ExxonMobil Chemical Company

Database Design Specification

Version 1.21

August 30, 2019

Prepared by:

ILS Automation Inc.

REVISION HISTORY

Version	Date	Author	Description
1.0	5/27/2014	PH	Initial draft
1.1	10/28/2014	PH	Updated recipe tables
1.2	11/10/2014	PH	Final check for recipe tables
1.3	1/13/2015	PH	Added Diagnostic Toolkit tables
1.4	2/20/2015	PH	Updated DtQuantOutput, removed DtConsoleSubscription, Updated unique constraints. Added unit tables
1.5	4/20/2015	PH	Revised core post, unit, and console tables.
1.6	5/11/015	PH	Added Delivery section
1.7	6/3/2015	PH	Added new CheckpointTimestamp to QueueMaster table, added ER diagram for common tables, documented TkWriteLocation table.
1.8	12/13/2015	PH	Added DtSQCDiagnosis table.
1.9	2/23/2016	PH	Added SfcRecipeDataKey tables. Moved UUID and DiagramUUID from the DtDiagnosisEntry table up to the DtFinalDiagnosis table.
1.10	3/11/2016	PH	Added ConsoleName to the TkConsole table. Updated documentation for the common tables. There were no changes to the table design. Added grade to the LtHistory table.
1.11	4/5/2016	PH	Modified Lab Data tables to record the timestamp that a user viewed a lab value rather than the id of the value that was there at the time. Updated lab data ER diagram and table descriptions. Added TkLogbook tables
1.12	4/19/2016	PH	Added column Constant to table DtFinalDiagnosis.
1.13	6/2/2016	PH	Added IgnoreMinimumIncrement to the DtQuantOutput.
1.14	8/8/2016	PH	Added SpecialPostProcessingCallback to table DtFinalDiagnosis and changed TimeOfMostRecentRecommendationImplementation to Not Null with a default of the current time.

Version	Date	Author	Description
1.15	8/22/2016	PH	In DtFinalDiagnosis renamed column SpecialPostProcessingCallback to PostProcessingCallback and dropped column TextRecommendationCallback. Added table DtTextRecommendation. Cleaned up redundancy between DtFinalDiagnosis (deleted Multiplier), DtDiagnosisEntry (renamed RecommendationMultiplier to Multiplier, deleted ManualMove and ManualMoveValue), and DtRecommendation.
1.16	9/9/2016	PH	Added messaging tables: TkMessageRequest, TkMessageReply
1.17	2/26/2017	PH	Updated ER Diagram in section 5.1
1.18	7/20/2017	PH	Added table TkSite, Moved the delivery section to the Migration manual to avoid overlap.
1.19	8/9/2017	PH	Updated DtApplication and the recipe toolkit to store the value type (RtValueDefinition and RtValueType).
1.20	1/25/2018	PH	Added table LtDisplayTableDetails to allow a lab value to appear in multiple display tables.
1.21	8/30/2019	PH	Added SFC tables

Table Of Contents

1 Introduction

The new Ignition-based toolkits are tightly integrated with a database. A common design pattern is that where the previous platform defined a class definition the new platform will define a table. Instances of the class correspond to records in the table.

This document describes the design of the common database used by the toolkits at all of the sites. This is commonly referred to as the XOM database.

The sites also have the freedom to design site-specific tables that will be contained in a site specific database. An example of this is the UIR functionality which varies greatly from site to site, therefore each site will define the database requirements that meet their needs.

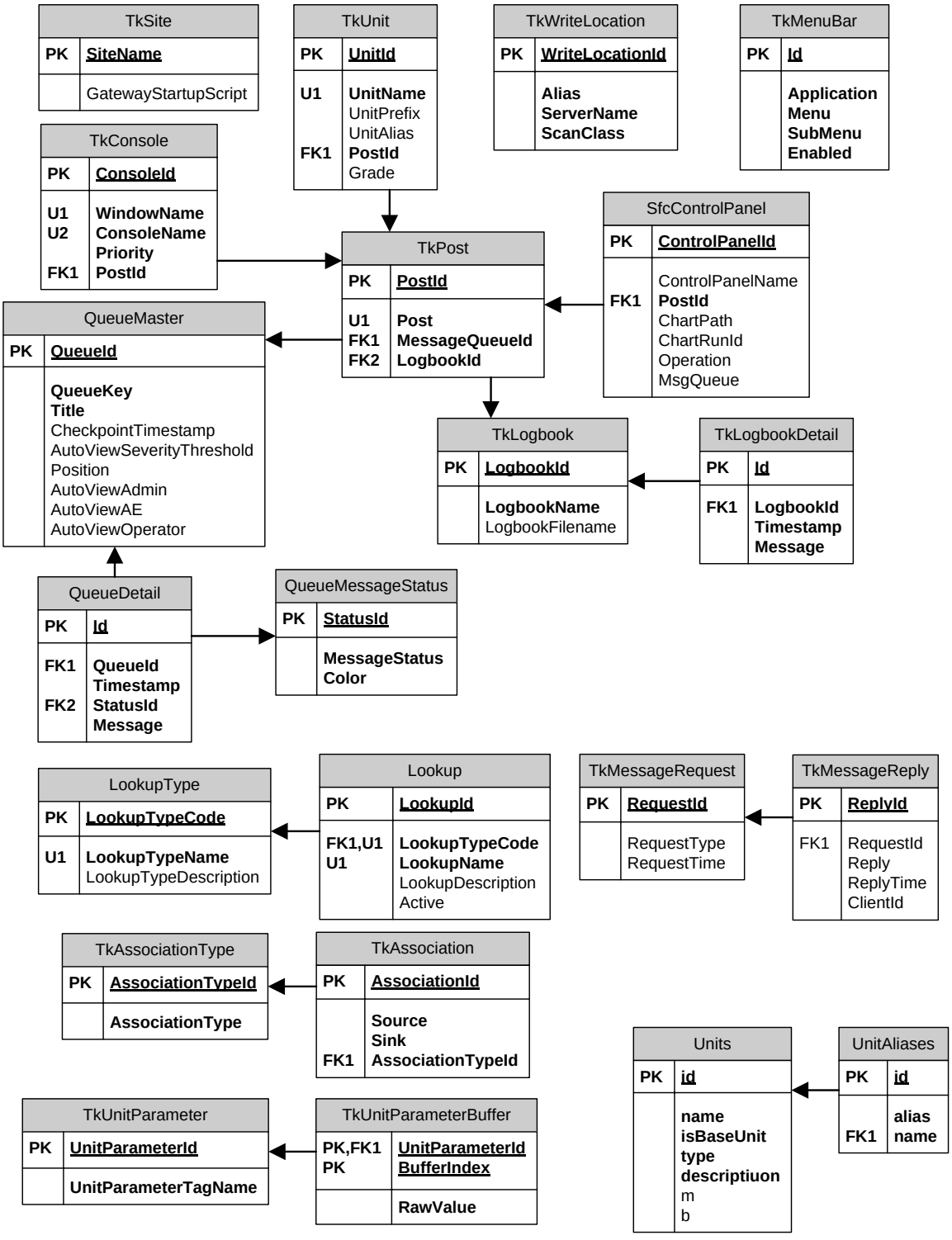
Finally, there is also a database dedicated solely to tag history. This database is commonly referred to as "XOMhistory"

2 Common Tables and Views

This section describes common tables and views.

2.1 Common Infrastructure Tables

This section describes tables that are common to the several of the toolkits and utilities.



2.1.1 TkSite Table

This table contains a definition of the site. There must be exactly one record. The main purpose of this table is to define the site specific Ignition startup script for the site. This is crucial for implementing a generic Ignition project. The generic site specific startup procedure reads this table and calls the site specific method specified in the database.

Column	Description	Datatype
SiteName	The name of the site, for reference only	Varchar(50)
GatewayStartupScript	The fully qualified name of the gateway startup script for this site.	Varchar(500), NN

2.1.2 TkPost Table

This table contains a definition of a post. A post corresponds to an operator logged into a workstation. The name of the post is the same as the username the operator will use to log in to the workstation.

Column	Description	Datatype
PostId	System defined id for this post.	Integer, PK
Post	The name of the post	Varchar(50), UK, NN
MessageQueueId	Id of the queue that will be used for this post.	Integer, FK
LogbookId	Id of the logbook used for this post	Integer, FK

2.1.3 TkUnit Table

This table defines a unit. A unit is a somewhat vague concept, but it corresponds to something physical.

Column	Description	Datatype
UnitId	The name of the Ignition role, i.e., AE, Operator.	Varchar(50), PK, NN
UnitName	The name of the unit	Varchar(50), UK, NN

Column	Description	Datatype
UnitPrefix	Prefix to be used when certain abbreviation and concatenations are used. This was relevant in the old system to help define a set of recipe tables for a unit. (I'n not sure if this is used in the new system).	Varchar(50)
unitAlias	An alias for the unit. (I'm not sure where this was used in the old platform and if it will be used in the new application)	Varchar(50)

2.1.4 TkConsole Table

This table lists the windows that will be opened for a post when the operator logs in to the system. The main window that is anchored to the left side is often referred to as the console, but any number of windows or consoles can be opened.

Column	Description	Datatype
ConsoleId	System defined id for this record.	integer, PK
PostId	Id of the post for this window.	Integer, FK, NN
WindowName	The full path to the Vision window that will be displayed for this console.	Varchar(100), UK, NN
ConsoleName	The name of the console as it appears in the pull-down console menu. The common XOM project is configured with all consoles for all sites. The data in this table defines which of those are relevant for an individual site. Irrelevant consoles will be deleted by the client startup script.	Varchar(100), NN, UK
Priority	Used for ordering the windows	Integer

2.1.5 TkWriteLocation Table

This table defines the OPC servers and their scan classes. This table should reflect what is configured in Ignition. There is not a mechanism to update Ignition from the contents of this table

Column	Description	Datatype
WriteLocationId	System defined id for this write location.	Int, PK

Column	Description	Datatype
Alias	A common name for this write location which may correspond to an alias used in the old recipe toolkit and throughout the old application.	Varchar(max), UK, NN
ServerName	The name of the OPC Connection in Ignition.	Varchar(max), NN
ScanClass	The name of the scan class in Ignition.	Varchar(max), NN

2.1.6 QueueMaster Table

This table defines queue instances. It contains one record for each queue.

Column	Description	Datatype
QueueId	A system defined id	Integer, PK, NN
QueueKey	A unique key for this queue that will be used during insert and fetches. (A preferred name for this column would be Key but that is a reserved word)	Varchar(50), UK, NN
Title	The title of the queue used in the user interface.	Varchar(100), NN
CheckpointTimestamp	This is used to mark a certain point in time that serves as way to distinguish between historic and current messages. This is most often used by the sequential control toolkit at the beginning of an operation.	Datetime
AutoViewSeverityThreshold	When a new message is inserted into the queue, if the severity of the message is greater than this value, then the queue view is shown on the client based on the role of the client and the corresponding AutoView setting	Real, NN
Position	Specifies where the queue view will be displayed if the view is auto displayed.	Varchar(50), NN
AutoViewAdmin	Specifies if the Auto View threshold applies to this client	Bit, NN
AutoViewAE	Specifies if the Auto View threshold applies to this client	Bit, NN

Column	Description	Datatype
AutoViewOperator	Specifies if the Auto View threshold applies to this client	Bit, NN

2.1.7 QueueDetail Table

This table contains the contents of a queue. There is one record for each entry in the queue.

