PViMS

Database Manual

July 2018

USAID
FROM THE AMERICAN PEOPLE

SIAPS
Systems for Improved Access to Pharmaceuticals and Services

This report is made possible by the generous support of the American people through the US Agency for International Development (USAID), under the terms of cooperative agreement number AID-OAA-A-11-00021. The contents are the responsibility of Management Sciences for Health and do not necessarily reflect the views of USAID or the United States Government.

About SIAPS

The goal of the Systems for Improved Access to Pharmaceuticals and Services (SIAPS) Program is to assure the availability of quality pharmaceutical products and effective pharmaceutical services to achieve desired health outcomes. Toward this end, the SIAPS result areas include improving governance, building capacity for pharmaceutical management and services, addressing information needed for decision-making in the pharmaceutical sector, strengthening financing strategies and mechanisms to improve access to medicines, and increasing quality pharmaceutical services.

Recommended Citation

This report may be reproduced if credit is given to SIAPS. Please use the following citation.

SIAPS Program. 2018. *PViMS Database Manual.* Submitted to the US Agency for International Development by the Systems for Improved Access to Pharmaceuticals and Services (SIAPS) Program. Arlington, VA: Management Sciences for Health.

Systems for Improved Access to Pharmaceuticals and Services
Pharmaceuticals and Health Technologies Group
Management Sciences for Health
4301 North Fairfax Drive, Suite 400
Arlington, VA 22203 USA
Telephone: 703.524.6575
Fax: 703.524.7898
E-mail: phtmis@msh.org
Website: www.siapsprogram.org

Contents

1 Introduction 7
1.1 Document Overview 7
1.2 Purpose of the Document 7
1.3 Audience 7
2 Database Entities 8
2.1 Work Plan Entities 8
2.1.1 Entity Relationship Diagram 8
2.1.2 Reference Entities 8
2.1.3 Bridging Entities 10
2.2 Dataset Entities 12
2.2.1 Entity Relationship Diagram 12
2.2.2 Reference Entities 13
2.2.3 Bridging Entities 14
2.2.4 Transaction Entities 17
2.3 Meta Framework Entities 27
2.3.1 Entity Relationship Diagram 27
2.3.2 Reference Entities 28
2.3.3 Transaction Entities 30
2.4 Patient Entities 34
2.4.1 Entity Relationship Diagram 34
2.4.2 Reference Entities 35
2.4.3 Transaction Entities 36
2.5 Encounter Entities 49
2.5.1 Entity Relationship Diagram 49
2.5.2 Transaction Entities 49
2.6 Workflow Entities 54
2.6.1 Entity Relationship Diagram 55
2.6.2 Reference Entities 56

2.6.3 Transaction Entities 58
2.7 Other Entities 62
2.7.1 Reference Entities 62
2.7.2 Custom Attribute Entities 72
2.7.3 Bridging Entities 73
2.7.4 Miscellaneous Entities 74
3 Database Security 79
3.1 Surface Area Reduction 79
3.2 Service Account Selection and Management 79
3.3 SQL Server Best Practises Analyser 80
3.4 Patching and Automated Windows Updates 80
3.5 Encryption 80
3.6 Administrator permissions 80
3.7 Database Ownership 82
3.8 Lockdown of System Stored Procedures 82
3.9 Authorisation 82
3.10 Authentication 83
3.11 Network Security 83
4 Database Maintenance 85
4.1 Database Backups 85
4.1.1 Full Backups 86
4.2 Maintenance Plans 88
4.2.1 Preparing the Maintenance Plan 90
4.2.2 Preparing the Database Backup Task (System Databases) 93
4.2.3 Preparing the Database Backup Task (PViMS Database) 99
4.2.4 Preparing the Database Integrity Task (PViMS Database) 104
4.2.5 Preparing the Index Optimise Task (PViMS Database) 106
5 Database Restore 108

1 Introduction

1.1 Document Overview

This document focuses on the technical aspects of the PViMS database implementation. Due to the use of an Object Relational Mapping tool (Entity Framework), PViMS is effectively database agnostic. However, the focal RDBMS of this implementation is MS SQL Server and as such this document is prepared with SQL Server in mind.

