABC's TG-Version-Cell inside the Version button (on Menu-Bar in the new TG-Tab).

The user can also shuffle the display order of each TG (via 'Move Next' menu entry) in said TG-Version-List, the system shall automatically put 'the TG currently displayed on the top of TG-Version-List' to become the 'parent TG (represented by TG-Row with Row-Level=1) of said TG-Version-List' in (by automatically updating the) Master-TG.

Save as: Use the Current-TG to create or update another Version-TG in the TG-Version-List (of Current-TG). When the user clicks Save As, the system displays TG-Version-List (of Current-TG) and lets the user select an existing Version-TG (other than Current-TG) in said list, then, the system shall use the content of Current-TG (in Current-Tab) to replace/overwrite said selected TG-Version.

If the user clicks '+' (instead of selecting an existing Version-TG) in the TG-Version-List (of Current-TG), the system shall use (the full content of) Current-TG to create a new Version-TG in the TG-Version-List (of Current-TG), the system shall pop-up a 'Rich-Text Editor-dialog' for the user to edit the 'TG-Name-Cell (:ML-Text) and TG-Version-Cell (:DDT of Versions)' of the newly created Version-TG.

Only a 'Private TG' (i.e. TG with its TG-Status ∋ Private) may have a duplicate TG-Name-Cell in the system, or the system shall reject said update with an error msg.

Only a 'Private TG' (i.e. TG with its TG-Status ∋ Private) may have a duplicate TG-Version-Cell in the current TG-Version-List, or the system shall reject said update with an error msg.

e.g. A TG-Version-List contains a '(parent) TG with its TG-Version=Vendor', and a 'child Version-TG with its TG-Version=Consumer'. The user may click Save As to create a new Version-TG with TG-Version=Vendor (i.e. duplicate TG-Version), however, if user tries to remove 'Private (Item) from said TG-Status-Cell (Cell)' of the new Version-TG. the system shall reject said action with an error msg.

User set 'Version ☑' Checkbox=On (default): The system opens an existing Version-TG (instead of Regular-TG) (specified by its TG-Version currently shown in Version button on Menu-Bar) as the Current-TG;

User set 'Version ☑' Checkbox=Off: System opens 'said Version-TG's parent TG (in Master-TG)' as Current-TG, however, the TG-Version (currently shown inside Version button on Menu-Bar) remains unchanged.

Rule.#Private-TG: A newly created TG shall always have its corrsp (TG-Row in Master-TG with) TG-Owner=Current-User (:User-ID; i.e. himself) and TG-Status ∋ Private (DDT of Statuses). Only users with the Editor ACL (or above) can remove said 'Private (Item)' from its corrsp 'TG-Status (Cell)', once removed, said TG is no longer Private and it shall be Visible to all other users.

User clicks Lang: System shows 'DDT of Display Language(s)-Value (from his User-Profile)' to let User set the UI Lang. Once set, system shall auto-translate (i.e. Google Translate) all content on screen into said Lang.

User clicks TG: System pop-up the AllTG-dialog and displays All-TG (as TG) inside the dialog. The user selects (via 'Open TG' menu entry) an existing TG (e.g. ABC) in All-TG, the system shall add a new TG-Tab for opening 'the (last opened) Version-TG (if exists) of said TG ABC', also, the system shows the Version-TG's TG-Version inside the Version button (on Menu-Bar of the new TG-Tab). If said TG ABC has no Version-TG or none of its Version-TG(s) were opened last time (at the last session of Current-User), the system shall open TG ABC itself as the Current-TG and show Default inside Version button on Menu-Bar.

Rule.#Cannot-Switch: While 'Timer ☑' Checkbox=On (on Menu-Bar), the user cannot switch the Current-TG (of Current-Tab), or the system shall reject with Msg: 'Cannot switch Current-TG(s) when Lock-Timer is On'.

User clicks Col's: System pop-up ChooseCol-dialog to let the user show/hide each existing Columns in the Current-TG (but the 'Hierarchy-Col' cannot be hidden), each 'column that is currently hidden' shall have its corrsp t-Format.Status ∋ Hidden. (See: Demo of ChooseCol using 'headerMenu')

User clicks View/Edit: System shall switch the Current-TG between View mode (default) and Edit mode (per ACL). All the Context Menu items in Green (i.e. Editing features) are 'dimmed' in View mode.

'Expand ⇳': User sets 'Expand-Level (≥ -3)'. When Expand-Level ≥ 0, the system shall Collapse (i.e. hide) the rows with Row-Level>Expand-Level, user can still Expand/Collapse any row(s) manually without changing the Expand-Level (on Menu-Bar). E.g. Expand-Level=99: The system collapses nothing, the system shows all the Visible rows.

Ex.#Expand-Level: A DDT-Source (:Row-ID) represents Label-Value's with RL0 (Row-Level=0): Continent(s), RL1 (Row-Level=1): Country(s), RL2: Region(s), RL3: State(s), RL4: County(s), RL5: City(s).

If Expand-Level = 2, the system Only shows rows with 'RL0: Continent(s) or RL1: Country(s) or RL:2: Region(s)', and Collapses (hide) other rows.

When Expand-Level = 1, the system only shows the visible rows with Row-level = 0 or 1, and collapses (hides) all other rows.

When Expand-Level = 0, the system only shows the visible rows with Row-level = 0 (i.e. Section-Head rows), and collapses (hides) all other rows.

When Expand-Level = -1, the system shows all the 'Visible (i.e. Non-Hidden)' and 'Hidden' rows.

When Expand-Level = -2, the system shows all the Visible and 'Hidden' rows, also, the system shows all the 'Hidden' Cells.

When Expand-Level = -3, the system shows all the Visible and 'Hidden' rows, also, the system shows all the 'Hidden' Cells and all the 'Hidden' Items.

User set 'Sort ☑ (the 'Sort Columns' function)' Checkbox=On: All combinations of Sorting(s) are allowed, but sorting only applies to the Rows with their 'Row-Level = The current Expand (on Menu-Bar)'.

Off (default): No sorting is allowed, turn off the 'Sort Columns' function. (See: https://tabulator.info/examples/5.5#sorters)

User set 'Filter ☑' (the 'Filter Columns' function) Checkbox: Turns On/Off(default) the 'Filter Data in Col Header' Mode. The filter reacts instantly to the user's 'Typeahead' key-in. (See Filter Data In Header)

When the Current-TG is (a subset of) All-Product, the system adds a 'Filter Field (:DDT of All-Product-Search-Set)', user can select (and activate) a Product-Search-Set as the filter (of Current-TG).

User clicks (per ACL) Level-Set to pop-up LevelSet-dialog with 3 buttons: Open, Save As, Close: Open (open a 'Level-Set' selected from the All-Level-Set TG) / Save as (use the current Level-Set to create or update another Level-Set in the All-Level-Set). (See 'Level-Set' paragraph)

During Save As, the system pop-up a 'Rich-Text Editor-dialog' for the User to key in 'TG-Name-Cell (:ML-Text) of said Level-Set (in Master-TG)'. (e.g. Level-Set's TG-Name = 'Vitamins', 'Car Services', etc).

User set 'Level-Set ☑' Checkbox=On (default) to Activate (enable) the current Level-Set (on the Menu-Bar) for the Current-TG, the current Level-Set (:Level-Set-ID) is saved in t-Format.TG-Level-set.

User set Checkbox=Off: Deactivate (disable) the current Level-Set (if exists), but the 'Level-Set (shown on Menu-Bar)' remains unchanged.

Rule.#Show-Set: When a row has its ShowSet-Toggle-Cell≠NULL, this row is 'Included' in the 'Show-Set (i.e. A set of rows with their ShowSet-Toggle-Cell≠NULL)'. (See Show-Set paragraph)

User set 'Show-Set ☑' Checkbox=On: Activate the ShowSet-Toggles (with all previously saved ShowSet-Toggle-Cell's).

'Show-Set ☑' Checkbox=Off (default): Deactivate the ShowSet-Toggles (but the system Saves all the existing ShowSet-Toggle-Cell's).

When the server sets 'Timer ☑' Checkbox=On: Timer starts auto-countdown (for the length of 'Lock-Period'), showing (scrolling text) Msg of 'Already-acquired lock(s) & Currently-waiting lock(s) real-time status';

When the server or user sets 'Timer ☑' Checkbox=Off (default): This action shall Cancel the current Timer, also, it Cancels the user's current Action (i.e. the action which triggered the current Timer).