Column	Description	Datatype
Id	A system defined id	Onteger, PK, NN
QueueId	The id of the queue for this message. Foreign key to the QueueMaster table.	Integer, FK, NN
Timestamp	The timestamp, generally the system time of when the message was inserted.	Datetime, NN
StatusId	The id of the status, which determines the background color, of the message in the UI. Foreign key to the QueueMessageStatus table.	Integer, NN, FK
Message	The text body of the entry.	Varchar(2000), NN

2.1.8 QueueMessageStatus Table

This table contains the mapping from message status to colors used in the queue user interface. The status/color mapping is global and applies to all queues in the application. Changes made in this table will be applied in the application the next time the queue view is opened.

Column	Description	Datatype
StatusId	A system defined id	Integer, PK, NN
MessageStatus	The status of the message.	Varchar(15), NN
Color	The background color that will be used in the queue user interface. The naming convention for colors is the generally accepted color names, although it is possible RGB values can also be used.	Varchar(15), NN

2.1.9 TkLogbook Table

This table contains the master definition for logbooks. The convention is that there is a logbook for each post and one for engineers.

Column	Description	Datatype
LogbookId	A system defined id	Integer, PK, NN
LogbookName	The common name for this logbook.	Varchar(15), NN
LogbookFilename	Filename used for writing the daily logbook to disk.	Varchar(15)

2.1.10TkLogbookDetail Table

This table contains the contents of the logbook. There is a record for each .

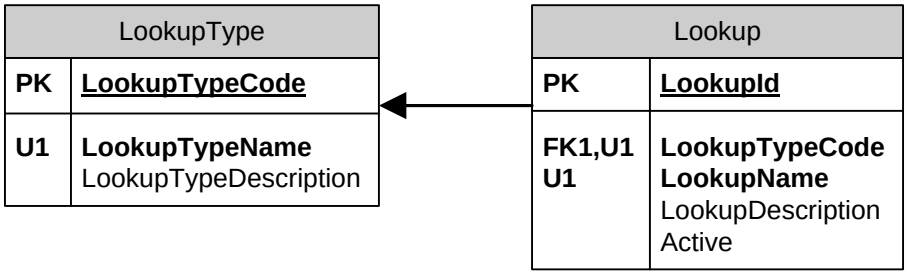
Column	Description	Datatype
Id	A system defined id	Integer, PK, NN
LogbookId	The status of the message.	Varchar(15), NN
Timestamp	Datetime when the record was inserted	Varchar(15), NN
Message	The text written to the logbook	Varchar(), NN

2.1.11SfcControlPanel Table

Refer to section 6 for details.

2.2 Lookup Tables

This section describes tables that are used in the lookup table facility. There are numerous different lookups used throughout the application. Rather than define a separate table for each of the different lookup types, this generic framework is provided.



2.2.1 LookupType Table

This table defines the constants necessary for the linear transformation from one unit to another.

Column	Description	Datatype
LookupTypeCode	A user defined code that defines the lookup type. A code is used rather than a system id to make maintaining the lookup table somewhat easier	Varchar(15), NN, PK
LookupTypeName	The name of the unit.	Varchar(64), UK, NN
LookupTypeDescription	Optional description of the units	Varchar(2000)

2.2.2 Lookup Table

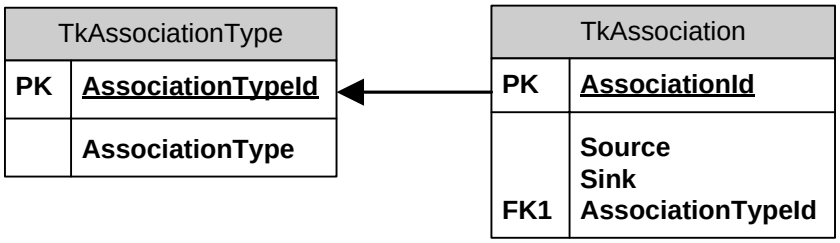
This table defines lookup values.

Column	Description	Datatype
LookupId	System assigned unique id.	Int, PK, NN
LookupTypeCode	Defines the type of the lookup.	Varchar(15), UK1, NN
LookupName	The lookup value that will be displayed in the dropdown.	Varchar(50), UK1, NN
LookupDescription	An optional description of the lookup	Varchar(500)

Column	Description	Datatype
Active	Specifies if the value should be included in the dropdown list. This is used rather than deleting the lookup record in order to prevent foreign key integrity constraints.	Bit, NN

2.3 Association Tables

This section describes tables that are used to define an association between entities. In a generic sense this replaces the relation and connection constructs in G2.



2.3.1 TkAssociationType Table

This table defines the association types.

Column	Description	Datatype
AssociationTypeId	Unique system assigned id	Int, NN, PK
AssociationType	The name of the association.	Varchar(64), UK, NN

2.3.2 TkAssociation Table

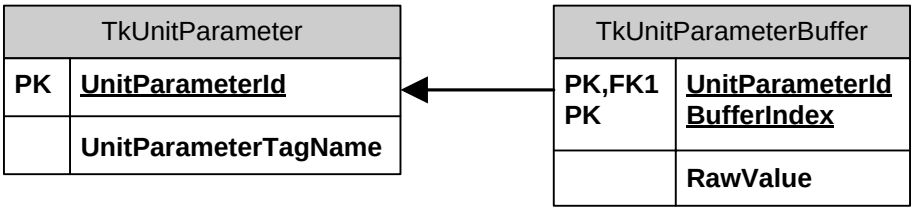
This table defines the association types. Associations can be directed by paying attention to the source and sink.

Column	Description	Datatype
AssociationId	Unique system assigned id	Int, NN, PK
Source	Th	
Sink		

Column	Description	Datatype
AssociationTypeId	The type of the association.	Varchar(64), UK, NN

2.4 Unit Parameter Tables

This section describes tables that are used to define and implement UnitParameter tags. A unit parameter is basically a global data entity that has built in filtering in the form of averaging the last n raw values. There is a UnitParameter UDT that defines Unit Parameter behavior and an instance of that UDT should exist for every row in this table.



2.4.1 TkUnitParameter Table

This table defines the unit parameter.

Column	Description	Datatype
UnitParameterId	Unique system assigned id	Int, NN, PK
UnitParameterTagName	The name of the unit parameter UDT instance.	Varchar(150), UK, NN

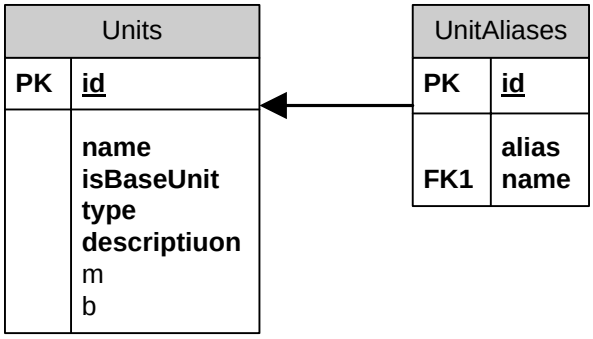
2.4.2 TkUnitParameterBuffer Table

This table implements the circular buffer used to store the history of values for a unit parameter. The implementation of the circular key is managed by the change handler on the unit parameter UDT. It bumps the BufferIndex and updates the appropriate cell in the circular buffer.

Column	Description	Datatype
UnitParameterId	Unique system assigned id	Int, NN, PK
BufferIndex	A pointer to the end of the circular queue.	Int, NN, PK
RawValue	The value	float

2.5 Engineering Unit Translation Tables

This section describes tables that are used in the common engineering unit translation utility.



2.5.1 Units Table

This table defines the constants necessary for the linear transformation from one engineering unit to another.

Column	Description	Datatype
Id	Unique system assigned id	Int, NN, PK
Name	The name of the unit.	Varchar(64), UK, NN
isBaseUnit	A base unit defines the reference that all other units of the same type are with respect to. There should always be exactly one base per type.	bit, NN
Type	The type of the unit, i.e, length, weight, temperature, etc.	Varchar(64), NN
Description	Optional description of the units	Varchar(2000)
M	The slope variable in the y=mx+b	Float
B	The y-intercept in y=mx+b	Float

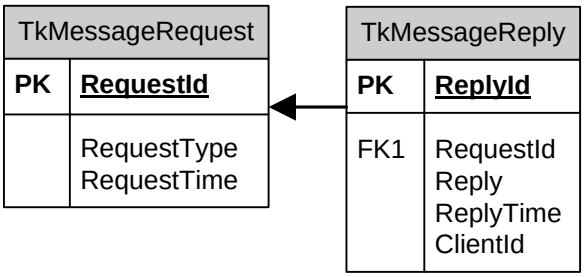
2.5.2 UnitAliases Table

This table contains aliases that are used for the situation where multiple names may be used for the same engineering unit, i.e., degC, DegCelcius, DegCentigrade.

Column	Description	Datatype
Id	System assigned unique id.	Int, PK, NN
Alias	An equivalent name for a unit defined in the Units table.	Varchar(64), UK, NN
Name	The name of a unit in the Units table.	Varchar(64), NN

2.6 Gateway/Client Message Tables

This section describes tables that are used in the common engineering unit translation utility.



2.6.1 TkMessageRequest Table

This table defines the constants necessary for the linear transformation from one engineering unit to another.

Column	Description	Datatype
RequestId	Unique system assigned id	Int, NN, PK
RequestType	The name of the unit.	Varchar(64), UK, NN
RequestTime	A base unit defines the reference that all other units of the same type are with respect to. There should always be exactly one base per type.	datetime, NN

2.6.2 TkMessageReply Table

This table contains.

Column	Description	Datatype
ReplyId	System assigned unique id.	Int, PK, NN
RequestId	The id of the request that this reply is in response to.	Int, FK, NN
Reply		
ReplyTime		Datetime
ClientId		Varchar, NN

2.7 Miscellaneous Tables

This section defines miscellaneous tables.

2.7.1 RoleTranslation Table

This table contains a translation from the Windows user roles defined by ExxonMobils IT policy and logical roles used in Ignition. The reason for this table is that there isn't a single role for AE that would be good for all of the ExxonMobil sites. Furthermore, a G-Line AE should not have access to a Vistalon site. Unfortunately, there is not a way to use this translation in any component level security configurations.

Column	Description	Datatype
IgnitionRole	The name of the Ignition role, i.e., AE, Operator.	Varchar(50), PK, NN
WindowsRole	The name of the Windows role	Varchar(50), UK, NN
QueueId	Id of the queue that will be used for this console.	Integer, FK, NN

2.7.2 TkMenuBar Table

This table contains information about how to configure the main menu. This is necessary because a common project is used for all sites but the menu needs to be customized for each site. The project contains the union of all site specific menus.