1.2 Purpose of the Document

The purpose of the document is to describe the technical implementation of the PViMS database component. This includes: -

- Entity relationship diagrams per core area
- Data dictionary for all entities
- Description of database security, owner vs user model
- Database maintenance
- Continuity planning

1.3 Audience

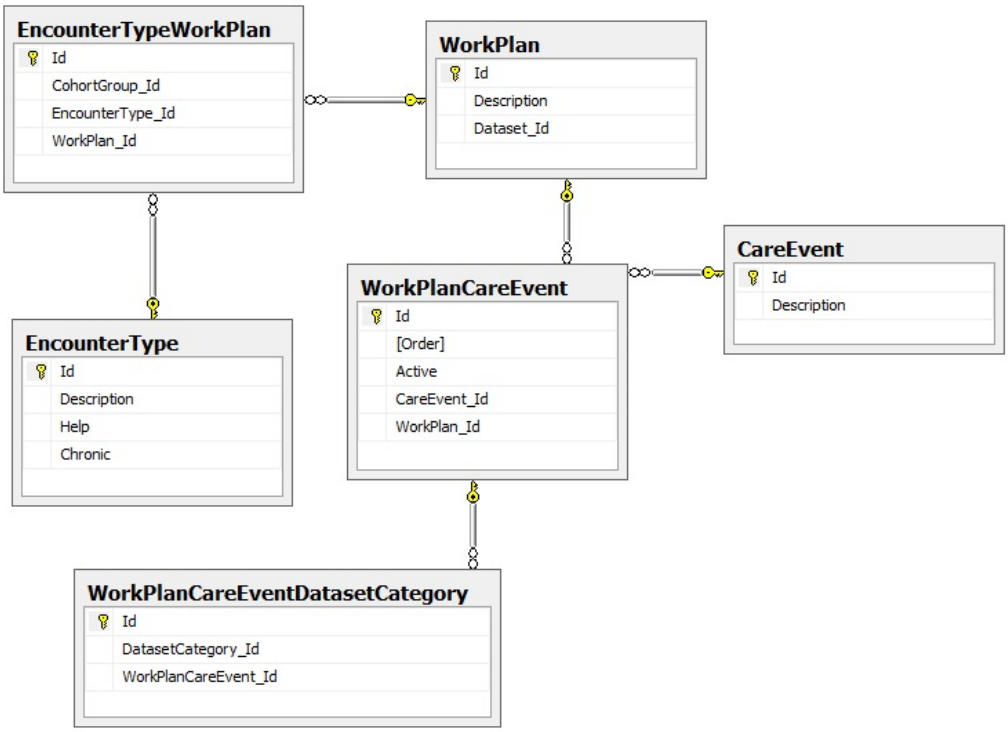
This document is targeted towards database administrators.

2 Database Entities

2.1 Work Plan Entities

This section contains all work plan related entities that facilitate the dynamic customization of clinical dataset elements to be rendered as part of the user interface. Work plans bridge the reason for encounter (encounter type) with the clinical elements needed for collection (dataset element)

2.1.1 Entity Relationship Diagram



2.1.2 Reference Entities

Field Name	Field Description	Data Type	Field Constraints	Primary/ Foreign	Nullable
Entity	CareEvent				
Description	Contains a list of care events. A care event is defined as an interaction between a health professional and a patient.				
Id	Unique auto-incremented seed for table	Integer	None	Primary	No

Description	The name of the care event	Nvarchar	Max length 50	N/A	No
Entity	EncounterType				
Description	Contains a list of encounter types. An encounter type is defined as the primary reason the patient is attending the facility.				
Id	Unique auto-incremented seed for table	Integer	None	Primary	No
Description	The name of the encounter type	Nvarchar	Max length 50	N/A	No
Help	Additional help that supplements the description of the encounter type.	Nvarchar	Max length 250	N/A	Yes
Chronic	Is this a chronic based encounter type? Dataset elements will be rendered based on the configuration of this encounter type.	Bit	N/A	N/A	No
Entity	WorkPlan				
Description	Contains a list of work plans. A work plan can be aligned to an encounter type and is fundamentally the approach taken by a facility to treat a patient. For instance, a **chronic repeat visit** work plan can be designed to treat chronic patients who are visiting the facility to collect medication only. A work plan shares a one to many relationship per care event.				
Id	Unique auto-incremented seed for table	Integer	None	Primary	No
Description	The name of the work plan	Nvarchar	Max length 50	N/A	No
Dataset_Id	The associated dataset which contains all data elements that will be rendered as part of this work plan	Integer	FK to Dataset	Foreign	Yes