In 'Edit' mode, when the user Mouse-hover over Timer: The system pop-up a multi-line 'Tooltip' to show all the (system-wide) currently active Locks and their wait status (of all the users).

e.g. Each active ColLock is displayed as a line in the dialog, as: "ColLock: TG:TG-ID.Col:Col-ID has been obtained by User1.Tab2 for N sec (out of max M sec); Its waiting queue: User3.Tab4, User5.Tab6"

Each active RowLock is displayed as a line in the dialog, as: "RowLock: TG:TG-ID.Row:Row-ID has been obtained by User1.Tab2 for N sec (out of max M sec); Its waiting queue: User3.Tab4, User5.Tab6"

Below the 'Menu-Bar', the 'Current-TG' shall stretch vertically - at any current screen size (i.e. Fully Responsive).

The system enables both Vertical & Horizontal Scroll-Bar's for the current TG (not affecting TG-Tab-Bar & Menu-Bar, they are always on-screen), also, always Disable the 'Pager Bar' (i.e. Each TG is a single webpage).

The webpage is 'Fully-Responsive' on the Windows/Mac browsers (with keyboard+mouse), and, Android/iOS browsers (1-finger-gestures in one-hand mode, and, 1&2-finger-gestures in two-hand mode).

e.g. Mobile users can perform either 1-finger-double-tab or 2-finger-zoom-gesture to Zoom-in/Zoom-out the current Treegrid.

Users can Drag-n-drop the 'edge' of a Column-Header to 'Resize' said column, also, Drag-n-drop a Column-Header to 'Move' its position (i.e. Col-Order) among all the Visible-Columns.

When a menu entry has the ☑ symbol - meaning the item shall function as an On/Off Switch, while its ☑ or ☐ symbol (currently displayed on the item) reflects the 'Current On/Off Status' of this Switch.

When a field (in any dialog) has the ☑ symbol - meaning this field is of Bool Data-type, while its ☑ or ☐ symbol (currently displayed on the field) reflects the 'Current Yes/No' of said field.

Context Menus

Column Menu – The user Right-click a 'Column-Header' in Current-TG, to pop-up (per ACL) Column Menu with these menu entries:

Edit Column: The user can Edit a Column via pop-up EditCol-dialog (with 2 buttons: Save, Cancel), and the fields (* means Mandatory) in said dialog are:
Column Name* (⇔ t-Col.Col-Name), Column Data-Type* (⇔ t-Col.Data-Type), Column DDT-Source (⇔ t-Col.DDT-Source; Said field is mandatory when Column Data-Type = DDT),
Column Default-Data (⇔ t-Format.Default), Column Status (⇔ t-Format.Status), Column Calculated? (⇔ t-Format.Calculated), Column Calc-Formula (⇔ t-Format.Calc-Formula).

Format Column: The user can Format a column via pop-up FormatCol-dialog (with 2 buttons: Save, Cancel), and the fields in said dialog are:
Hierarchy Column? (:Bool; Helps determine its containing TG's t-Format.TG-Hierarchy-Col), Freeze Column? (:Bool; Helps determine its containing TG's t-Format.TG-Freeze-Col), Column Min-Width (⇔ t-Format.Col-Min-Width), Column Data-Formula (⇔ t-Format.Data-Formula), Column Font-Style (⇔ t-Format.Font-Style), Column Comment (⇔ t-Format.Comment), Column Audit-Trail (⇔ t-Format.Audit-Trail).

Each Column Name-Cell is unique in a TG, if the system detects the current Column Name-Cell in Add/EditCol-dialog is not unique, Msg: "Duplicate Column-Name found in Treegrid" inside the Dialog.
Rule.#Calc: A 'Calculated Cell/Item' is auto-calculated (by the system in real-time) based on its (existing) dominant (most relevant) 'Item/Cell/Col Calc-Formulas (:Formula)'.
Foreign-Data: It's the Foreign-key to records in an external database or online data sources, e.g. PostgreSQL, MongoDB, or Web services.
DDT-Source: A 'Subset of Rows' in All-DDT (as the source to said DDT Column). The DDT-Source field is 'Dimmed' by default, Only Enabled when Current-Column's Data-Type is 'DDT'.
A Cell of 'DDT' Data-Type contains 0-or-many DDT-Item(s), the user may select DDT-Item(s) from 'all visible Hierarchy-Cell's listed under the DDT-Source'.
Any 'System-Column' is a read-only field that is 'Dimmed' in Add/EditCol-dialog, only the Admin ACL can edit this field. e.g. 'RowID' is a System-Column.
Rule.#Initial-Cols: Each TG in the system shall be created with these 'Initial Columns': RowID* (⇔ t-Row.RowID, :Row-ID), Row* (⇔ t-Row.Row, :Row-ID), Share (⇔ t-Row.Share, :Row-ID), Row Status (⇔ t-Format.Status, :DDT of Statuses), Row Type (:DDT of Row Types), Row Comment (⇔ t-Format.Comment, :ML-Text), Row Audit-Trail* (⇔ t-Format.Audit-Trail, :Formula; History of all the previous updates on said row since it was created, the system shall always keep all Audit-Trails up-to-date). (When updating a Col's Data-Type, all the affected Cells of said Col shall update their corrsp Audit-Trails)
For each row: 'Row-Type' identifies the type of Row in TG, and 'ShowSet-Toggle' stores the 'ShowSet-Toggle-Cell' of said Row.
A Col is 'Calc Col' when its corrsp Col-Format's t-Format.Status ∋ Calculated.
A Col is 'Read-only Col' when its corrsp Col-Format's t-Format.Status ∋ Read-only, or, its corrsp TG-Format's t-Format.Status ∋ Read-only.
A Col is 'Hidden Col' when its corrsp Col-Format's t-Format.Status ∋ Hidden: A Hidden Col is not displayed in the TG containing it.
A Col is 'System Col' when its corrsp Col-Format's t-Format.Status ∋ System: A System Col is only Editable per Admin ACL. (e.g. Each RowID-Col is a System Col)
The 'Row-Types' DDT-Item's are: Name (row with its 'Value-Cell' as the unique Name), User (row represents a Registered User in Profiles-TG), Unit (row represents a Unit in Units-TG), StdUnit (row represents a Standard-Unit in Units-TG), Category (row represents a Category-TG), Product (row represents a Product-TG), Section-Head (row as a Section-Header), Definition (row represents basic info of Current-TG), Part-of (row represents a portion or BOM of its Parent Row), Ingredient (row represents an Ingredient), Key-Ingredient (row represents a Key Ingredient), Packaging (row represents a Packaging info), Serving (row represents a Serving info), Formula (row represents parts of a Formula), Price (row represents a Price), Best-Price (row represents the Lowest Price), Review (row represents a Product Review), Best-Review (row represents the Best Review), Worst-Review (row represents the Worst Review), Feature (row represents a general feature).
The dialog shall acquire ColLock, user can wait or click Cancel anytime. Once ColLock is acquired, the Timer (inside Dialog) restarts auto-countdown: User clicks Save or Forced-cancel when time-up. Lock released.
The dialog shall Save the 'Column Default-Data field data (e.g. of Col-X)' as a 't-Item record (e.g. Item-Y)', and set the Col-X's corrsp t-Format.Default (:Item-ID) ⇦ Item-Y's t-Item.Item (:Item-ID).
Rule.#DataType-Change: When the user changes 'Data-Type' and clicks SAVE, the system shall instantly convert all the existing 'Non-NULL Cell's corrsp to this column. For each Cell that failed to convert (e.g. Convert 'ABC' to the Int data-type shall fail): The System shall replace said Cell with the 'Default Data of the new Data-Type'.
'Hierarchy Column?'=Yes: This is the Only column showing hierarchy levels (e.g. Row-Level:1/2/3) in this TG. NULL/No: This column is not showing hierarchy. Only one column may show hierarchy in each TG.
Rule.#Data-Formula: t-Format.Data-Formula is Formula with 'i18n Format & Parse (Similar to URL)' and 'Conditional Formatting (Similar to URL)'. 'Data-Formula' does not modify data, only sets the data format.
Rule.#FontStyle: User clicks any 'Col/Row/Cell/Item Font-Style' field to pop-up FontStyle-dialog (with 2 buttons: Save, Cancel). System saves all Field-data's in said FontStyle-dialog into one t-Item.JSON-Data.