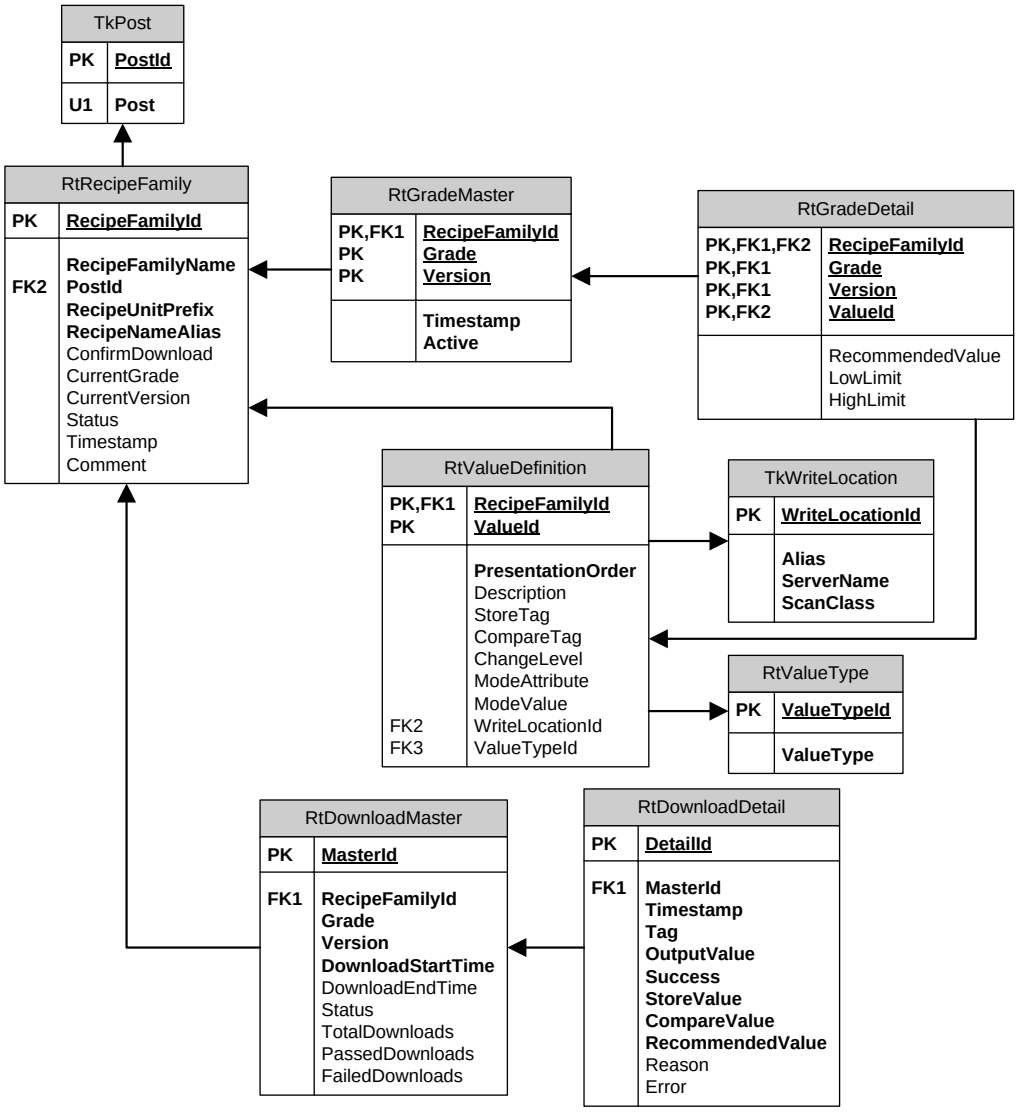
Column	Description	Datatype
Id	System assigned id.	Int, PK, NN
Application	There are two projects, both of which share this table: XOM and dbManager	Varchar(50), NN
Menu	The name of the menu.	Varchar(50), NN
SubMenu	The name of the sub-menu.	Varchar(50), NN
Enabled	Specifies if the submenu is enabled or disabled.	Bit, NN

3 Recipe Toolkit Tables and Views

This section describes the tables that pertain to the Recipe Data toolkit. It DOES NOT include the recipe data database tables, but only the tables required to implement the recipe data functionality.

3.1 Core Recipe Tables

The relationship between the core tables that make up the recipe database is shown below:



3.1.1 RtRecipeFamily

This table corresponds to the class definition recipe-cell-contents in the old platform. In the Vistalon application, the records in the database correspond to the named instances RLA3, RLA3_TWR, and VFU. Records in this table are manually inserted and configured via SQL*Server. Because these records are very static, there are only a couple of rows, and they are not likely to change, there is not a GUI to configure them.

Column	Description	Datatype
RecipeFamilyId	System defined id that uniquely identifies a recipe family	Integer, PK
RecipeFamilyName	Identifies the recipe family.	Varchar(50), UK, NN
RecipeUnitPrefix	Not sure if this is used	Varchar(50), NN
RecipeNameAlias	Not sure if this is used	Varchar(50), NN
PostId	The post that can download or view this recipe.	Integer, FK, NN
ConfirmDownload	Specifies if the operator must confirm the recipe download.	Bit
CurrentRecipeGrade	The id of the currently selected recipe. This is dynamically updated by the recipe download system.	Varchar(256), NN
CurrentRecipeVersion	The version of the currently selected recipe. This is dynamically updated by the recipe download system.	Integer
Status	The status of the recipe download action. This is dynamically updated by the recipe download system.	Varchar(50)
Timestamp	The time that this record was last updated. This is dynamically updated by the recipe download system.	Datetime

3.1.2 RtGradeMaster

This table holds the list of grades available for processing units. Its primary key, *recipeFamilyId,grade,version* corresponds to a family-grade or family-product combination. It also contains a flag marking the active version. The application guarantees that at most only one version is active for any unit-grade combination.

Column	Description	Datatype
RecipeFamilyId	The recipe family	Integer, PK
Grade	The grade	Varchar(50), PK

Column	Description	Datatype
Version	The version	Integer, PK
Timestamp	Time the recipe was created	Datetime, NN
Active	There can only be one active version for a Grade. The active version will be the one that is automatically downloaded.	Bit

3.1.3 RtGradeDetail

Rows in this table hold settings for a single recipe parameter. The primary key for the table is a combination of *RecipeFamilyId, Grade, Version* and *ValueId*

Column	Description	Datatype
RecipeFamilyId	The recipe family	Integer, PK
Grade	The grade	Varchar(50), PK
Version	The version	Integer, PK
ValueId		Integer, NN
RecommendedValue	The recommended value to be downloaded to the DCS tag or Ignition tag. Unless changed by the operator, the PEND value in the Operator spreadsheet will be the Recommend value. Note that a quantity (number) or text entry is allowed. Text values can be downloaded to DCS tags that support such features.	Varchar(max)
LowLimit	The lowest value allowed to be entered into the Recommend column by a user of dbManager or the PEND column by a user of the Console Operator spreadsheet.	Varchar(max)
HighLimit	The highest value allowed to be entered into the Recommend column by a user of dbManager or the PEND column by a user of the Console Operator spreadsheet.	Varchar(max)

3.1.4 RtValueDefinition

This table defines all of the parameters that are included in the recipe for a unit. It defines and orders recipe parameters for a unit. When retrieving parameters for a recipe, the definitions should `ORDER BY` *presentationOrder*.

Column	Description	Datatype
ValueId	This uniquely identifies a recipe database	Varchar(10), PK
RecipeFamilyId	Identifies the recipe family.	Integer, FK, NN

Column	Description	Datatype
PresentationOrder	Defines the order in which parameter records will appear when the recipe is viewed. The order is important because this is the order of download to the DCS or Ignition tag as the recipe is instantiated. In the old system the order was manually set, in the new system this value is automatically adjusted as rows are dragged and dropped on the GUI.	Integer, NN
Description		Varchar(max)
StoreTag	An Ignition tag, either a memory tag or an OPC tag, intended to receive the data. Tags are created dynamically within the Recipe Toolkit, as needed. Use a tag name of the form DTA(nn) or PVC(nn) for parameters which are arrays, where nn is the number of the element. For example CAF001:DTA(13) in the DCS would be CAF001.DTA(13) in the recipe.	Varchar(max)
CompareTag	Either a DCS tag or Ignition tag that holds a current setting shown to an operator, whereas the StoreTag holds the ultimate target which may be reachable only after some process delay. In the case of a process delay, the StoreTag value may be a target stored to a data word and used by an application to set a process SP after some delay and the CompareTag value would be the process SP that will be eventually set equal to the store value. In many cases both the StoreTag and CompareTag are the same DCS tag or Ignition tag. Note: If you specify a compare value but not a store value the Recipe Toolkit will ignore the compare value. If you specify a store value but not a compare value the Recipe Toolkit will set the compare value to the same tag_parameter as the store value.	Varchar(max)

Column	Description	Datatype
ChangeLevel	ChangeLevel: Defines whether the operator does or does not have access to the PEND value. (NOTE: There is no PEND value in any table) The following entries are valid for the change level. **OC**: Operator (anyone using operator mode in Ignition) can see and change the PEND value. **AE**: Operator can see PEND value but cannot change the value. **EO**: OC can neither see nor change the PEND value. **CC**: Comment line that can be used to provide a visual separation between row entries or logical grouping of rows. No StoreTag or CompareTag is entered. If a PEND value is entered, the value is shown appended to the end of the Description in the Ignition spreadsheet. Normally, the PEND value is left blank.	Varchar(max)
ModeAttribute	Certain settings in a DCS require that a controller be put into a mode attribute setting so that an application external to the DCS can download a value to the controller. If the controller requires such a setup, the tag attribute to set is filled in this field. For example, a DCS could require that a controller mode attribute be put to PROGRAM to accept values. The Recipe Toolkit should in an ideal scenario place the controller to the correct setting and return the controller to the previous setting when the value is downloaded. Note: At this time, the operator is required to have the controller in the proper control mode (computer, cascade, etc.) so that a value such as a setpoint can be successfully downloaded. For example, if a SP is downloaded to a controller in MAN with PV tracking enabled, the SP store would fail.	Varchar(max)
ModeValue	Defines the value (text or number) to be downloaded to ModeAttribute.	Varchar(max)
WriteLocation	An alias for the system where the recipe value will be written.	Varchar(max)

Column	Description	Datatype
ValueTypeId	Defines the type of the value (String, Float, or Integer. Foreign key to the RtValueType table. Most recipe data is a float with the notable exception of the grade tag.	Int, FK

3.1.5 RtValueType

This table defines the legal data types for recipe values. Initially, the system supports floats, integers and Strings.

Column	Description	Datatype
ValueTypeId	System generated id	Integer, PK
ValueType	The data type	Varchar(50)

3.2 Download Log Tables

This section describes the tables that log a recipe download from the recipe database to the DCS. Even though these tables are populated from the on-line control application, they are also visible from the DB Manager application.

3.2.1 RtDownloadMaster

This table contains a record for each recipe download. This table corresponds to the class definition *recipe-cell-contents* in the old platform. In the Vistalon application, the records in the database correspond to the named instances RLA3, RLA3_TWR, and VFU.