2.1.3 Bridging Entities

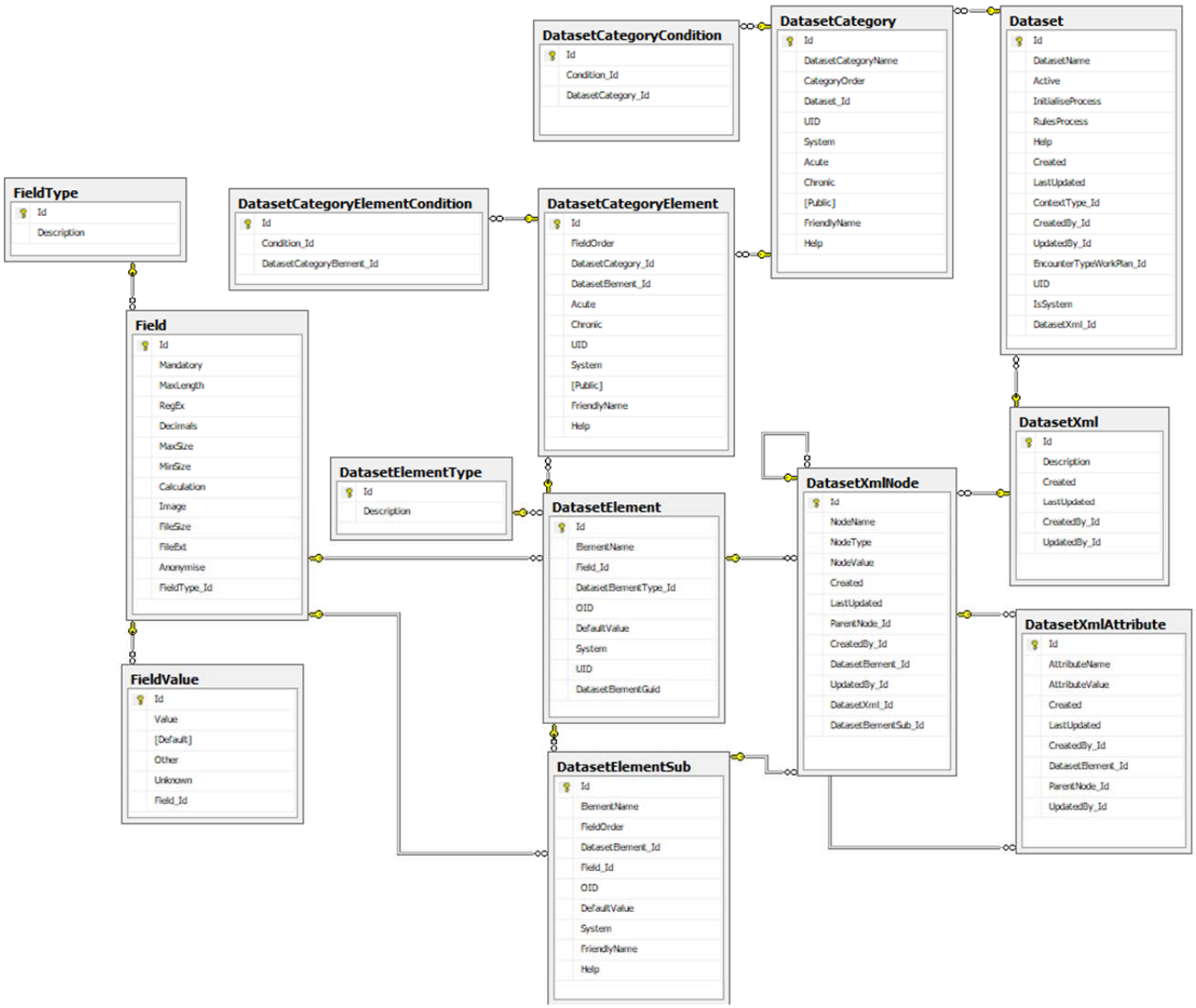
Field Name	Field Description	Data Type	Field Constraints	Primary/ Foreign	Nullable
Entity	EncounterTypeWorkPlan				
Description	Bridging table that links an encounter type to a work plan. In theory more than work plan can be defined per encounter type, but practically the system implements this as a one to one relationship. This is done to maintain simplicity but provides future scalability.				
Id	Unique auto-incremented seed for table	Integer	None	Primary	No
CohortGroup_Id	Is this work plan implementation of an encounter type linked to a cohort study? This work plan will only be activated for patients on a particular study.	Integer	FK to CohortGroup	Foreign	Yes
EncounterType_Id	The encounter type that the work plan is associated to	Integer	FK to EncounterType	Foreign	No
WorkPlan_Id	The work plan that defines the approach to the encounter	Integer	FK to WorkPlan	Foreign	No
Entity	WorkPlanCareEvent				
Description	Bridging table that links a work plan to its underlying care events.				
Id	Unique auto-incremented seed for table	Integer	None	Primary	No
Order	The order of the care event in relation to the work plan. Care events run sequentially. Please note that this order is not currently enforced in PViMS but has been catered for to facilitate future upscaling	Integer		N/A	No
Active	Is this care event currently active	Bit	None	N/A	No
CareEvent_Id	The care event that has been linked to the work plan	Integer	FK to CareEvent	Foreign	No
WorkPlan_Id	The work plan that the care event has been linked to	Integer	FK to WorkPlan	Foreign	No
Entity	WorkPlanCareEventDatasetCategory				
Description	Bridging table that links various dataset categories to a specific work plan care event.				