View Col: User can View a column (via ViewCol-dialog with 1 button: Close): Display all the fields from EditCol-dialog and FormatCol-Dialog, all fields are read-only here. No Lock is required.

Add Col: The user can (per ACL) Create a new Unique column (on the right side of the current Column) via AddCol-dialog (similar to EditCol-dialog): Save shall create the new Column, Cancel does not.
In Add/EditCol-dialog: If the user sets it as 'Mandatory Col', the system shall display said Column Default-Data as Column Default-Data*, meaning: A Mandatory Column cannot have a NULL Default-Data.

Del Col: The user can (per ACL) Delete a column (except any of the Initial-Columns, or, the current Hierarchy-Col) via DelCol-dialog with 2 buttons: OK, Cancel, the dialog requires ColLock and activates the Lock Timer. The system shall prompt the User to confirm if Deleting the current Col (before the Timer runs out). The Column deleted by 'Del Col' cannot be undone.

Freeze Cols ☑ Checkbox=On: Freeze the current column and all its left-hand-side columns (together), the 'Col Order (:Int)' of the Current-Col is saved in the corrsp t-Format.TG-Freeze-Col.
Checkbox=Off (default): Unfreeze all columns, the 'TG Freeze-Col' (on the Menu-Bar) is not active. See: https://tabulator.info/examples/5.5#frozen-col

Row Menu – The user Right-click the 'Row-Header' of any row in the Current-TG, to pop-up 'Row Menu' (per ACL) with these menu entries:

Add Prev, **Add Next**; **Add Child**; **Edit Row**: Users can (per ACL) Add a row as previous/next sibling (via AddRow-dialog); Add a row as Child (via AddRow-dialog); Edit a row (via EditRow-dialog).
EditRow-dialog has 2 buttons: Save, Cancel. The dialog shall acquire RowLock, the user can wait or Cancel anytime.
Once RowLock is acquired, the Timer (inside Dialog) restarts auto-countdown: The user either clicks Save before time-up, or shall be Forced-Cancel when time-up. The lock is then released.
AddPrev/Next/Child shall pop-up AddRow-dialog (similar to EditRow-dialog, but the system shall generate new unique Row & RowID in the dialog): Save creates the new Row, Cancel does not.
Inside each AddPrev/AddNext/AddChild/EditRow-dialog: (Note: All the Hidden Cells and Hidden Items of said Row are also displayed in EditRow-dialog, they are also editable but marked as Hidden.)
The user Right-click any 'Field (but not touching any existing Item(s) in said Field)' to pop-up (per ACL) the: **Add Item** and **Format Cell** menu entries.
The user Right-click any 'existing Item in any Field' to pop-up (per ACL) the: **Add Item** and **Edit Item** and **Format Item** and **Delete Item** menu entries.
The user Drag-and-drop any 'existing Item in any Field' to 'Move' its position (i.e. sequential order) among 2-to-many Items contained in said Field.
Rule.#Section-Head: When a row has its Row-Type='Section-Head', said row is a Section-Header (i.e. Divider between sections of Rows in a TG), the system shall set the Row-Level=0 for said row.
Also, said row is not affected by the 'Sort Columns' function (i.e. the sort order of 'Section-Head Rows' never changes, because section dividers shall not move).
However, said row is affected by the 'Settings for Row-Level=0' in the corrsp Level-Set (when Activated). By default, the system displays all Section-Head Rows in **Bold**.

Format Local-Row: Format said Row (in the Current-TG) via FormatLocalRow-dialog (with 2 buttons: Save, Cancel). The fields (* means Mandatory) in said dialog are:
Row Status (⇔ t-Format.Status, :DDT of Statuses), Row Type (:DDT of Row-Types),
Row Audit-Trail* (⇔ t-Format.Audit-Trail, :Formula), Row Font-Style (⇔ t-Format.Font-Style, :Formula), Row Comment (⇔ t-Format.Comment).
This dialog shall acquire a single RowLock (via said row's t-Row.Row). If said Row is a 'Shared Row', this 'Format Local-Row' operation shall create/update a local format-setting (i.e. Format-Record) which dominates over said 'Shared Row's existing format-setting (i.e. Format-Record).

Rule#.LocalRow: When a Row is 'Local Row (i.e. t-Row.Share=NULL)', its corrsp Row-Format's t-Format.Status ∌ (does not contain) Share.

Format Share-Row: Format said 'Shared Row' (Menu entry is Dimmed when said Row is 'Local Row') via FormatShareRow-dialog (with 2 buttons: Save, Cancel).

This dialog shall acquire 2-or-more RowLocks and create/update the format-settings (i.e. corrsp Format-Record) for 'all Shared Rows with the same t-Row.Share'. The fields (* means Mandatory) are:

Row Status (⇔ t-Format.Status, :DDT of Statuses), Row Type (:DDT of Row-Types),

Row Audit-Trail* (⇔ t-Format.Audit-Trail, :Formula), Row Font-Style (⇔ t-Format.Font-Style, :Formula), Row Comment (⇔ t-Format.Comment),

Row Shared-TG's (Display a read-only list of TG's: Each TG contains 1-or-more Row's that are currently 'Sharing said row' or 'Inheriting said Row').

Rule.#Row-Format: For each Row, its corrsp Row-Format takes precedence (i.e. most relevant, which dominates) over its corrsp 'Cols/TG-Format(s)'.

In the FormatLocal-dialog, all fields with NULL data shall inherit (by default) from their corrsp 'Cols/TG-Format(s)', then, the user edits the fields before he Saves (which updates existing or creates) the corrsp Format-Record.

All the Row Audit-Trail-Cells in the system are System Cells (per Admin ACL), they are all automatically updated by the system.

A Row is 'Read-only Row' when its corrsp Row-Format's t-Format.Status ∋ Read-only, or, its 'containing TG's corrsp TG-Format's t-Format.Status ∋ Read-only.

A Row is 'Shared Row (e.g. t-Row.Share=11)' when its corrsp Row-Format's t-Format.Status ∋ Share: Said 'Share (Item)' may be deleted (per ACL) from said t-Format.Status 'only when' said Row is currently 'the only (undeleted) Row with t-Row.Share=11' in the system, i.e. other Rows with t-Row.Share=11 have been deleted.

A Row is 'Hidden Row' when its corrsp Row-Format's t-Format.Status ∋ Hidden: A Hidden Row is not displayed (in the TG containing it).

A Row is 'System Row' when its corrsp Row-Format's t-Format.Status ∋ System: A System Row is only Editable per Admin ACL.

View Row: The user can View said row (via ViewRow-dialog with 1 button: Close): Displays all the fields from 'EditRow-dialog and FormatLocalRow-dialog and FormatShareRow-dialog', all fields are read-only here. No Lock is required.

Open TG: If (and only if) said row's Value-Cell is a TG-ID, the system shall add a new TG-Tab for opening the said TG-ID (as TG). If said row's Value-Cell is not a TG-ID, this Menu entry is 'Dimmed'.

Select Rows: Select and Lock row(s) – The rows-select-action is very similar to how MS Windows File Explorer allows a user to select a mixture of Files (leaf rows) and Folders (non-leaf rows).

User clicks 'Select Rows': System shall Activate Tickbox (URL). When the user sets the first Checkbox=On, this action turns on the 'Row-Selection-Mode', the system shall set 'Timer ☑' Checkbox=On and displays the 'Acquired/Released/Waiting-for Lock(s) status'.

User sets any Checkbox=On to select '1-or-many row(s) in 1-or-many Row-Level(s)', and he continues to set/unset more Checkboxes. The system shall acquire RowLocks for all the selected rows.

User can wait for all selected rows to be locked, or, he can manually set 'Timer ☑' Checkbox=Off to Stop waiting for more locks. All rows he has Locked (and not yet Released) so far are highlighted (in the color defined in his User-Profile). Once all selected rows are Locked, or when the user Stops waiting: The Timer shall restart auto-countdown, the user has to click Copy/Move/Share/Del/FormatRows, or shall be Forced-cancel when the timer is up. Then, Row-Selection-Mode ends.