Column	Description	Datatype
MasterId	System defined id	Integer, PK
RecipeFamilyId	The recipe family	Integer, FK, NN
Grade	The grade being downloaded	Varchar(50), NN
Version	The version of the recipe	Integer, NN
DownloadStartTime	The time that the download started	Datetime, NN
DownloadEndTime	The time that the download finished	Datetime
Status	“Success” or “Failure”. Success if all tags that were scheduled to be downloaded were successfully written	Varchar(50)
TotalDownloads	# of tags that were scheduled to be written.	Integer
PassedDownloads	# of tags that were successfully written	Integer
FailedDownloads	# of tags that were NOT successfully written	Integer

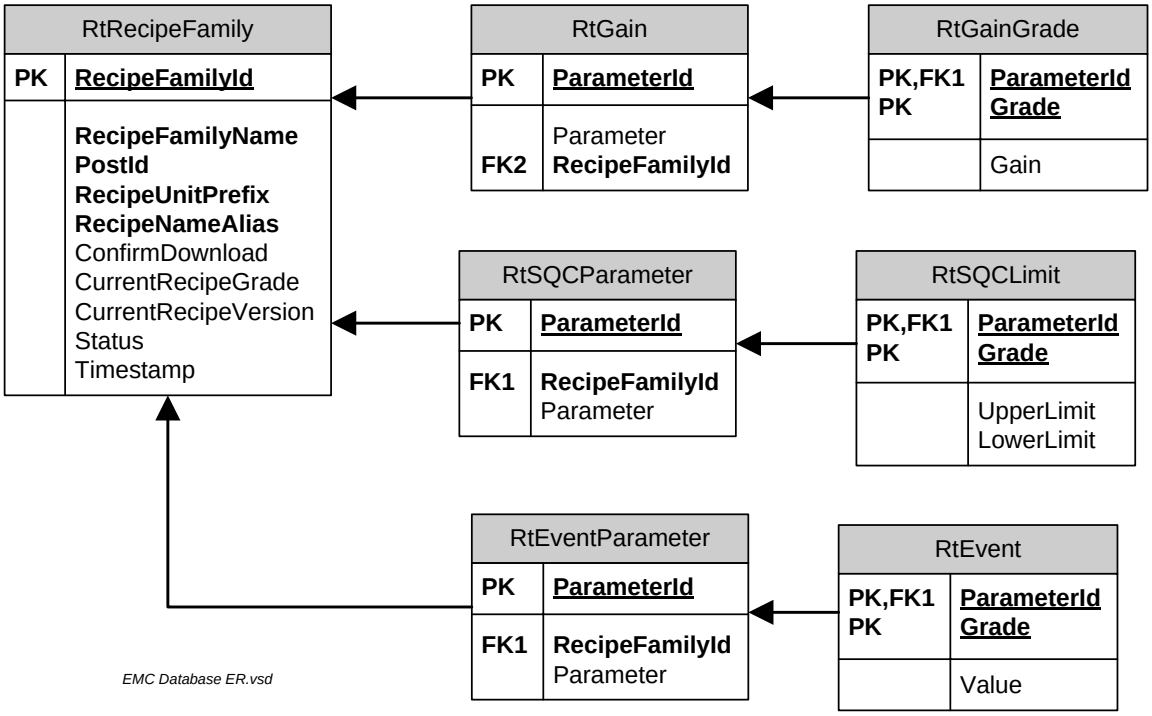
3.2.2 RtDownloadDetail

This table stores the individual record of each tag write that is attempted including the tag, value, and the error if it failed.

Column	Description	Datatype
DetailId	System defined id	Integer, PK
MasterId	The id of the download session. This column allows NULL values in order to be able to record tag writes that occur outside the scope of a recipe download. (I'm not sure if this is necessary)	Integer, FK
Timestamp	Time of the download	Datetime, NN
Tag	The tag name	Varchar, NN
OutputValue	The value that was written.	float, NN
Success	True is successfully written, False otherwise.	bit, NN
StoreValue	The value of the STORE tag before the download	Float, NN
CompareValue	The value of the COMPARE tag before the download	Float, NN
RecommendedValue	The value recommended from the recipe.	Float, NN
Reason	The reason (free text) for downloading a PEND value that is different from the Recommend value. After the operator enters a reason, the operator can then change the PEND value.	Varchar(2000)
Error	The reason that the write failed, NULL if it succeeded	Varchar(2000)

3.3 SQC Tables

There are a collection of tables were historically contained in an SQC database. Sites will only implement the tables relevant to that site, rarely are all tables implemented.



3.3.1 RtSQCParameter

Defines the SQC parameters for a unit. The primary key is *parameterId*. The list of parameters is not consistent across units, but it is for every grade for a unit.

Column	Description	Datatype
ParameterId	System defined id	Integer, PK
RecipeFamilyId	Id of the family this parameter applies to	Integer, NN
Parameter	The parameter name	Varchar(50), NN

3.3.2 RtSQCLimit

Upper and lower SQC limits are stored for the various parameters. by *initId* , *grade*, and *parameterId*..

Column	Description	Datatype
ParameterId	Id of the SQC parameter	Integer, PK
Grade	The grade being downloaded	Varchar(50), NN
Version	The version of the recipe	Integer, NN
UpperLimit	The upper SQC limit	Float

Column	Description	Datatype
LowerLimit	The lower SQC limit	Float

3.3.3 RtGain

This table defines the gain parameters that apply to a specific unit.

Column	Description	Datatype
ParameterId	System defined id	Integer, PK
RecipeFamilyId	Id of the family this parameter applies to	Integer, NN
Parameter	The name of the gain parameter	Varchar(50), NN

3.3.4 RtGainGrade

This table contains the gain values for a specific grade parameter pair.

Column	Description	Datatype
ParameterId	System defined id	Integer, PK
Grade	The grade that this gain applies to	Varchar(50), NN
Gain	The gain value	Float

3.3.5 RtEventParameter

This table defines the event parameters that apply to a specific unit.

Column	Description	Datatype
ParameterId	System defined id	Integer, PK
RecipeFamilyId	Id of the family this parameter applies to	Integer, NN
Parameter	The name of the event parameter	Varchar(max), NN

3.3.6 RtEvent

This table contains the event values for a specific grade parameter pair.

Column	Description	Datatype
ParameterId	System defined id	Integer, PK
Grade	The grade being downloaded	Varchar(50), NN
Value	The event value	Float, NN

3.4 Miscellaneous Tables

This section describes the remaining tables that are generally considered part of the recipe toolkit.

3.4.1 RtAdhocCatalog

This table lists the adhoc database tables that are custom for the site. Putting a record in this table with a valid table definition in the database will allow the generic table editor to work in *DBManager*.

Column	Description	Datatype
TableName	System defined id	Integer, PK

3.4.2 RtAllowedFlyingSwitch

This table defines the allowed flying switch transitions.

Column	Description	Datatype
Id	System generated key	Integer, PK
CurrentGrade	The current grade	Varchar(50), NN
NextGrade	The next grade	Varchar(50), NN

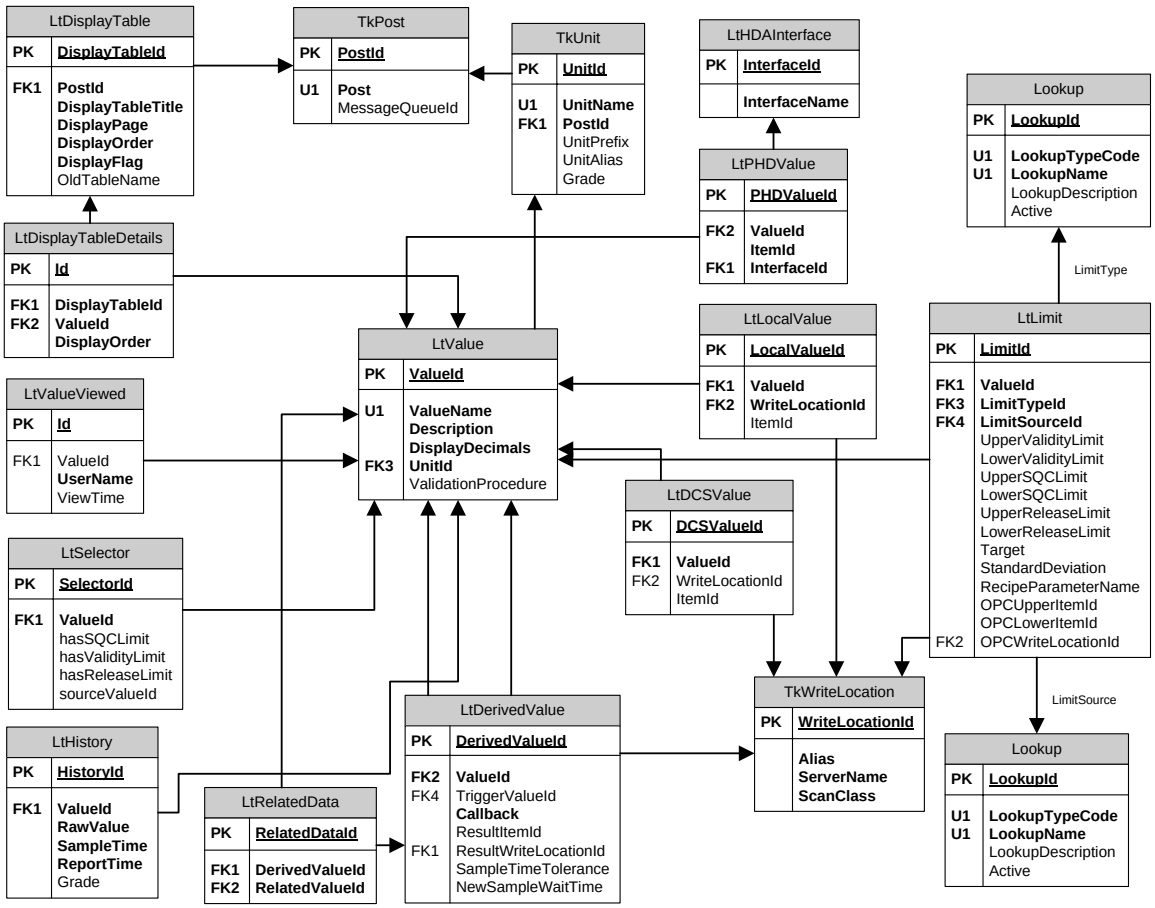
3.4.3 RtWatchDog

I have no idea how data gets into this table or what the data means.

Column	Description	Datatype
Observation	System defined id	Integer, PK
Timestamp		Datetime, NN

4 Lab Data Tables and Views

This section describes the tables that pertain to the Lab Data toolkit. The relationship between the Lab Data tables is shown below



4.1 Core Tables

The core Lab Data tables are discussed below.

4.1.1 LtValue Table

This table defines a lab measurement.

Column	Description	Datatype
ValueId	System defined id	Integer, PK, NN
ValueName	The name of the measurement	Varchar(50), UK, NN
Description	Optional description of the measurement	Varchar(500)
DisplayDecimals	The number of decimals to display in the user interface for this measurement	Integer, NN
UnitId	Id of the unit that is associated with this lab data.	Integer, FK, NN
ValidationProcedure	Name of Python method that can provide validation of the value.	Varchar(250)

4.1.2 LtHistory Table

This table contains a history of all of the laboratory measurements.

Column	Description	Datatype
HistoryId	System defined Id	Integer, PK, NN
ValueId	Id of the measurement	Integer, FK, NN
RawValue	The value of the measurement	Float, NN
SampleTime	The time that the sample was taken and sent to the lab	Datetime, NN
ReportTime	The time that the value was reported.	Datetime, NN
Grade	The grade that was running at the time the sample was taken.	Varchar(50)

4.1.3 LtDisplayTable Table

This table defines the organization of the screens that the operator uses to view lab data.

Column	Description	Datatype
DisplayTableId	System defined Id	Integer, PK, NN
DisplayTableTitle		NN Varchar(50)
DisplayPage	Specifies the page/tab that this table name will appear on for the lab data chooser window.	Integer, NN
DisplayOrder	The order on the page that this	Integer, NN,
PostId	Id of the post where this will be displayed	FK
DisplayFlag		
OldTableName		

4.1.4 LtDisplayTableDetails Table

This table defines the lab values that are displayed in a particular display table.