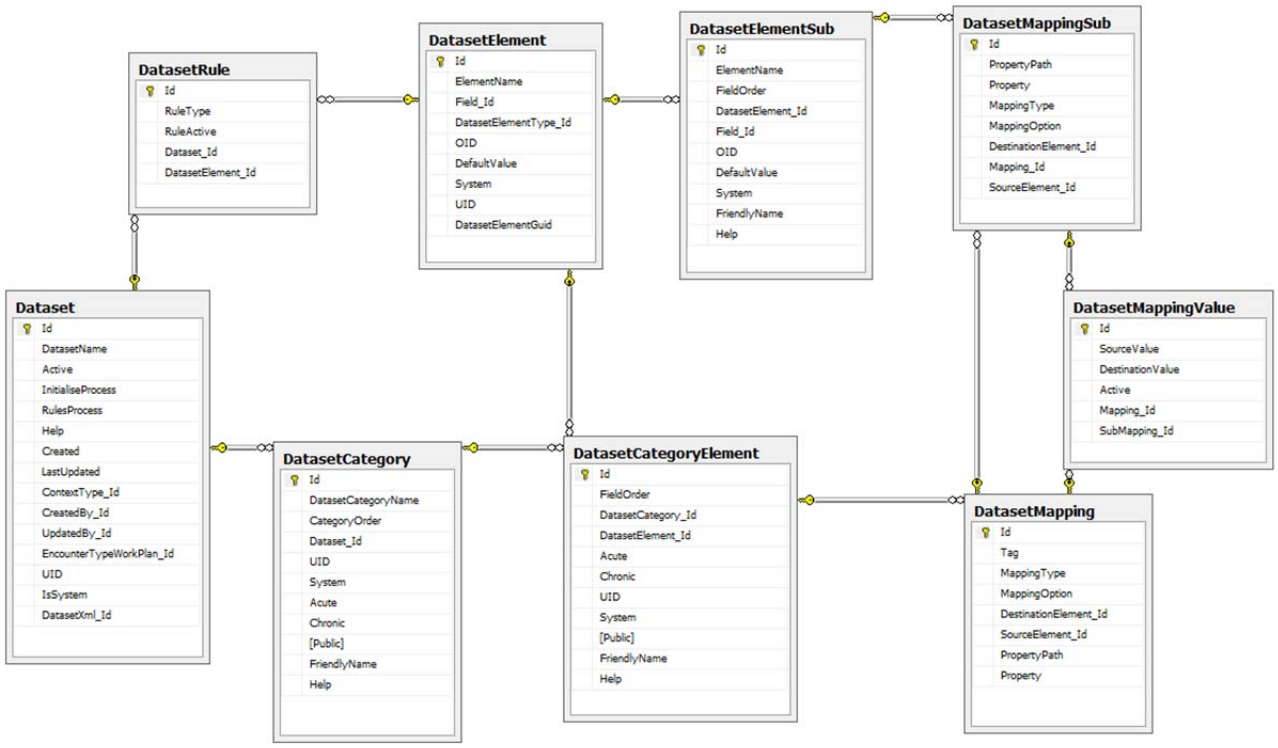
	Functional implementation means these categories would be displayed for data collection depending on the associated care event being enforced				
Id	Unique auto-incremented seed for table	Integer	None	Primary	No
DatasetCategory_Id	The dataset category that has been linked to the work plan care event. The dataset category must be linked to the dataset that has been assigned to the work plan	Integer	FK to DatasetCategory	Foreign	No
WorkPlanCareEvent_Id	The work plan care event that the dataset category has been linked to	Integer	FK to WorkPlanCareEvent	Foreign	No