When Row-Selection-Mode ends, the system shall: Release all acquired RowLocks; Cancel (and clear) the Timer; Deactivate the Tickbox, and Turn-off color-highlight for selected rows.

Format Rows: 'Batch-Format' all currently Selected Rows (together) via (a single) pop-up FormatRow-dialog (with 2 buttons: Save, Cancel).

FormatRow-dialog lets users batch-edit the same way how 'Foobar2000 Media Player' lets users batch-edit the Properties of multiple tracks (together, in a single dialog).

The 'Format Rows' menu entry is 'Dimmed' by default, it is Only Enabled when the Row-Selection-Mode is currently On of the Current-TG.

Delete Rows: Delete the currently selected Row(s) via DelRows-dialog with 2 buttons: Delete, Cancel. ('Delete Rows' menu entry is 'Dimmed' by default, only Enabled when said TG's Row-Selection-Mode is On)

Rule.#Orphan-Row: If the system detects the user's current Move/DelRows action shall generate Orphan row(s), the system shall reject the user's current action with the pop-up Message: "Move/DelRow(s) is Rejected: Such action shall generate Orphan row(s)".

Copy→Next; Copy→Child: Copy the currently selected Rows as 'next siblings' or 'children' of the Target Row. ('Target Row' is the Row where Current-User triggered the 'Row Menu')

Move→Next; Move→Child: Move the currently Selected Rows as 'next sibling' or 'children' of the Target Row.

Share→Next; Share→Child: Share the currently selected Rows as 'next siblings' or 'children' of the Target Row. (See 'Rows' paragraph)

Rule.#New-Row: When the user 'Copy/Share row(s)', the system shall auto-generate new unique t-Row.Row(s) for all the (newly created) row(s).

For the TG-Tab(s) on the current webpage, the Copy/Move/Share menu entries are 'Dimmed' by default, Only Enabled in 1-or-many Target-TG's when: One of the TG-Tab (i.e. Source-TG-Tab) contains Source-TG which is currently in Row-Selection-Mode, also, the Target-TG (in 1-or-many targeted TG-Tab) has 'Sort Columns ☑' Checkbox=Off (on Menu-Bar) (because active Sorting interferes with the Copy/Move/Share results).

The user clicks one of the Copy/Move/Share menu entries: The system shall acquire RowLock for the Target Row (the user can Cancel waiting anytime), once the Target Row is locked, the system shall perform the Copy/Move/Share from Source-TG to Target-TG (same or different TG). Then, the system releases all the acquired locks. The result row(s) generated by Copy/Move/Share are highlighted (in the color defined in his User-Profile) - until the user performs the next action in Target-TG.

Format Cell – The user Right-click any 'Cell (but not touching any existing Item(s) in the Cell)' in a row (in Current-TG), to pop-up (per ACL) FormatCell-dialog (with 2 buttons: Save, Cancel).

The dialog shall acquire RowLock of the Current-Row, and the fields (* means Mandatory) in the said dialog are:

Cell Status (⇔ t-Format.Status), Cell Default-Data (⇔ t-Format.Default), Cell Calculated? (⇔ t-Format.Calculated), Cell Calc-Formula (⇔ t-Format.Calc-Formula),

Cell Data-Formula (⇔ t-Format.Data-Formula), Cell Font-Style (⇔ t-Format.Font-Style), Cell Comment (⇔ t-Format.Comment), Cell Audit-Trail (⇔ t-Format.Audit-Trail).

For each Cell, its corrsp Cell-Format takes precedence (i.e. most relevant, which dominates) over its corrsp 'Row/Col/TG-Formats'.

In the FormatCell-dialog, all fields with NULL data shall inherit (by default) from their corrsp 'Row/Col/TG-Formats' (in the order of relevance, if exists), the user edits the fields before he Saves (which updates existing or creates) the corrsp Format-Record.

A Cell is 'Calc Cell' when its corrsp Cell-Format's t-Format.Status ∋ Calculated, or, its corrsp Col-Format's t-Format.Status ∋ Calculated.

A Cell is 'Read-only Cell' when its corrsp Cell-Format's t-Format.Status ∋ Read-only, or, its corrsp 'Row/Col/TG-Format's t-Format.Status has Read-only.

A Cell is 'Hidden Cell' when its corrsp Cell-Format's t-Format.Status ∋ Hidden: A Hidden Cell is displayed as 'Blank' (as a placeholder) in the Row containing it.

A Cell is 'System Cell' when its corrsp Cell-Format's t-Format.Status ∋ System: A System Cell is only Editable per Admin ACL. (e.g. Each RowID-Cell is a System Cell)

Add Item – Pop-up (per ACL) Add-XXX-dialog (with 2 buttons: Save, Cancel), where XXX is the Data-Type of said Cell (containing the Item to be added)

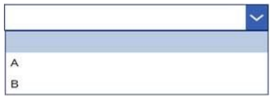
Edit Item – Pop-up (per ACL) Edit-XXX-dialog (with 2 buttons: Save, Cancel), where XXX is the Data-Type of said Item (to be edited).

Delete Item – Pop-up (per ACL) Del-Item-dialog (with 2 buttons: Delete, Cancel) which displays said Item (to be deleted) in the dialog.

Format Item – The user Right-click an existing 'Item' in any Cell, to pop-up (per ACL) FormatItem-dialog (with 2 buttons: Save, Cancel).
The dialog shall acquire RowLock of the Current-Row, and the fields (* means Mandatory) in the said dialog are:
Item Data-Range (⇔ t-Item.Data-Range), Item Calculated? (⇔ t-Format.Calculated), Item Calc-Formula (⇔ t-Format.Calc-Formula), Item Status (⇔ t-Format.Status),
Item Data-Formula (⇔ t-Format.Data-Formula), Item Font-Style (⇔ t-Format.Font-Style), Item Comment (⇔ t-Format.Comment), Item Audit-Trail (⇔ t-Format.Audit-Trail).
For each Item, its corrsp Item-Format takes precedence (i.e. most relevant, which dominates) over its corrsp 'Cell/Row/Col/TG-Formats'.
In the FormatItem-dialog, all fields with NULL data shall inherit (by default) from their corrsp 'Cell/Row/Col/TG-Formats' (in the order of relevance, if exists), user edits the fields before he Saves (which updates existing or creates) the corrsp Format-Record.
An Item is 'Calc Item' when its corrsp Item-Format's t-Format.Status ∋ Calculated, or, its corrsp 'Cell/Row/Col Format's t-Format.Status ∋ Calculated.
An Item is 'Read-only Item' when its corrsp Item-Format's t-Format.Status ∋ Read-only, or, its corrsp 'Cell/Row/Col/TG-Format's t-Format.Status ∋ Read-only.
An Item is 'Hidden Item' when its corrsp Item-Format's t-Format.Status ∋ Hidden: A Hidden Item is not displayed (in the Cell containing it).
An Item is 'System Item' when its corrsp Item-Format's t-Format.Status ∋ System: A System Item is only Editable per Admin ACL.
Data Ranges are: >, ≥, <, ≤, ! (Not), Est (estimate), Avg (average), Med (median). Ex: A Cell has 2 Items: '>, 2000-1-1' & '≤, 2000-1-9', meaning 'A range of 8-days'. (Samples in 'User-Profile' Sheet)