Column	Description	Datatype
Id	System defined Id	Integer, PK, NN
DisplayTableId	Id of the display table	Integer, NN, FK
ValueId	Id of the lab value	Integer, NN, PK
DisplayOrder	The order of this lab value in the table.	Integer, NN

4.1.5 LtPHDValue Table

For a measurement that is received from the PHD data source, this table defines the information necessary to acquire the measurement.

Column	Description	Datatype
PHDValueId	System defined Id	Integer, PK, NN
ValueId	The id of the measurement	Integer, NN, FK
ItemId	The item-id of tag in PHD for this measurement	Varchar(50), NN
InterfaceId	The id of the PHD interface	Integer, NN, FK

4.1.6 LtDCSValue Table

For a measurement that is received from a DCS data source, this table defines the information necessary to acquire the measurement.

Column	Description	Datatype
DCSValueId	System defined Id	Integer, PK, NN
ValueId	The id of the measurement	Integer, NN, FK
ItemId	The item-id of tag in PHD for this measurement	Varchar(50), NN
WriteLocationId	The id of the interface/scan class to use for reading this value	Integer, NN, FK

Should this table contain information about where to store the data?

4.1.7 LtLocalValue Table

This table defines measurements that will ALWAYS be entered manually. It also defines the location in PHD where the value will be written.

Column	Description	Datatype
LocalValueId	System defined Id	Integer, PK, NN
ValueId	The id of the measurement	Integer, NN, FK
ItemId	The item-id of tag in PHD where the value will be stored after it is manually entered.	Varchar(50), NN
InterfaceId	The id of the PHD interface	Integer, NN, FK

4.1.8 LtDerivedValue Table

This table defines lab values that are derived from one or more lab data and or tag values. A derived lab data always has a trigger value which initiates the calculation. There may be one or more related values (see below) that are also used in the calculation. The calculation is performed in a custom Python script that is called once the trigger and related data are consistent.

Column	Description	Datatype
DerivedValueId	System defined Id	Integer, PK, NN
ValueId	The id of the measurement	Integer, NN, FK
TriggerValueId		
Callback	The Python script that will perform the calculation	
ResultItemId	The item-id of tag in PHD where the value will be stored after it is manually entered.	Varchar(50), NN
ResultWriteLocataionId		

Column	Description	Datatype
SampleTimeTolerance	The time window that defines a consistent data. The trigger and related data must all have a sample times that fall within this window to be consistent	Integer, NN
NewSampleWaitTime	The time that the system will wait for the related data to become consistent with the trigger data	Integer, NN

4.1.9 LtRelatedData Table

This table defines data that is related to a derived value. Typically the related data is used in the calculation of the derived value and must be consistent with the trigger value.

Column	Description	Datatype
RelatedDataId	System defined Id	Integer, PK, NN
DerivedValueId	The id of the derived value that this data is related to	Integer, NN, FK
RelatedValueId	The id of the lab data that is related to the derived value.	Integer, NN, FK

4.1.10LtHDAInterface Table

This table contains a definition of the PHD interfaces that support OPC-HDA to acquire the lab measurements.

Column	Description	Datatype
InterfaceId	System defined Id	Integer, PK, NN
InterfaceName	Name of the HDA interface in Ignition	Varchar(50), NN

4.1.11LtLimit Table

This table contains limit information. It has columns for validity, SQC, and release limits. They all allow NULL values but it is assumed that at least one pair of limits is not null.

Column	Description	Datatype
LimitId		Integer, PK, NN
ValueId	Id of the corresponding measurement	Integer, NN, FK
LimitTypeId		Int, FK
LimitSourceId		Int, FK
UpperValidityLimit		Float
LowerValidityLimit		Float
UpperSQCLimit		Float
LowerSQCLimit		Float
UpperReleaseLimit		Float
LowerReleaseLimit		Float
Target		Float
StandardDeviation		Float

Column	Description	Datatype
RecipeParameterName		Varchar(100)
OPCUpperItemId		Varchar(50)
OPCLowerItemId		Varchar(50)
OPCWriteLocationId		Int, FK

4.1.12LtSelector Table

A lab data selector is used to switch the source where measurements are taken from. This generally is used to account for some change in the physical configuration of the unit. The power of selectors is that it allows the other toolkits (diagnostic and sequential control) to look at a consistent set of lab objects independent of the physical configuration of the unit. For example, the various toolkits reference a lab data object named Mooney-Lab-Data even though the source of the data is different depending on whether the reactor configuration is single or series. The business rules that determine the source of the selector is often multi-dimensional, such as reactor configuration and flash drum usage. Therefore, the logic that configures the source of the selector is best implemented in a custom Python script rather than in a database table. This table contains the definition of a selector.

Column	Description	Datatype
SelectorId	System defined Id	Integer, PK, FK, NN
ValueId		Int, NN
hasSQCLimit		Bit, NN
hasValidityLimit		Bit, NN
hasReleaseLimit		Bit, NN
sourceValueId		Int

4.1.13LtValueViewed Table

This table records when a user viewed a particular lab value. This is used to implement the animation that helps notify the user when new lab data has arrived.

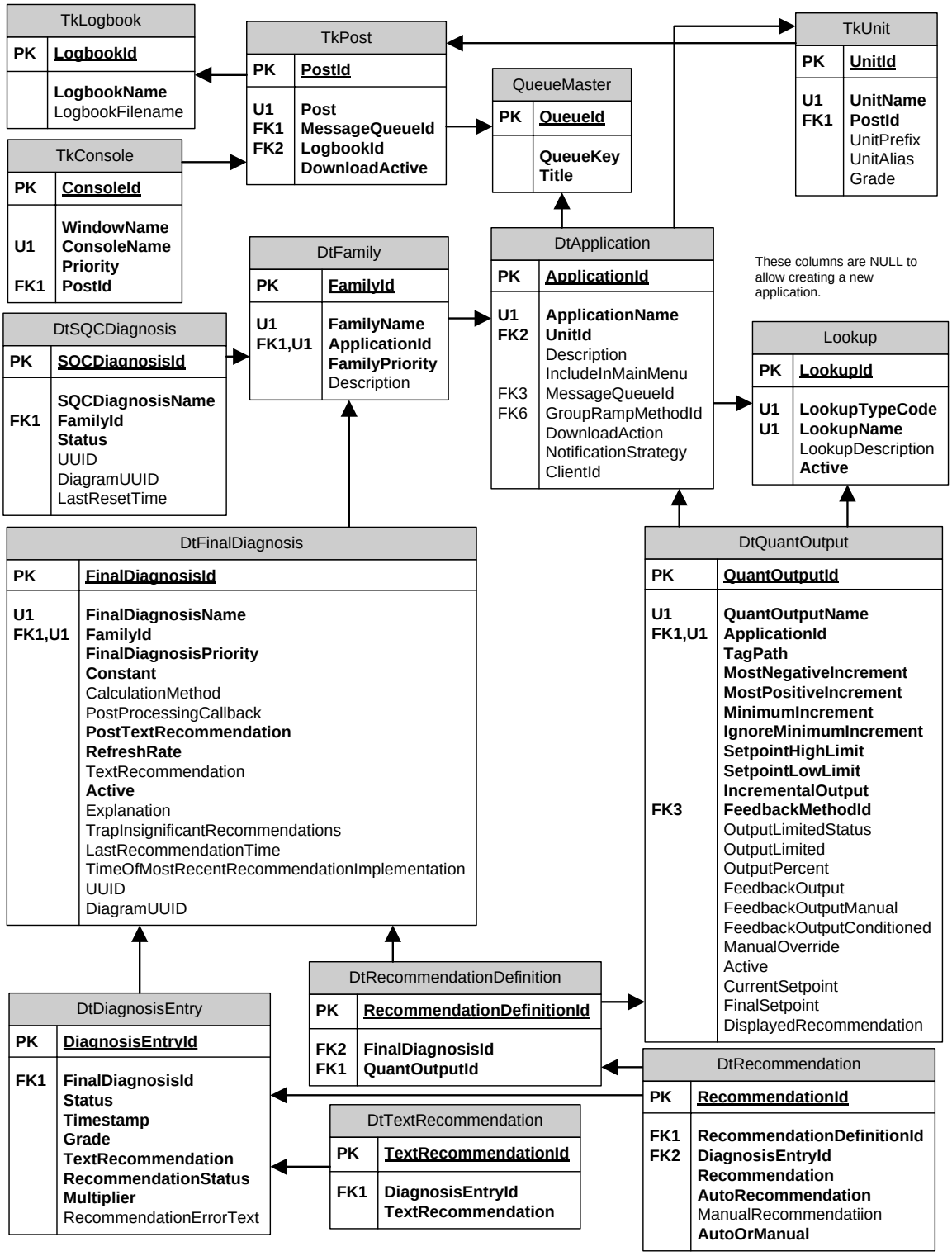
Column	Description	Datatype
Id	System defined Id	Integer, PK, NN
ValueId	The id of the lab value	Integer, NN, FK, UK1
Username	The user's username	Varchar(25), NN, UK1
ViewTime	The time that the value was viewed by the user	Datetime, NN

5 Diagnostic Toolkit Tables and Views

This section describes the tables that pertain to Diagnostic toolkit.

5.1 Entity Relationship Diagram

The diagram shown below shows the tables used in the Diagnostic Toolkit and shows the relationship between tables.



5.2 Tables

The tables used by the diagnostic toolkit are described below.

5.2.1 DtApplication Table

This table defines the diagnostic applications present in the project. There is one record for each application. Records in this table correspond to applications in the Diagnostic Toolkit resource tree in Ignition. The data is maintained by configuring (adding, deleting, updating) applications in Ignition Designer. Although this table needs to be completely configured for the Application to be functional, the columns allow NULL values so that a new application row can be inserted as applications are created / imported.

Column	Description	Datatype
ApplicationId	A unique system generated id.	Integer, PK
ApplicationName	A unique name for the application.	Varchar(250), NN, UK
UnitId	The id of the unit that this application is associated with. The post can be found using the unit.	Integer
Description	An optional description of the application.	Varchar(2000)
IncludeInMainMenu	Flag to indicate if this application will be displayed in some pull down menu. (I'm not sure if there is a pull-down menu in the new system)	Bit
MessageQueueId	The key to use for the message queue for any post message blocks in the application.	Integer
GroupRampMethodId	Not sure if this is used	Integer
DownloadAction	The most recent download action taken for this application.	Varchar(50)
NotificationStrategy	There are strategies: "ocAlert" and "clientId". "ocAlert" is the default strategy where the standard OC alert loud window is used to notify clients that there is an alert for this application. The "clientId" strategy was implemented specifically for Rate Change where the action is triggered from a client by pressing a button and we want the setpoint to be displayed on this client as fast as possible without the OC alert.	Varchar(50)

Column	Description	Datatype
ClientId	If the NotifiationStrategy is ClientId then this is the id of the client to notify via a message.	Varchar(50)

5.2.2 DtFamily Table

This table defines a family of problems. There is one record for each family. Records in this table correspond to families in the Diagnostic Toolkit resource tree in Ignition. The data is maintained by configuring (adding, deleting, updating) families in Ignition Designer.