2.2 Dataset Entities

This section contains all dataset related entities that facilitate the dynamic customization of clinical elements within the system. All clinical data can be distilled and governed on the basis of a dataset element which contains the definition of the element and governs data integrity in based on this definition.

2.2.1 Entity Relationship Diagram





2.2.2 Reference Entities

Field Name	Field Description	Data Type	Field Constraints	Primary/ Foreign	Nullable
Entity	FieldType				
Description	Contains a list of field types that specify what type of data will be stored within the dataset element.				
Id	Unique auto-incremented seed for table	Integer	None	Primary	No
Description	The name of the field type	Nvarchar	Max length 50	N/A	No
	Field Types Listbox DropDownList AlphaNumericTextbox NumericTextbox YesNo Date Table System				

Entity	DatasetElementType				
Description	Contains a list of dataset element types. This will be used for future scalability. Primary definition of a data type is currently defined through the FieldType element. **** Not in use ****				
Id	Unique auto-incremented seed for table	Integer	None	Primary	No
Description	The name of the dataset element type	Nvarchar	Max length 50	N/A	No

2.2.3 Bridging Entities

Field Name	**Field Description**	**Data Type**	**Field Constraints**	**Primary/ Foreign**	**Nullable**
Entity	DatasetCategoryCondition				
Description	Contains a list of chronic conditions that a category has been associated to (relevant if the chronic field has been set to true for the associated category)				
Id	Unique auto-incremented seed for table	Integer	None	Primary	No
Condition_Id	The chronic condition the patient must have for the category to be rendered	Integer	FK to Condition	Foreign	No
DatasetCategory_Id	The dataset category that the chronic condition has been linked to	Integer	FK to DatasetCategory	Foreign	No
Entity	DatasetCategoryElement				
Description	Contains a list of elements associated to a category				
Id	Unique auto-incremented seed for table	Integer	None	Primary	No
FieldOrder	The order of the element in relation to the category	Smallint	None	N/A	No
DatasetCategory_Id	The dataset category that the element has been linked to. Elements can be linked to more than one category in more than one dataset	Integer	FK to DatasetCategory	Foreign	No
DatasetElement_Id	The element that is being	Integer	FK to DatasetElement	Foreign	No

	linked to the dataset category				
Acute	Display element for acute encounter types (non-chronic)	Bit	None	N/A	No
Chronic	Display this element for chronic encounter types	Bit	None	N/A	No
UID	A unique ID for the dataset element **** not in use ****	Nvarchar	Max length 10	N/A	Yes
System	Is the dataset element system generated? System defined elements may not be removed.	Bit		N/A	No
Public	Collected in public mode only (not logged into PViMS) **** not in use ****	Bit		N/A	No
FriendlyName	The friendly name of the dataset element when rendered in the current category	Nvarchar	Max length 150	N/A	Yes
Help	Additional information that defines the use of the dataset element when rendered in the current category	Nvarchar	Max length 350	N/A	Yes
Entity	DatasetCategoryElementCondition				
Description	Contains a list of chronic conditions that an element has been associated to (relevant if the chronic field has been set to true for the associated element)				
Id	Unique auto-incremented seed for table	Integer	None	Primary	No
Condition_Id	The chronic condition the patient must have for the element to be rendered	Integer	None	N/A	No
DatasetCategoryElement_Id	The dataset category element that the chronic condition has been linked	Integer	FK to DatasetCategoryElement	Foreign	No