PostgreSQL Database Design

For the definition of all mandatory TG's and PostgreSQL DB Tables: Please refer to the 'TG & DB' Sheet in the latest 'AIC Design' Excel doc.
In the 'TG & DB' Sheet of the latest 'AIC Design' Excel doc:
The 1st table: Defines the 'Initial-Columns' in each TG, the system automatically added these Columns to each newly created TG (of any TG-Type).
The 9th table: Defines the Additional Columns (in addition to the Initial-Columns) in each Product-TG, please refer to the 'Product-TG' sheet in AIC Design doc for a real-life sample of Product-TG.
In the SQL-Table 't-Format', each Format-Record (i.e. t-Format record) means:
When t-Format.TG≠NULL: Said TG-Format is formatting said TG (with t-Format.TG = t.TG.TG)
When t-Format.Col≠NULL: Said Col-Format is formatting said Col (with t-Format.Col = t-Col.Col).
When t-Format.Row≠NULL: Said Row-Format is formatting said Row (with t-Format.Row = t-Row.RowID).
1. If (t-Format.Row = t-Row.RowID = t-Row.Row): Said Row-Format is formatting 1 'Local Row'.
2. If (t-Format.Row = t-Row.RowID = t-Row.Share): Said Row-Format is formatting 2-to-more 'Shared Rows' (with the matching t-Row.Share).
When a Row has 'conflicting Row-Format settings', the Row-Format of t-Row.Row (i.e. Case 1) takes precedence (i.e. more relevant, which dominates) over the Row-Format of t-Row.Share (i.e. Case 2).
When t-Format.Cell≠NULL: Said Cell-Format is formatting said Cell (with t-Format.Cell = t-Cell.Cell).
When t-Format.Item≠NULL: Said Item-Format is formatting said Item (with t-Format.Item = t-Item.Item).
Rule.#FormatConflict: System shall resolve all the 'format setting conflict(s)' by referring to the following 'order of dominance (i.e. relevance)': Item-Format > Cell-Format > Row-Format > Col-Format > TG-Format.
For the definitions of PostgreSQL Database Tables, we use the term 'SQL-Column (i.e. Column of a SQL DB Table)' to avoid confusion with the word 'Column (of a TG)', we use the term 'SQL-DataType (i.e. DataType of a SQL DB Table Column)' to avoid confusion with the term 'Data-Type (of a TG's Column)'.
Some SQL-DataType's are in the color **Blue**: These are the foreign-keys (of the same database) represented by the format 'SQL-Table (i.e. Database Table) **.** SQL-Column (i.e. Primary-Key of said SQL-Table)'
When a Row (with its Row-Level=0 or 1) has no Parent Row (in the TG containing it), the system shall set its t-Row.Parent-Row=NULL.
When a Row has no 'Previous Sibling-Row' (in the TG containing it), the system shall set its t-Row.Sibling-Row=NULL.
In All-Unit, each Unit (row) has a corrsp Std-Unit (row): Std-Unit is the 'Previous Sibling-Row' of said Unit (i.e. Unit and Std-Unit have the same Parent Row in All-Unit), but Std-Unit has its Row-Type=StdUnit.
Ruile.#EditItem: Each of the below Editor/Selector-dialog's shall have 2 buttons (when the user Edits each Field, the system shall instantly validate its corrsp t-Format.Status and rejects said Edit with an error msg if said validation failed): Save (system saves all Fields-Data in the current dialog), Cancel (discard the current edit).
When the user clicks Add Item, the system creates a new Item, and fills said Item with the applicable Default Data (if exists), as follows:
1. The Cell containing said Item: Said Cell's corrsp t-Format.Default (if exists) shall apply, otherwise,
2. The Col corresponds to said Cell: Said Col's corrsp t-Format.Default (if exists) shall apply.
To edit an Item with Data-Type=Amount: The system shall pop-up 'Edit-Amount-dialog with 4 fields' to let the user edit t-Item.Qty-Data (:Num) and select t-Item.Unit-Data (:DDT), then, the system shall auto-calculate the t-Item.Std-Unit-Data and t-Item.Std-Qty-Data, with this formula: t-Item.Std-Qty-Data = t-Item.Qty-Data **x** (Unit.Factor of t-Item.Unit-Data) **÷** (Unit.Factor of t-Item.Std-Unit-Data).
Each Item (i.e. t-Item record) with Data-Type=ML-Text (i.e. Multilingual Rich-Text) shall store 1-or-more pair(s) of key-value(s) in said t-Item.JSON-Data. For example:
e.g. The 1st pair of key-value: {(RowID of) English : "(Language-specific) Rich-text in English"}, 2nd pair: {(RowID of) French : "(Language-specific) Rich-text in French"}, etc.
To edit an Item with Data-Type=ML-Text: System shall pop-up 'Edit-ML-Text-dialog (with a 2xN Table)' to let the user edit the corrsp (0-or-more 'key-value pairs' as) t-Item.JSON-Data.
To edit an Item with Data-Type=URL: System shall pop-up 'Edit-URL-dialog (i.e. a mini browser window)' to let the user key in a URL to display the desired webpage inside the dialog.
Once the webpage showed up correctly in the dialog, the user clicks Save and the system shall immediately make a backup of said webpage by downloading it as a single HTML-file (similar to this Tool) (unless said webpage backup file already exists) on the server. Each Item (i.e. t-Item record) with Data-Type=URL stores said data in t-item.JSON-Data, with 2 pair of key-values:
The 1st pair of key-value: "Original URL" : "*Original URL provided by the user*", The 2nd pair: "Backup URL" : "*The internal URL referring to the webpage's backup HTML file on server*".
To edit an Item with Data-Type=Text: System shall pop-up 'Edit-Text-dialog' to let the user edit a single pair of {0 : "(Language-neutral) Rich-Text"} as t-Item.JSON-Data.
To edit an Item with Data-Type=Color: System shall pop-up 'Edit-Color-dialog' to let the user select the corrsp (Color as) t-Item.Color-Data.
To edit an Item with Data-Type=DDT: System shall pop-up 'Edit-DDT-dialog' to let the user select corrsp (DDT-Item as) t-Item.Row-Data. User can always select NULL in said dialog, e.g.



To edit an Item with Data-Type=Category-ID: System pop-up 'Edit-Category-ID-dialog' to let the user select the corrsp (Category-ID as) t-Item.BigInt-Data.
To edit an Item with Data-Type=Product-ID: System pop-up 'Edit-Product-ID-dialog' to let the user select the corrsp (Product-ID as) t-Item.BigInt-Data.
To edit an Item with Data-Type=Search-Set-ID: System pop-up 'Edit-Search-ID-dialog' to let the user select the corrsp (Search-Set-ID as) t-Item.BigInt-Data.
To edit an Item with Data-Type=Level-Set-ID: System pop-up 'Edit-Level-Set-ID-dialog' to let the user select the corrsp (Level-Set-ID as) t-Item.Bignt-Data.
To edit an Item with Data-Type=Unit: System shall pop-up 'Edit-Unit-dialog' to let the user select the corrsp t-Item.Unit-Data.
e.g. A t-Item record has its Data-Type=Amount, the system shall pop-up 'Amount Editor-dialog with 4 fields' to let the user Edit: Qty=5 and select Unit=Kg, then, the system auto-calculates Std-Qty=5000 and Std-Unit=g. Then, the system displays said Item in ML-Text format {(RowID of) EN : "50 Kg"}.

Master-TG

(Please refer to the 'Master-TG' Sheet of the latest 'AIC Design' Excel doc.)

Master-TG (with its TG-ID=1) is a System-TG that lists (and manages) all existing TG's: Each TG-Row (contained in Master-TG) represents a unique TG in the system.

Rule.#Admin-ACL: The Admin ACL may CRUD all TG's (including System-TG's), it has no access limits in the system.

Rule.#TG-Name: Each TG-Row (in Master-TG) shall have a 'unique TG-Name', the system shall reject any 'new/updated TG-Name which is not unique' with an error message.

TG-Version-List (of a TG, e.g. ABC): A subset of TG-Rows (of Master-TG), with the 'said parent TG (ABC) as the first TG-Row in said list' followed by 'all its child TG-Row(s), each representing a Version-TG', each (Non-Private) Version-TG in the said list has a different 'TG-Version (:DDT of Versions)'.

Each Category-TG (represented by a TG-Row in All-Category) has 0-or-many (TG-Rows representing its) 'Descendant Category-TG(s)'.

Category-TG: A TG (represented by the TG-Row) in Master-TG with TG-Type=Category. Product-TG: A TG (represented by the TG-Row) in Master-TG with TG-Type=Product.

Profiles-TG: A System-TG which contains all the existing User-Profiles, each User-Profile belongs to a (Registered) User in the system.

(Registered) User: Each Row (in Profiles-TG) with Row-Type=User represents a User, said Row is also contained in 'All-User (a subset of Profiles-TG)'.

User-Profile (of a User): A 'User Profile Section' of the rows (contained in Profiles-TG) which contains (said User represented by) 1 Row with its Row-Type=User (in said Section).