Column	Description	Datatype
FamilyId	A unique system generated id.	Integer, PK
ApplicationId	The application that the family belongs to.	Integer, FK,UK1
FamilyName	A unique name for the family.	Varchar(250), UK1
FamilyPriority	The priority of this family, the higher the number the more important the family.	Float, NN
Description	Option description of the family	Varchar(2000)

5.2.3 DtFinalDiagnosis Table

This table defines a final diagnosis

Column	Description	Datatype
FinalDiagnosisId	A unique system generated id.	Integer, PK
FinalDiagnosisName	A unique name	Varchar(250), UK1
FamilyId	The id of the family that this diagnosis belongs to.	Integer, FK, UK1
FinalDiagnosisPriority	The priority of this final diagnosis, the higher the number the more important the final diagnosis	Float, NN
CalculationMethod	The name of the Python calculation method, including the full path.	Varchar(1000)

Column	Description	Datatype
Constant	Flag that indicates if this is a plant status type of Final diagnosis which does not make a recommendation but rather serves to block all lower priority diagnosis in the family.	Bit, NN
PostTextRecommendation	Not sure how this is used	Bit, NN
PostProcessingCallback	The name of the Python calculation method, including the full path, that will be called after an action (Download, NoDownload) is taken by the operator for numeric recommendations or when the operator acknowledges a text recommendation. This will normally be NULL. (This is a new capability for numeric recommendations and duplicates the functionality specified by PostTextRecommendation for text recommendations.	Varchar(1000)
RefreshRate	Interval in seconds that the recommendations will be automatically refreshed	Int, NN
TextRecommendation	Text that will be used for the recommendation when the textRecommendationCallback is null.	Varchar(1000)
State	??	Bit, NN
Active	Flag used by the recommendation manager to indicate that this final diagnosis is currently in the list of highest priorities and is being acted upon.	Bit, NN
Explanation		Varchar(1000)
TrapInsignificantRecommendations	Specifies if a very small recommendation should be displayed in the setpoint spreadsheet that is displayed to the operator.	Bit
LastRecommendationTime	The time that the last recommendation was made for this Final Diagnosis	Datetime

Column	Description	Datatype
TimeOfMostRecentRecommendationImplementation	The time that a recommendation was last acted upon (downloaded or not downloaded). This was changed to Not Null to avoid a problem for a query tag that is an input to a diagnostic diagram where a NULL value could cause problems. The default value is the current time. This should only have an effect when a new final diagnosis is created. The default value is curdate().	Datetime, NN
FinalDiagnosisUUID	The UUID of the final diagnosis.	Varchar(100)
DiagramUUID	The UUID of the diagram containing the final diagnosis.	Varchar(100)

5.2.4 DtSQCDiagnosis Table

This table defines a SQC diagnosis.

Column	Description	Datatype
SQCDiagnosisId	A unique system generated id.	Integer, PK
SQCDiagnosisName	A unique name	Varchar(250), UK1
Status	The current status of the SQC diagnosis. Possible values are Active, Inactive, Unknown	Varchar(50), NN
FamilyId	The id of the family that this diagnosis belongs to.	Integer, FK, UK1
SQCDiagnosisUUID	The id of the Block in the Block Language Toolkit. This is used as part of the SQC plotting utility where the SQC diagnosis is the entry point to the utility and then the parameters and setting from the upstream SQC observation blocks are discovered.	Varchar(50)
DiagramUUID		Varchar(100)
LastResetTime	The time this SQC diagnosis reset time.	Datetime

5.2.5 DtQuantOutput Table

This table defines a Quant Output.

Column	Description	Datatype
QuantOutputId	System defined primary key	Integer, PK
QuantOutputName	A unique name for this Quant Output	Varchar(1000), UK1, NN
ApplicationId	Id of the application that contains this quant output. This is used to populate the list of available QuantOutputs when specifying the Quant Outputs touched by a Final Diagnosis	Integer, FK, UK1
TagPath	The full path to the OPC tag governed by this Quant Output. The tag provider must be included in the path.	Varchar(1000), NN
MostNegativeIncrement	Self-explanatory	Float, NN
MostPositiveIncrement	Self-explanatory	Float, NN
IgnoreMinimumIncrement	A flag that is set by certain diagnosis to bypass the minimum increments that generally apply. This is automatically set and reset and is not exposed through the user interface. This was implemented to replace a specific need at Vistalon which used a procedure, bypass-output-limits(), which saved the limits, set them to 0.0, and then restored the original limits.	Bit, NN
MinimumIncrement	Self-explanatory	Float, NN
SetpointHighLimit	Self-explanatory	Float, NN
SetpointLowLimit	Self-explanatory	Float, NN
FeedbackMethodId	Method to use when multiple moves are recommended for the same output. Legal choices are: Most Positive, Most Negative, Average, or Simple sum. This is not case sensitive.	Varchar(50), NN
IncrementalOutput	Specifies if the calculated recommendations will be treated as Absolute or Incremental changes.	Bit, NN

Column	Description	Datatype
OutputLimitedStatus	Automatically set by the recommendation engine.	Varchar(50)
OutputLimited	Automatically set by the recommendation engine. If True then the output has been limited.	Bit
OutputPercent	The percent of the recommended move that will be used. The percent will be less than 100% if the output is bound or if another output is bound and vector clams are enabled. Automatically set by the recommendation engine.	Float
FeedbackOutput	The raw output from the recommendations. Automatically set by the recommendation engine.	Float
FeedbackOutputManual	A manually entered output value. Automatically set by the recommendation engine.	Float
FeedbackOutputConditioned	The final validated output value. Automatically set by the recommendation engine.	float
ManualOverride	Automatically set by the recommendation engine.	Bit
Active	Automatically set by the recommendation engine.	Bit
DownloadAction		Varchar(25)
DownloadStatus		Varchar(100)
CurrentSetpoint	The current value of the tag at the time the recommendation was made. Automatically set by the recommendation engine.	Float
FinalSetpoint	If Incremental, then the current setpoint + the recommendation, if Absolute, then the recommendation. Automatically set by the recommendation engine.	Float

Column	Description	Datatype
DisplayedRecommendation	Used for display purposes only, this is what is shown in the setpoint spreadsheet. Automatically set by the recommendation engine.	Float

5.2.6 DtRecommendationDefinition Table

This table defines the list of Quant Outputs touched by a final Diagnosis. The data in this table is updated when a final diagnosis is edited in Ignition Designer.

Column	Description	Datatype
RecommendationDefinitionId	System defined primary key	Integer, PK
FinalDiagnosisId	Identifies a Final Diagnosis	Integer, FK, NN
QuantOutputId	Identifies a Quant Output	Integer, FK, NN

5.2.7 DtDiagnosisEntry Table

This table defines a dynamic diagnosis entry. A record is inserted every time a Final Diagnosis becomes true. The contents of this table are displayed in the diagnosis queue.

Column	Description	Datatype
DiagnosisEntryId	System defined primary key	Integer, PK
FinalDiagnosisId	Id of the final diagnosis.	Integer, FK
Status	The status of the entry	Varchar(50), NN
Timestamp	Timestamp when the record was created, which is the same as when the final diagnosis became true.	Datetime, NN
Grade	The grade that was running at the time the final diagnosis became true.	Varchar(50), NN
TextRecommendation	The text describing the diagnosis.	Varchar(1000), NN
RecommendationStatus	The status of a recommendation. Possible values: MADE, NOT-MADE, RESCINDED	Varchar(50), NN

Column	Description	Datatype
Multiplier	Entered from the recommendation map. The multiplier will be applied to all of the recommendations made by the final diagnosis for this diagnosis entry.	Float, NN
RecommendationErrorText	Error description if an error is encountered during automated processing. Not sure if this is used.	Varchar(1000)

5.2.8 DtRecommendation Table

This table defines a recommendation for a specific output in response to a diagnosis entry. Records in this table are inserted and deleted dynamically when the state of Final Diagnosis changes.

Column	Description	Datatype
RecommendationId	System generated primary key	Integer, PK
RecommendationDefinitionId	Identifies the quant output that this recommendation is for.	Integer, FK
DiagnosisEntryId	The Diagnosis Entry that this recommendation was made for.	Integer, FK
Recommendation	The active recommendation, initially the auto recommendation but will be overwritten with the manual recommendation.	Float, NN
AutoRecommendation	The automatically calculated recommendation.	Float, NN
ManualRecommendation	A manually entered recommendation, this is normally NULL until the recommendation is manually edited by the operator.	Float
AutoOrManual	Initially AUTO but updated to MANUAL when a manual recommendation is entered.	Varchar(50), NN

5.2.9 DtTextRecommendation Table

This table defines a text for a text only recommendation.

Column	Description	Datatype
TextRecommendationId	System generated primary key.	Integer, PK
DiagnosisEntryId	The Diagnosis Entry that this recommendation was made for.	Integer, FK
TextRecommendation	The text of a text only recommendation.	Varchar(2500), NN

6 Sequential Control Toolkit Tables and Views

This section describes the tables that pertain to the Sequential Control toolkit.

6.1 General SFC Tables

The tables in this section contain information about the charts and steps in the system. Unless otherwise noted, all of these tables are updated automatically whenever a save is performed from the designer and is implemented in a save hook in our custom SFC module.

6.1.1 SfcChart

This table contains a record for every chart in the system.

Column	Description	Datatype
ChartId	System assigned unique Id	Integer, PK, NN
ChartPath	The full path including name of the chart	Varchar(800), NN
ChartResourceId	Ignition assigned id for this resource. This might be used if a chart is renamed or moved.	Int
CreateTime	Not used	Datetime
IsProduction	Not used	Bit, NN

6.1.2 SfcHierarchy

This table maintains the parent – child relationship between charts. It builds the relationship between encapsulation steps and the charts that they call. It treats unit procedure, operation, and phase steps as extensions of an encapsulation.

Column	Description	Datatype
HierarchyId	System assigned unique Id	Integer, PK, NN
StepId	The id of the encapsulation, unit procedure, operation, or phase step.	Int, FK, NN
ChartId	The id of the chart on which the encapsulation exists.	Int, FK, NN

Column	Description	Datatype
ChildChartId	The id of the chart called by the encapsulation step.	Int, FK, NN

6.1.3 SfcHierarchyHandler

This table maintains additional parent – child relationships between charts when the relationship is determined by stop, abort, or cancel handlers. This is possible by parsing chart references specified in the Python handlers. This will not search through external Python.

Column	Description	Datatype
HierarchyId	System assigned unique Id	Integer, PK, NN
ChartId	The id of the chart on which the handler is defined.	int, FK, NN
Handler	The handler (onAbort, onStop, onCancel)	Varchar(50), NN
HandlerChartId	The id of the chart called by the handler.	Int, NN

6.1.4 SfcNames

This table is only used for debugging purposes and provides the dropdown list on the SFC Runner window with a list of runnable charts. Test charts that need to run from a client should be added to this list. Only the top level charts should be in this table. This table is manually maintained.

Column	Description	Datatype
SfcName	Full chart path and name for the runnable chart.	Varchar(500), PK, NN

6.1.5 SfcRunLog

This table is defines the SFC charts that are currently running.

Column	Description	Datatype
RunId	System assigned unique Id	Integer, PK, NN
ChartPath		Varchar(250), NN
StepName		Varchar(50), NN
StepType	Currently only unit procedures and operations are monitored.	Varchar(50), NN

Column	Description	Datatype
StartTime		Datetime, NN
EndTime		Datetime
Status	Terminal status of the run. Should indicate if it completed normally or was cancelled or aborted.	Varchar(20)
Notes		Varchar(2000)

6.1.6 SfcStep

This table is defines the SFC charts that are currently running.