	to				

2.2.4 Transaction Entities

Field Name	Field Description	Data Type	Field Constraints	Primary / Foreign	Nullable
Entity	Dataset				
Description	Contains a list of datasets. A dataset is effectively a collection of dataset categories which are further sub divided into a collection of dataset elements per category. A dataset is effectively a collection of dataset elements that correlate to each other.				
Id	Unique auto-incremented seed for table	Integer	None	Primary	No
DatasetName	The name of the dataset	Nvarchar	Max length 50	N/A	No
Active	Is this dataset currently active for data collection	Bit	None	N/A	No
InitialiseProcess	A C# method that can be called to initialize the dataset on creation **** not in use ****	Nvarchar	Max length 100	N/A	Yes
RulesProcess	A C# method that can be called to instantiate business rules against the dataset **** not in use ****	Nvarchar	Max length 100	N/A	Yes
Help	Additional information that defines the use of the dataset	Nvarchar	Max length 250	N/A	Yes
Created	AUDIT: Date record was created	Datetime	None	N/A	No
LastUpdated	AUDIT: Date record was last updated	Datetime	None	N/A	Yes
ContextType_Id	The type of entity that this dataset associated to	Integer	FK to ContextType	Foreign	Yes
	Context Types Encounter Patient Pregnancy Global PatientClinicalEvent DatasetInstance				
CreatedBy_Id	AUDIT: The user that created this record	Integer	FK to User	Foreign	Yes
UpdatedBy_Id	AUDIT: The user that last updated	Integer	FK to User	Foreign	Yes

	this record				
EncounterTypeWorkPlan_Id	The work plan that has been used to implement this dataset	Integer	FK to EncounterTypeWorkPlan	Foreign	Yes
UID	A unique ID for the dataset **** not in use ****	Nvarchar	Max length 10	N/A	Yes
IsSystem	Is the dataset system generated? System defined datasets may not be renamed or removed.	Bit		N/A	No
DatasetXml_Id	The XML dataset structure this dataset is linked to. E2B R2 is defined through this XML dataset structure.	Integer	FK to DatasetXml	Foreign	Yes
Entity	DatasetCategory				
Description	Contains a list of categories associated to the dataset. A dataset category is effectively made up of a collection of dataset elements that are related to each other as defined by the category itself. For instance, a **Vitals** category would contain all elements related to vitals, such as blood pressure, weight etc.				
Id	Unique auto-incremented seed for table	Integer	None	Primary	No
DatasetCategoryName	The name of the category	Nvarchar	Max length 50	N/A	No
CategoryOrder	The order of the category in relation to the dataset	Smallint	None	N/A	No
Dataset_Id	The dataset the category is associated to	Integer	FK to Dataset	Foreign	No
UID	A unique ID for the dataset category **** not in use ****	Nvarchar	Max length 10	N/A	Yes
System	Is the dataset category system generated? System defined categories may not be renamed or removed.	Bit		N/A	No
Acute	Render the category for all non-chronic related patients	Bit		N/A	No
Chronic	Render the category for all chronic related patients. For instance, this category will only be rendered if the patient has malaria.	Bit		N/A	No