User-Profile

(Please refer to the 'User-Profile' Sheet of the latest 'AIC Design' Excel doc. Here is a sample User-Profile (i.e. a subset of Profiles-TG) of the User 'Test123'.)

Rule.#Multi-Item: When a Cell contains 2-or-more Items, the first Item (in said Cell) shall be the 'Dominant Item' of said Cell.

e.g. When Display Languages(s)-Value = 'EN; FR': Meaning this user can read both EN & FR, but his default language is EN. So, system shall auto-translate all content 'in language other than EN or FR' into EN for him.

Rows

Each 'Shared Row' in the system: Its t-Row.Share≠NULL, and, its corrsp t-Format.Status ∋ Share.

Rule.#ShareRow: Each Row may be 'Shared (via Share Next/Child menu entry)' in '2-or-many locations' contained in '1-or-many TG(s) (with 1-to-many identical Col-ID's and 0-to-many different Col-ID's)'.

When the 'ShareNext/Child' menu entry is creating a 'Target Shared Row' from the said 'Source Row':

If the 'Source Row' has its t-Row.Share≠NULL (i.e. said 'Source Row' is already a 'Shared Row'), the newly created 'Target Shared Row' shall copy t-Row.Share from the 'Source Shared Row', and, all the Data-Items in 'both the Source/Target Shared Rows' shall have their corrsp (Data-Items') t-Format.Status ∋ Share.

If the 'Source Row' has its t-Row.Share=NULL (i.e. said 'Source Row' is a 'Local Row'), the system shall convert said 'Source Local Row' to 'Source Shared Row': After the 'Target Shared Row' is created, the system shall 'generate a new' t-RowID.Row and assign it (i.e. same data) to both 'Source Shared Row's t-Row.Share and 'Target Shared Row's t-Row.Share.

e.g. Rows with RowID:11 may show up in 5 locations among 3 TG's, and in each location: Said 'Shared Row' has different parent/sibling/child rows, and, said Row can be deleted (moved to RB) and restored by itself.

Each of the 5 rows with RowID:11 is treated equally by the system, none of these 5 rows is considered 'original row or replicated row', the system makes sure that they are always 100% in sync.

When the user edits any of the 5 rows (with RowID:11), the system has to first acquire RowLocks for all 5 rows (together), then after the edit, sync the update to all 5 rows, then release the RowLocks (together).

Rule.#Inherit: Each 'Category/Product-TG (e.g. TG-X)' shall automatically Inherit 'ALL the Visible Rows' contained in '1-to-many Category-TG(s) (e.g. TG-Y)' found in TG-X's 'Parent Category(s)-Value'.

Each Inherited Row's corrsp t-Format.Status ∋ Inherit and t-Row.Inherit≠NULL, also, said Row cannot be 'Deleted' (i.e. said 'Delete Row' menu entry is Dimmed), but it can be 'Hidden' (via 'Format Local-Row').

Each Inherited Item's corrsp t-Format.Status ∋ Inherit and t-Item.Inherit≠NULL, also, said Item cannot be 'Delete or Edit' (i.e. 'Edit/Del Item' menu entry is Dimmed), but said Item can be 'Hidden' (via 'Format Item').

(Refer to 'AIC Design' doc for the Notes on t-Row.Inherit and t-Item.Inherit. The system shall access these 2 columns when executing the 'Sync Descendent' operation)

When (the above) Row-Y (i.e. Any Row that is currently being Inherited) was Edited in TG-Y (by User or by System), the system shall 'Sync Descendant' said update to all the 'directly and indirectly Inherited Row-X(s) contained in said TG-Y's 0-to-many 'Descendant Category/Product-TG(s) (based on each descendant Category/Product-TG's Parent Category(s)-Value)', also, each said Row-X has the same Label-Cell as Row-Y does.

But said 'Sync Descendant (system shall acquire RowLocks)' operation shall not update/delete any 'User added Local Item(s) (with corrsp t-Format.Status ∌ Inherit)' in any of said 'Inherited Row-X(s)'.

Users can CopyNext/Child rows within TG A or from TG A to TG B (in 2 TG-Tabs), the Copy action requires RowLock on the 'Target Row'.

Copy rows – The system shall generate new unique Row-Cell(s) because newly created Copied Rows shall have new unique Row-Cell(s).

Users can MoveNext/Child rows within TG A or from TG A to TG B (in 2 TG-Tabs), the Move action requires RowLock's on both the 'Source Row(s) (to be moved)' and the 'Target Row'.

Move rows – The system does not generate new Row-Cell(s), because no new rows are created, they are being relocated.

Users can ShareNext/Child rows within TG A or from TG A to TG B (in 2 TG-Tabs), the Share action requires RowLock on the 'Target Row'.

The Row-Header of each Row: The area 'to the left of first/left-most Cell' of each Row, said area has a background color of 'Row-Header Background Color' (as shown in the 'Product-TG' sheet).

When user 'Single-click' the Row-Header of each Row, system shall switch said Row between the Full-Length-Mode (displays each Cell/Item in full-length with no truncation, said Row in full-length shall overwrite the Row Height Max-Line(s) setting in User-Profile), and, Partial-Length-Mode (Default) (each Cell/Item in said Row maybe truncated. Please refer to the Row:304 in 'User-Profile' Sheet in the latest Excel doc).

Rule.#Tooltips: When the user 'Mouse-hover' over any of the following objects, the system shall pop-up a corrsp Tooltip of said object.

Mouse-hover over a Column-Header in a TG, or, a Field-Label (i.e. Column-Name) in a Dialog: Tooltip content = said Column's corrsp t-Format.Comment (if exists)

Mouse-hover over the Row-Header in a Row: Tooltip content = said Row's corrsp t-Format.Comment (if exists)

Mouse-hover over the Label-Cell (if exists) in a Row: Tooltip content = corrsp t-Format.Comment (if exists) of 'the matching Row in All-Label (TG), i.e. Said Row (in All-Label) has the matching Label-Cell'.

Mouse-hover over a Cell in a Row (but not touching any Item inside said Cell): Tooltip content = said Cell's corrsp t-Format.Comment (if exists)

Mouse-hover over a Field (i.e. Cell) in a Dialog (but not touching any Item inside said Field): Tooltip content = said Cell's corrsp t-Format.Comment (if exists)

Mouse-hover over an Item (if exists) (inside a Cell or Field): Tooltip content = Display said Data-Item in full-length (i.e. no truncation), and, said Item's corrsp t-Format.Comment (if exists)

Mouse-hover over the '+N' (if exists) (inside a Cell or Field): Tooltip content = Display all the hidden 'N Data-Item(s)' in full-length (i.e. no truncation), and, these Item(s)' corrsp t-Format.Comment(s) (if exists)

Rule.#RB: Each Row (contained in any TG) deleted by the 'Delete Rows' menu entry is moved to Recycle Bin (RB). All Deleted Rows (in RB) are locked/unlocked the same way as Undeleted Rows.

Each Deleted Row in RB can be Restored to any TG (i.e. Target-TG), the user shall perform the ‘SelectRows + MoveNext/Child’ menu entry to ‘Move’ the Deleted Rows to Target-TG: Only if each of the Deleted Rows shall match (i.e. Data-type compatible) the existing Columns in Target-TG. Otherwise the system prompts an error msg: “Error: Failed to restore Row(s) due to incompatible Column(s): xxx-Col, yyy-Col, etc”.
If parent-child rows are deleted together, they will remain parent-child rows inside RB. Even when parent-child rows were deleted separately (i.e. in 2-or-more delete actions), they remain parent-child rows inside RB. The Parent-child rows in RB can be restored (via ‘SelectRows + MoveNext/Child’ menu entry) individually (i.e. restore 1 row) or restored together (i.e. restore 2-or-many rows).

***** Milestone 2 & 3 *****

ACL Rank

(Please refer to the ‘ACL Rank’ Sheet of the latest ‘AIC Design’ Excel doc.)