Column	Description	Datatype
StepId	System assigned unique Id	Integer, PK, NN
StepUUID	This was used in the early stages of development but no longer is since a cloned step does not get a unique UUID	Varchar(256), NN
StepName		Varchar(500), NN
StepTypeId	Id of the type of the step	Int, FK, NN
ChartId	Id of the chart on which this step exists	Int, FK, NN

6.1.7 SfcStepType

This table is defines the SFC charts that are currently running.

Column	Description	Datatype
StepTypeId	System assigned unique Id	Integer, PK, NN
StepType	Name of the step type.	Varchar(50), NN
FactoryId	Java class for this type of step. Useful when a new step is encountered but otherwise not used.	Varchar(50), NN

6.2 SFC Window and Client Support Tables

The following tables support the user interface between a running chart and interested clients. The data in all of these tables is transient. It is created when a chart / step runs and is deleted as the step / chart completes.

6.2.1 SfcControlPanel

This table contains the messages sent to the SFC control panels.

Column	Description	Datatype
ControlPanelId	System assigned Id	PK, int, NN
ControlPanelName		Varchar(900), NN
PostId	Id of the post that this console is pertinent to.	Varchar(256), NN
ChartPath	Path of the chart that will be started from the control panel.	Varchar(900), NN
ChartRunId	The id of the running chart that was launched	Varchar(900)
Operation	The name of the current operation that is running. This may change as the unit procedure runs.	Varchar(900)
MsgQueue	The name of the message queue for the current running chart	Varchar(900)
Originator	The username of the client that started the chart	Varchar(900)
Project	Name of the project that launched the SFC. Even though the SFC is global and runs in the gateway, it displays GUIs to a one or more clients in a specific project	Varchar(900)
IsolationMode	Mode that the chart is running in. If a client switches to isolation mode then it must match this.	Bit
EnableCancel	Used to drive the state of control buttons on the console.	Bit
EnablePause	Used to drive the state of control buttons on the console.	Bit
EnableReset	Used to drive the state of control buttons on the console.	Bit
EnableResume	Used to drive the state of control buttons on the console.	Bit
EnableStart	Used to drive the state of control buttons on the console.	Bit

6.2.2 SfcControlPanelMessage

This table contains the messages sent to the SFC control panels.

Column	Description	Datatype
id	System assigned unique id	PK, int, NN
controlPanelId	Id of the control panel that this message will be displayed in	FK, int, NN
message	The message to be displayed.	Varchar(256), NN
priority	Determines the appearance of the message in the control panel. The supported priorities are: Info, Warning, and Error.	Varchar(20), NN
createTime	The time the message was created. This will be added to the beginning of the message.	Datetime, NN
ackRequired	If true, then the SFC will halt execution until the message is acknowledged. If true, the message will blink in the control panel.	Bit, NN

6.2.3 SfcDialogMessage

This table is used to post a notification window to the client.

Column	Description	Datatype
windowId		Integer, PK, NN
Message	The message to display.	Varchar(900), NN
ackRequired	Species if the message must be acknowledged before execution can proceed.	bit, NN
Acknowledged	True when the message has been acknowledged.	Bit

6.2.4 SfcDownloadGUI

This table contains one record (the master) for the SFC Download GUI.

Column	Description	Datatype
WindowId		Integer, PK, FK, NN
State		Varchar(25), NN

Column	Description	Datatype
TimerRecipeDataId	Recipe id of the timer used for this download.	int, NN
LastUpdated		Varchar(16), NN
StartTime		Datetime

6.2.5 SfcDownloadGUITable

This table contains a record (the detail) for every row in the table. See above for more details

Column	Description	Datatype
WindowId		Integer, PK, FK, NN
RecipeDataId	Recipe id of the timer used for this download.	Varchar(25), NN
RecipeDataType		
LabelAttribute	Determines what will be shown in the label column. Choices are: ??	int, NN
RawTiming		Varchar(16), NN
Timing		Datetime
DcsTagId		Varchar(900)
SetPoint	The target value or the value that will be written.	Varchar(50)
Description	The description of the IO	Varchar(900)
StepTimestamp	The actual time of when the IO was downloaded	Varchar(900)
PV	The current value for the IO being monitored.	Varchar(50)
DownloadStatus	Indicates if the IO has been downloaded and if it was successful	Varchar(900)
PVMonitorStatus	The status of the PV with respect to the SP.	Varchar(900)
SetpointStatus	The oveall status of the row (I'm not sure about this)	Varchar(900)

6.2.6 SfcInput

This table is defines the SFC charts that are currently running.

Column	Description	Datatype
windowId	The id of the window.	Integer, PK, FK, NN
Prompt		Varchar(64), NN
lowLimit	For validation of numeric input	float
highLimit	For validation of numeric input	float
targetStepId	The location where the input value will be stored	Int
keyAndAttribute	The location where the input value will be stored	Varchar(255)
defaultValue	Optional	Varchar(255)

6.2.7 SfcManualDataEntry

This table is defines the SFC charts that are currently running.

Column	Description	Datatype
userName		Integer, PK, NN
chartName		Varchar(64), NN
chartRunId		Varchar(64), NN
Status		Varchar(16), NN
Operation		Varchar(64)
startTime		Datetime
lastChangeTime		Datetime

6.2.8 SfcManualDataEntryTable

This table is defines the SFC charts that are currently running.

Column	Description	Datatype
userName		Integer, PK, NN

Column	Description	Datatype
chartName		Varchar(64), NN
chartRunId		Varchar(64), NN
Status		Varchar(16), NN
Operation		Varchar(64)
startTime		Datetime
lastChangeTime		Datetime

6.2.9 SfcReviewData

This table is defines the SFC charts that are currently running.

Column	Description	Datatype
userName		Integer, PK, NN
chartName		Varchar(64), NN
chartRunId		Varchar(64), NN
Status		Varchar(16), NN
Operation		Varchar(64)
startTime		Datetime
lastChangeTime		Datetime

6.2.10 SfcReviewDataTable

This table is defines the SFC charts that are currently running.

Column	Description	Datatype
userName		Integer, PK, NN
chartName		Varchar(64), NN
chartRunId		Varchar(64), NN
Status		Varchar(16), NN
Operation		Varchar(64)

Column	Description	Datatype
startTime		Datetime
lastChangeTime		Datetime

6.2.11SfcReviewFlows

This table is defines the SFC charts that are currently running.

Column	Description	Datatype
userName		Integer, PK, NN
chartName		Varchar(64), NN
chartRunId		Varchar(64), NN
Status		Varchar(16), NN
Operation		Varchar(64)
startTime		Datetime
lastChangeTime		Datetime

6.2.12SfcReviewFlowsTable

This table is defines the SFC charts that are currently running.

Column	Description	Datatype
userName		Integer, PK, NN
chartName		Varchar(64), NN
chartRunId		Varchar(64), NN
Status		Varchar(16), NN
Operation		Varchar(64)
startTime		Datetime
lastChangeTime		Datetime

6.2.13SfcSaveData

This table is defines the SFC charts that are currently running.

Column	Description	Datatype
userName		Integer, PK, NN
chartName		Varchar(64), NN
chartRunId		Varchar(64), NN
Status		Varchar(16), NN
Operation		Varchar(64)
startTime		Datetime
lastChangeTime		Datetime

6.2.14SfcSelectInput

This table is defines the SFC charts that are currently running.

Column	Description	Datatype
userName		Integer, PK, NN
chartName		Varchar(64), NN
chartRunId		Varchar(64), NN
Status		Varchar(16), NN
Operation		Varchar(64)
startTime		Datetime
lastChangeTime		Datetime

6.2.15SfcTimeDelayNotification

This table is defines the SFC charts that are currently running.

Column	Description	Datatype
userName		Integer, PK, NN
chartName		Varchar(64), NN

Column	Description	Datatype
chartRunId		Varchar(64), NN
Status		Varchar(16), NN
Operation		Varchar(64)
startTime		Datetime
lastChangeTime		Datetime

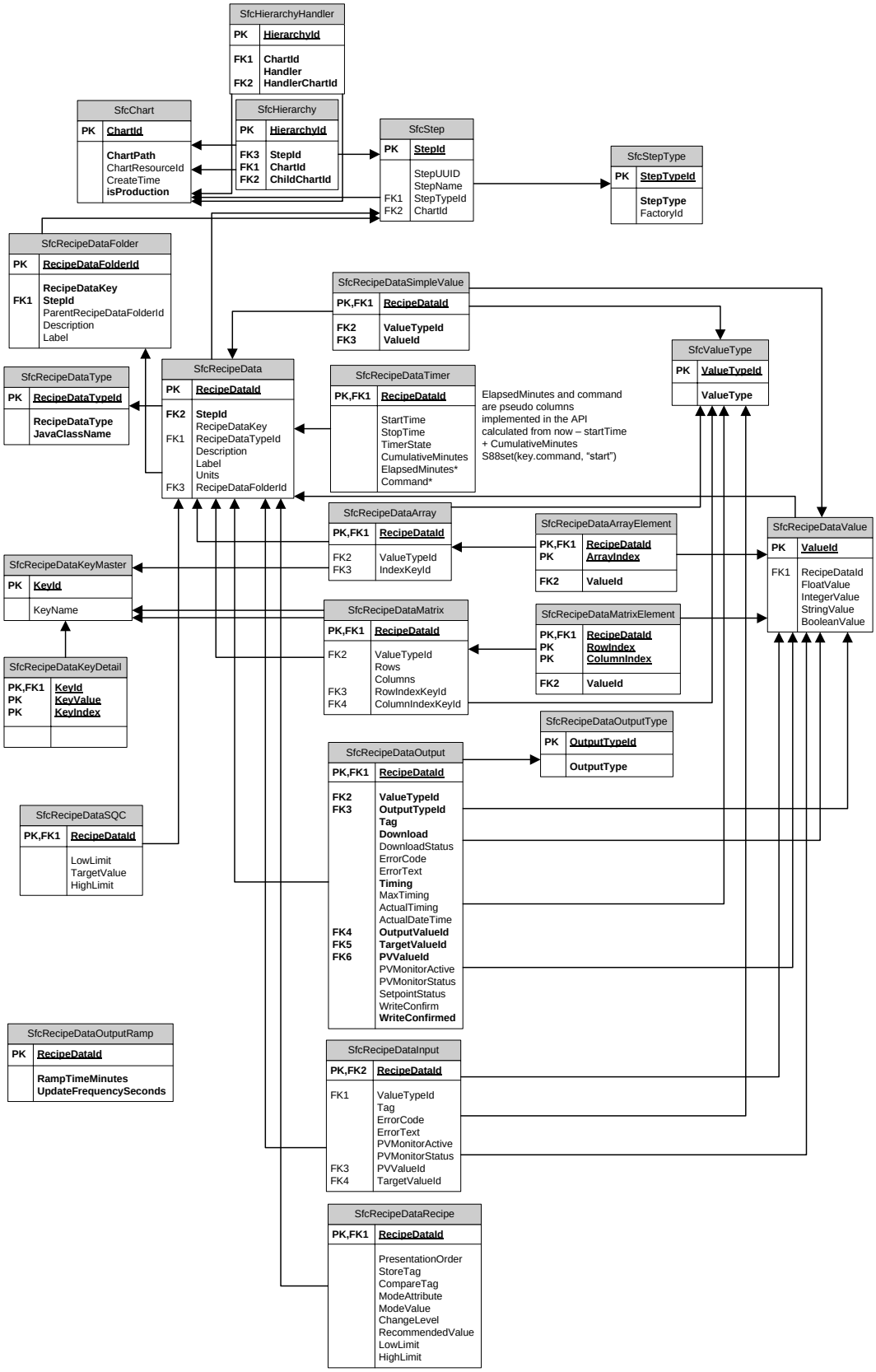
6.2.16SfcWindow

This table is defines the SFC charts that are currently running.