Each newly registered user is assigned ‘Formatter’ ACL-Rank by default, said ACL-Rank is saved as System-Row(s) in his User-Profile.
The ACL-Rank of a registered user may be changed by Admin or by another user of ‘Higher ACL-Rank’ (than himself).
Each anonymous Viewer, or, a User currently not logged-in: He shall be assigned ‘Viewer’ ACL-Rank for the current Client.
Viewer: Cannot update TG’s, Cannot update Format-Record’s.
Formatter: Cannot update TG’s, Can update Format-Record’s.
Editor: Can CRUD all Regular-TG’s - with default Lock-Period.
Publisher: Can CRUD all Regular-TG’s - with extended Lock-Period.
Admin: Can do anything, e.g. CRUD all TG’s, Unlimited Lock-Period, CRUD all rows in the Recycle Bin, etc.

Category-TG:

(Please refer to the ‘Category-TG’ Sheet of the latest ‘AIC Design’ Excel doc.)

Following the above (1st-TG) example in the ‘All-Cat-Tab paragraph’, the ‘Fish (Category-TG)’ has 3 Parent Category’s (according to its Parent Category(s)-Value): Seafood, Aquatic, Pet. Which triggers the system to inherit ‘all the current Visible Rows contained in these 3 Parent Category’s as the ‘Inherited Rows’ in said ‘Fish (Category-TG)’.
In a Product-TG, if the user adds an Item=‘Fish (Category)’ to its Add-Category(s)-Value, which triggers the system to also add said ‘Item=Fish (Category)’ to its Parent Category(s)-Value: Means the user just added ‘Fish (Category)’ as one of the Parent Category(s) of said Product-TG, which triggers the system to also inherit ‘all the Visible Rows contained in said Fish (Category-TG)’ as the ‘Inherited Rows’ in said Product-TG.
As follows:
RowID:1002/1003, meaning: The ‘Fish (Category)’ has 2 (Calculated) Parent Category(s): The system automatically adds ‘(calculated) Item(s)’ to the Parent Category(s)-Value of ‘Fish (Category-TG)’ - by calculating each ‘Search-Formula-Value (:Formula)’ in all the Category-Search-Set’s that are listed in All-Category-Search-Set (TG). The ‘Fish (Category)’ also has 1 ‘user-added Parent Category’: User can Edit (per ACL) the Add-Category(s)-Value to add 0-or-many ‘(Category) Item(s)’ to the Parent Category(s)-Value. If any two ‘Category Items’ in said Parent Category(s)-Value have an Ancestor-Descendant relationship as listed in All-Category (TG), said ‘Descendant Category’ shall take precedence (over the other ‘Ancestor Category’, which shall be displayed as a ‘Dimmed Item’) in being the effecting Parent Category (of Fish).
RowID:1004 has Row-Type=Name, meaning: The Value-Cell (i.e. Fish) of said Row shall always be the ‘TG-Name (in Master-TG)’ of said (Category) TG.

Locks

Add/Edit/Del Col – acquires a ColLock. Add/Edit Row – acquires a RowLock. Select Rows – acquires RowLock(s) on all the user-selected row(s).
Copy/Move/Share Rows – acquires a RowLock on the ‘Target Row’ (in addition to RowLocks already obtained by ‘Select Rows’).

Rule.#Lock:
In Current-TG, a user-occupied RowLock excludes all other users acquiring the same RowLock – they shall wait in the queue.
In Current-TG, a user-occupied RowLock excludes all other users acquiring any ColLock – they shall wait in the queue.
In Current-TG, a user-occupied ColLock excludes all other users acquiring any ColLock or any RowLock – they shall wait in the queue.

Rule:#Timer: The system controls all Timers on all the clients.
If server lost connection with any client Timer longer than ‘Threshold’: Server shall Force-cancel said client Timer, and Force-cancel user’s current Action (which triggered said Timer), and release all his acquired Locks.
e.g. UserA's EditRow-dialog Timer displays: "Now waiting for locks: User1.Tab3 (12s, EditRow) + User2.Tab4 (30s, EditCol) + User3.Tab5 (30s, SelectRows)” also "User51.Tab6 (EditRow), User53.Tab7 (SelectRows) are waiting behind you". Meaning User1 is currently holding this lock, he has 12sec remains to Save or Cancel, or shall be Forced-cancel when time-up. Of course, User1/2/3 each may release his lock sooner. The server reflects the current locks’ status by continuously updating UserA’s Timer. After User1 releases the lock, User2 and User3 will in turn receive the lock, then the lock is passed in turn to UserA, User51, User53, etc.
If User A obtains the RowLock before Timer runs out, his Timer shall display: “Row Locked for EditRow, 30 sec remains.”, then, his Timer restart auto-countdown.

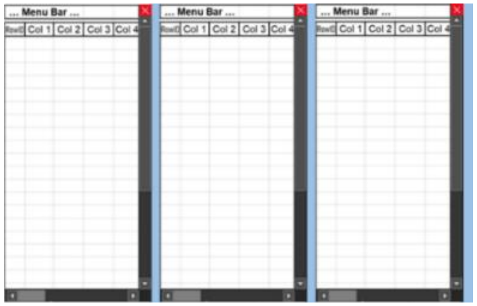
When UserB clicks the ‘Select Rows’ menu entry to enter ‘Row-Selection-Mode’, the system Activates Tickbox, also, the system (Activates) sets ‘Timer ☑’ Checkbox=On (on Menu-Bar).
UserB set ‘Timer ☑’ Checkbox=Off (on Menu-Bar) to Cancel the current Timer, which Ends the Row-Selection-Mode, system shall release all the RowLocks he has obtained so far.
e.g. UserB uses the Tickbox to select 5 rows and his Timer displays: “Now locked 3 rows (highlighted in the color defined in his User-Profile); Waiting for these locks: RowID:156 (Row-Key:20189) for 42s, RowID:169 (Row-Key:9856) for 26s” and "User61, User67, User69 are waiting behind you".
The 3 rows that UserB has acquired: UserB holds Each RowLock for the length of ‘Lock-Period’ before it is Force-released by the server, then, its color-highlight is turned-off.
When all the acquired RowLocks have been Force-released by the server, the current ‘Row-Selection-Mode’ shall End.

Whenever the user clicks ‘Copy/Move/Share/Del Rows’ in ‘Row-Selection-Mode’, the system shall:

1. Stop Waiting for RowLocks (whatever UserB is currently waiting on), only keep the locks that the user has acquired (and not yet Force-released by the server due to Lock-Period) so far.
2. Performs 'Copy/Move/Share/Del Rows' on all the selected-and-locked rows, and the result rows are highlighted (in the color defined in his User-Profile).
3. Release all acquired locks. Ends 'Row-Selection-Mode'.

All-Cat-Tab

User clicks All Cat (on Menu-Bar) to add a new All-Cat-Tab for opening 1-or-many TG's, as shown in the following picture:
Users can horizontally-resize each TG by dragging the **blue** divider-bar on the right-side of said TG.



1. All-Category
By default, system shall open All-Category as the Left-most-TG (i.e. 1st TG-Main) in All-Cat-Tab.
In the following example, the (subset sample of) 'All-Category (as TG)' contains 1-or-more Shared Row(s) (e.g. Fish):
(Please refer to the 'All-Category' Sheet of the latest 'AIC Design' Excel doc.)

2. All-Product
When user Clicks any 'Read-only (Calc) Cell' in the 1st TG-Main: System shall open 'a Subset of All-Product (as TG)' as 2nd TG-Main (i.e. Second-left-most-TG in All-Cat-Tab) which contains all (TG-Rows representing) the Product-TG(s) calculated by said Cell.
ex. User clicks the Cell '15' (of Pet Category): System opens the 2nd TG-Main with 15 TG-Rows, each TG-Row represents a 'Category-Product (i.e. Product-TG with its Parent as a Descendant-Category)' of Pet. Among these 15 Products: 6 are Pet Fish, 5 are Pet Dog, and 4 are General Pet (not Fish or Dog, maybe Horse, Pig, etc).

3. Product-TG
Users can Open (via 'Open TG') 0-or-many TG-Row(s) in 2nd TG-Main: The system shall open each Product-TG (represented by said TG-Row in 2nd TG-Main) as the 3rd TG-Main/4th TG-Main/5th TG-Main, ..etc.