Column	Description	Datatype
userName		Integer, PK, NN
chartName		Varchar(64), NN
chartRunId		Varchar(64), NN
Status		Varchar(16), NN
Operation		Varchar(64)
startTime		Datetime
lastChangeTime		Datetime

6.3 SFC Recipe Data Tables

6.3.1 Entity Relationship Diagram


SfcHierarchyHandler
PK HierarchyId
FK1 ChartId
Handler
FK2 HandlerChartId
SfcChart
PK ChartId
ChartPath
ChartResourceId
CreateTime
isProduction
SfcHierarchy
PK HierarchyId
FK3 StepId
FK1 ChartId
FK2 ChildChartId
SfcStep
PK StepId
StepUUID
StepName
FK1 StepTypeId
FK2 ChartId
SfcStepType
PK StepTypeId
StepType
FactoryId
SfcRecipeDataFolder
PK RecipeDataFolderId
FK1 RecipeDataKey
StepId
ParentRecipeDataFolderId
Description
Label
SfcRecipeDataSimpleValue
PK,FK1 RecipeDataId
FK2 ValueTypeId
FK3 ValueId
SfcValueType
PK ValueTypeId
ValueType
SfcRecipeDataType
PK RecipeDataTypeId
RecipeDataType
JavaClassName
SfcRecipeData
PK RecipeDataId
FK2 StepId
RecipeDataKey
FK1 RecipeDataTypeId
Description
Label
Units
FK3 RecipeDataFolderId
SfcRecipeDataTimer
PK,FK1 RecipeDataId
StartTime
StopTime
TimerState
CumulativeMinutes
ElapsedMinutes*
Command*
ElapsedMinutes and command are pseudo columns implemented in the API calculated from now – startTime + CumulativeMinutes
S88set(key.command, "start")
SfcRecipeDataArray
PK,FK1 RecipeDataId
FK2 ValueTypeId
FK3 IndexKeyId
SfcRecipeDataArrayElement
PK,FK1 RecipeDataId
PK ArrayIndex
FK2 ValueId
SfcRecipeDataValue
PK ValueId
FK1 RecipeDataId
FloatValue
IntegerValue
StringValue
BooleanValue
SfcRecipeDataKeyMaster
PK KeyId
KeyName
SfcRecipeDataMatrix
PK,FK1 RecipeDataId
FK2 ValueTypeId
Rows
Columns
FK3 RowIndexKeyId
FK4 ColumnIndexKeyId
SfcRecipeDataMatrixElement
PK,FK1 RecipeDataId
PK RowIndex
PK ColumnIndex
FK2 ValueId
SfcRecipeDataKeyDetail
PK,FK1 KeyId
PK KeyValue
PK KeyIndex
SfcRecipeDataOutputType
PK OutputTypeId
OutputType
SfcRecipeDataOutput
PK,FK1 RecipeDataId
FK2 ValueTypeId
FK3 OutputTypeId
Tag
Download
DownloadStatus
ErrorCode
ErrorText
Timing
MaxTiming
ActualTiming
ActualDateTime
FK4 OutputValueId
FK5 TargetValueId
FK6 PVValueId
PVMonitorActive
PVMonitorStatus
SetpointStatus
WriteConfirm
WriteConfirmed
SfcRecipeDataSQC
PK,FK1 RecipeDataId
LowLimit
TargetValue
HighLimit
SfcRecipeDataOutputRamp
PK RecipeDataId
RampTimeMinutes
UpdateFrequencySeconds
SfcRecipeDataInput
PK,FK2 RecipeDataId
FK1 ValueTypeId
Tag
ErrorCode
ErrorText
PVMonitorActive
PVMonitorStatus
FK3 PVValueId
FK4 TargetValueId
SfcRecipeDataRecipe
PK,FK1 RecipeDataId
PresentationOrder
StoreTag
CompareTag
ModeAttribute
ModeValue
ChangeLevel
RecommendedValue
LowLimit
HighLimit

6.3.2 SfcRecipeData

This table is defines the SFC charts that are currently running.

Column	Description	Datatype
userName		Integer, PK, NN
chartName		Varchar(64), NN
chartRunId		Varchar(64), NN
Status		Varchar(16), NN
Operation		Varchar(64)
startTime		Datetime
lastChangeTime		Datetime

6.3.3 SfcRecipeDataArray

This table is defines the SFC charts that are currently running.

Column	Description	Datatype
userName		Integer, PK, NN
chartName		Varchar(64), NN
chartRunId		Varchar(64), NN
Status		Varchar(16), NN
Operation		Varchar(64)
startTime		Datetime
lastChangeTime		Datetime

6.3.4 SfcRecipeDataArrayElement

This table is defines the SFC charts that are currently running.

Column	Description	Datatype
userName		Integer, PK, NN

Column	Description	Datatype
chartName		Varchar(64), NN
chartRunId		Varchar(64), NN
Status		Varchar(16), NN
Operation		Varchar(64)
startTime		Datetime
lastChangeTime		Datetime

6.3.5 SfcRecipeDataFolder

This table is defines the SFC charts that are currently running.

Column	Description	Datatype
userName		Integer, PK, NN
chartName		Varchar(64), NN
chartRunId		Varchar(64), NN
Status		Varchar(16), NN
Operation		Varchar(64)
startTime		Datetime
lastChangeTime		Datetime

6.3.6 SfcRecipeDataInput

This table is defines the SFC charts that are currently running.

Column	Description	Datatype
userName		Integer, PK, NN
chartName		Varchar(64), NN
chartRunId		Varchar(64), NN
Status		Varchar(16), NN

Column	Description	Datatype
Operation		Varchar(64)
startTime		Datetime
lastChangeTime		Datetime

6.3.7 SfcRecipeDataKeyDetail

This table defines the values for a key and maps them to a specific array index. For a specific key the indexes should be 0 based. There is nothing in the table definition that ensures that the key is zero based and uses consecutive values. The validation of the index values is left to the user interface. The table does enforce appropriate uniqueness.

Column	Description	Datatype
KeyId	A foreign key into the master table	Integer, FK, NN
KeyValue	The symbolic name used to access an element of an array.	Varchar(64), NN
KeyIndex	The index to use for this value to access the array or matrix.	Varchar(64), NN

6.3.8 SfcRecipeDataKeyMaster

This table defines the keys that are used for "keyed" array and matrix recipe data. At Vistalon, there is a single key named VISTALON-MONOMER-KEY. The use of recipe data keys allows the elements of various recipe data arrays and matrices to be accessed using symbolic values rather than hard-coded indexes.

Column	Description	Datatype
KeyId	A system defined integer.	Integer, PK, NN
KeyName	The user defined name of the key	Varchar(50), NN

6.3.9 SfcRecipeDataMatrix

This table is defines the SFC charts that are currently running.

Column	Description	Datatype
userName		Integer, PK, NN

Column	Description	Datatype
chartName		Varchar(64), NN
chartRunId		Varchar(64), NN
Status		Varchar(16), NN
Operation		Varchar(64)
startTime		Datetime
lastChangeTime		Datetime

6.3.10SfcRecipeDataMatrixElement

This table is defines the SFC charts that are currently running.

Column	Description	Datatype
userName		Integer, PK, NN
chartName		Varchar(64), NN
chartRunId		Varchar(64), NN
Status		Varchar(16), NN
Operation		Varchar(64)
startTime		Datetime
lastChangeTime		Datetime

6.3.11SfcRecipeDataOutput

This table is defines the SFC charts that are currently running.

Column	Description	Datatype
userName		Integer, PK, NN
chartName		Varchar(64), NN
chartRunId		Varchar(64), NN
Status		Varchar(16), NN

Column	Description	Datatype
Operation		Varchar(64)
startTime		Datetime
lastChangeTime		Datetime

6.3.12SfcRecipeDataOutputRamp

This table is defines the SFC charts that are currently running.

Column	Description	Datatype
userName		Integer, PK, NN
chartName		Varchar(64), NN
chartRunId		Varchar(64), NN
Status		Varchar(16), NN
Operation		Varchar(64)
startTime		Datetime
lastChangeTime		Datetime

6.3.13SfcRecipeDataOutputType

This table is defines the SFC charts that are currently running.

Column	Description	Datatype
userName		Integer, PK, NN
chartName		Varchar(64), NN
chartRunId		Varchar(64), NN
Status		Varchar(16), NN
Operation		Varchar(64)
startTime		Datetime
lastChangeTime		Datetime

6.3.14SfcRecipeDataRecipe

This table is defines the SFC charts that are currently running.

Column	Description	Datatype
userName		Integer, PK, NN
chartName		Varchar(64), NN
chartRunId		Varchar(64), NN
Status		Varchar(16), NN
Operation		Varchar(64)
startTime		Datetime
lastChangeTime		Datetime

6.3.15SfcRecipeDataSimpleValue

This table is defines the SFC charts that are currently running.

Column	Description	Datatype
userName		Integer, PK, NN
chartName		Varchar(64), NN
chartRunId		Varchar(64), NN
Status		Varchar(16), NN
Operation		Varchar(64)
startTime		Datetime
lastChangeTime		Datetime

6.3.16SfcRecipeDataSQC

This table is defines the SFC charts that are currently running.

Column	Description	Datatype

Column	Description	Datatype
userName		Integer, PK, NN
chartName		Varchar(64), NN
chartRunId		Varchar(64), NN
Status		Varchar(16), NN
Operation		Varchar(64)
startTime		Datetime
lastChangeTime		Datetime

6.3.17SfcRecipeDataStash

This table is defines the SFC charts that are currently running. *** I DON't THINK THIS IS USED ***

Column	Description	Datatype
userName		Integer, PK, NN
chartName		Varchar(64), NN
chartRunId		Varchar(64), NN
Status		Varchar(16), NN
Operation		Varchar(64)
startTime		Datetime
lastChangeTime		Datetime

6.3.18SfcRecipeDataTimer

This table is defines the SFC charts that are currently running.

Column	Description	Datatype
userName		Integer, PK, NN
chartName		Varchar(64), NN

Column	Description	Datatype
chartRunId		Varchar(64), NN
Status		Varchar(16), NN
Operation		Varchar(64)
startTime		Datetime
lastChangeTime		Datetime

6.3.19SfcRecipeDataType

This table is defines the SFC charts that are currently running.

Column	Description	Datatype
userName		Integer, PK, NN
chartName		Varchar(64), NN
chartRunId		Varchar(64), NN
Status		Varchar(16), NN
Operation		Varchar(64)
startTime		Datetime
lastChangeTime		Datetime

6.3.20SfcRecipeDataValue

This table is defines the SFC charts that are currently running.

Column	Description	Datatype
userName		Integer, PK, NN
chartName		Varchar(64), NN
chartRunId		Varchar(64), NN
Status		Varchar(16), NN
Operation		Varchar(64)

Column	Description	Datatype
startTime		Datetime
lastChangeTime		Datetime

6.3.21SfcValueType

This table is defines the SFC charts that are currently running.

Column	Description	Datatype
userName		Integer, PK, NN
chartName		Varchar(64), NN
chartRunId		Varchar(64), NN
Status		Varchar(16), NN
Operation		Varchar(64)
startTime		Datetime
lastChangeTime		Datetime