Product-Comparison

An 'All-Cat-Tab' contains 0-to-many Product-TG(s) as the (from left to right in All-Cat-Tab) 3rd/4th/5th-TG..., etc.
For each Product-TG in All-Cat-Tab, the user can set its 'Compare ☑' Checkbox=On (on its Menu-Bar): System shall switch said Product-TG to 'Product-Comparison-Mode'.
The system shall (in real-time) compare 2-or-many Product-TGs (in All-Cat-Tab) that are currently in 'Product-Comparison-Mode'.
Product comparison is performed only on 'all Visible Rows' contained in each Product-TG of the Product-Comparison-Mode' (e.g. Product-A vs Product-B), as follows:
If Product-A contains a row (e.g. Row-X) that has Label-Cell=Y but Product-B does not contain a row with Label-Cell=Y, the system shall add a temporary Virtual Row (with a new unique RowID, but Label-Cell=NULL and Value-Cell=NULL) to Product-B. The system fills all the missing rows (mentioned above) with Virtual Rows to make sure all products in Product-Comparison-Mode have 1-to-1 row-comparison with each other. Also, this is a good moment for the user to edit (fill the missing Cell's in) Virtual Row in Product-B while referring (side-by-side) to Row-X in Product-A. Virtual Rows that are edited (by user) shall be instantly created in Product-B, and Virtual Rows that are not edited (by user) shall be deleted when leaving the (current) Product-Comparison-Mode.
Rule.#Row-Compare: When Comparison-Pref='Prefer High': Value-Cell higher is better, if Value-Cell is NULL, Label-Cell-Item# higher is better. When Comparison-Pref='Prefer Low': Value-Cell lower is better, if Value-Cell is NULL, Label-Cell-Item# lower is better.
If RowX in Product-A ranked highest among all (RowX's of) Product-TGs currently in Product-Comparison-Mode, the system shall highlight RowX (in the defined color his User-Profile) in Product-A.
If RowX in Product-A ranked lowest among all (RowX's of) Product-TGs currently in Product-Comparison-Mode, the system shall highlight RowX (in the defined color his User-Profile) in Product-A.

Search-Set

Here is a sample 'Search-Set (TG)', it searches (and sets the Parent Category(s)-Value) for all the Products that belong to the 'Seafood (Category)':
(Please refer to the 'Search-Set' Sheet of the latest 'AIC Design' Excel doc.)

When Category-Value≠NULL, means:
This is a 'Catalog-Search-Set' which searches all the matching Products in All-Product that shall belong to the said 'Category-Value (e.g. SeaFood)' Category (:Category-ID).
The system shall add a '(Calculated) Item = said Category-ID (e.g. SeaFood)' to the Parent Category(s)-Value of each matching Product that was searched by said Category-Search-Set.
When Parent Category(s)-Value=NULL, means:
This is a 'Product-Search-Set' which searches all the matching Products in All-Product, but shall not change the Parent Category(s)-Value of the matching Products.
When searching for each Product (in All-Product), the Param-Label-Cell and Param-Value-Cell (of said Search-Set) shall match the Label-Cell and Value-Cell (of all rows) contained in said Product-TG.

The DDT-Item's of Param Opr (:DDT of Param Oprs) are: '∋ (Contains)', '∌ (Does not contain)', >, ≥, <, ≤, =, ≠, etc.
The system shall execute the 'Search-Formula-Value (:Formula)' of said Category-Search-Set against EACH Product listed in All-Product, to determine the 'Parent Category(s)-Value' of said Product-TG shall 'Add/Remove/Unchange' the '(Calculated) Item = said Category-ID (e.g. Seafood)'. If said execution resulted in 'True (i.e. Matched)', the system shall add said Item (or Unchanged, if previously added). If False, the Item shall Not be added (i.e. if the Item was previously added, the system shall remove it from said Parent Category(s)-Value).

Show-Set

Here is an example of (partial) All-DDT with ShowSet-Toggles currently Activated. The said ShowSet-Toggle feature is similar to but more complex than the Tickbox.
(Please refer to the 'Show-Set' Sheet of the latest 'AIC Design' Excel doc.)

User clicks each ShowSet-Toggle to set its 'ShowSet-Toggle-Cell' via pull-down DDT (of the following single-character DDT-Item's):

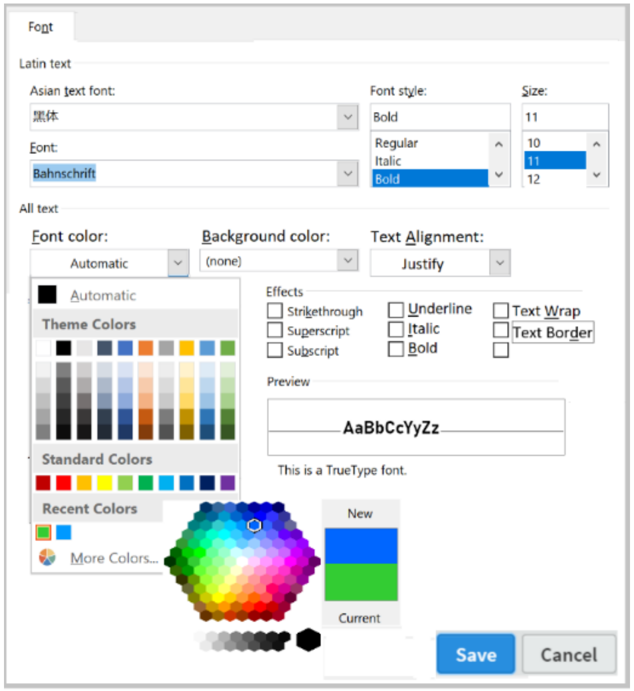
ShowSet-Toggle □	: NULL	The current Row is Not included in the Show-Set.
ShowSet-Toggle X	: R	The current Row (i.e. R) is included in the Show-Set.
ShowSet-Toggle c	: c	All Current Children of R (exclude R)
ShowSet-Toggle C	: C	All Current and Future Children of R (exclude R)
ShowSet-Toggle f	: c + R	Current Family
ShowSet-Toggle F	: C + R	Current and Future Family
ShowSet-Toggle d	: d	All Current Descendant of R (exclude R)
ShowSet-Toggle D	: D	All Current and Future Descendants of R (exclude R)
ShowSet-Toggle —	: —	Partial Current and/or Future Descendant of R (exclude R)
ShowSet-Toggle v	: d + R	Current Family Tree
ShowSet-Toggle V	: D + R	Current and Future Family Tree (This is the Default Data of all ShowSet-Toggle-Cell's)
ShowSet-Toggle ℓ	: ℓ	All Current Leaf-Node Descendant of R (include R)
ShowSet-Toggle L	: L	All Current and Future Leaf-Node Descendant of R (include R)
ShowSet-Toggle n	: n	All Current Non-Leaf-Node Descendant of R (include R)
ShowSet-Toggle N	: N	All Current and Future Non-Leaf-Node Descendant of R (include R)

Level-Set

Here is an example of Level-Set (as TG, opened in a TG-Tab):
(Please refer to the 'Level-Set' Sheet of the latest 'AIC Design' Excel doc.)

e.g. When the above Level-Set is Activated, all the Rows with Row-Level=3 in Current-TG shall be formatted by the RowID:203, as follows:
The 'BGColor: Gray' (Level-FontStyle: Formula) takes precedence over Cols-Format (but does not precede over Rows-Format), 'Sorted-by (See: URL)' in the order of Col-9:Descend followed by Col-8:Acsend (Level-SortBy='Col-9 ↓ , Col-8 ↑ '), 'Grouped-by (See: URL)' Col-9 followed by Col-8 (Level-GroupBy= 'Col-9, Col-8'), 'Filtered' by the Filter-formula (Level-Filter: Text), Hierarchy-Cell's (with Row-Level=3) are Allowed to be selected as DDT-Item's (Level-DDT=Yes). Also, RowID:203 has its Checkbox=On: Meaning the 'Level-Set' setting for Row-Level=3 is 'Activated' (vs. Deactivated if Checkbox=Off) on Current-TG.
Filter-formula: A Formula consists of these parameters: Column(s), Regular expressions, If-then-else, Math functions, etc.

FontStyle-Dialog



Latin text

Asian text font: 黑体

Font: Bahnschrift

Font style: Bold

Size: 11

All text

Font color: Automatic

Background color: (none)

Text Alignment: Justify

Effects

□ Strikethrough □ Underline □ Text Wrap

□ Superscript □ Italic □ Text Border

□ Subscript □ Bold □

Preview

AaBbCcYyZz

This is a TrueType font.