Public	Collected in public mode only (not logged into PViMS) **** not in use ****	Bit		N/A	No
Dataset_Id	The dataset the category is associated to	Integer	FK to Dataset	Foreign	No
FriendlyName	The friendly name of the dataset category	Nvarchar	Max length 150	N/A	Yes
Help	Additional information that defines the use of the dataset category	Nvarchar	Max length 350	N/A	Yes
Entity	DatasetElement				
Description	Contains a list of unique dataset elements defined within the system.				
Id	Unique auto-incremented seed for table	Integer	None	Primary	No
ElementName	The unique name of the dataset element (e.g. Weight)	Nvarchar	Max length 100	N/A	No
Field_Id	The field definition for this dataset element. Contains rules that are implemented to ensure data integrity	Integer	FK to Field	Foreign	No
DatasetElementType_Id	The dataset element type **** not in use ****	Integer	FK to DatasetElementType	Foreign	Yes
OID	The OID for the dataset element. Used for E2B extracts.	Nvarchar	Max length 50	N/A	Yes
DefaultValue	The default value for an OID field in an E2B extract.	Nvarchar	Max length MAX	N/A	Yes
System	Is this a system defined element? System defined elements may not be removed or renamed.	Bit	None	N/A	No
UID	A unique ID for the dataset element **** not in use ****	Nvarchar	Max length 10	N/A	Yes
DatasetElementGuid	A globally unique definition for the element. Used to match an element when called through the interoperability end point	Uniqueidentifier	None	N/A	No
Entity	DatasetElementSub				

Description	Contains a list of unique dataset elements for the **TABLE** field type				
Id	Unique auto-incremented seed for table	Integer	None	Primary	No
ElementName	The unique name of the dataset element (e.g. Weight)	Nvarchar	Max length 100	N/A	No
FieldOrder	The order of the element in relation to the table as a whole	Smallint	None	N/A	No
DatasetElement_Id	The dataset element that the sub element is linked to	Integer	FK to DatasetElement	Foreign	No
Field_Id	The field definition for this dataset element. Contains rules that are implemented to ensure data integrity	Integer	FK to Field	Foreign	No
OID	The OID for the dataset element. Used for E2B extracts.	Nvarchar	Max length 50	N/A	Yes
DefaultValue	The default value for an OID field in an E2B extract.	Nvarchar	Max length MAX	N/A	Yes
System	Is this a system defined element? System defined elements may not be removed or renamed.	Bit	None	N/A	No
FriendlyName	The friendly name of the dataset element	Nvarchar	Max length 150	N/A	Yes
Help	Additional information that defines the use of the dataset element within the table	Nvarchar	Max length 350	N/A	Yes
Entity	DatasetMapping				
Description	Contains a list of mappings that maps one dataset element within one dataset to another dataset element within another dataset. These mappings are used to automate the defaulting of the E2B dataset on creation.				
Id	Unique auto-incremented seed for table	Integer	None	Primary	No
Tag	Active for active reporting and Spontaneous for spontaneous reporting	Nvarchar	Max length 350	N/A	Yes
MappingType	The type of mapping • ElementToElement	Integer		N/A	No

	• ValueToValue • AttributeToElement • AttributeToValue • FirstClassToElement • FirstClassToValue				
MappingOption	Additional formatting options when mapping takes place • yyyyMMdd for date fields	Nvarchar	Max length MAX	N/A	Yes
DestinationElement_Id	The dataset category element that will receive the translated value	Integer	FK to DatasetCategoryElement	Foreign	No
SourceElement_Id	The dataset category element that will submit the translated value	Integer	FK to DatasetCategoryElement	Foreign	Yes
PropertyPath	If a source element is not specified, then we need to reference a custom attribute	Nvarchar	Max length MAX	N/A	Yes
Property	If a source element is not specified, then we need to reference a custom attribute	Nvarchar	Max length MAX	N/A	Yes
Entity	DatasetMappingSub				
Description	Contains a list of mappings that maps one dataset element sub within one dataset mapping to another dataset element sub within another dataset mapping. These mappings are used to automate the defaulting of the E2B dataset on creation.				
Id	Unique auto-incremented seed for table	Integer	None	Primary	No
MappingType	The type of mapping	Integer		N/A	No
	Mapping Types ElementToElement ValueToValue AttributeToElement AttributeToValue FirstClassToElement FirstClassToValue				
MappingOption	Additional formatting options when mapping takes place • yyyyMMdd for date fields	Nvarchar	Max length MAX	N/A	Yes
DestinationElement_Id	The dataset element sub that will	Integer	FK to	Foreign	No

	receive the translated value		DatasetElementSub		
Mapping_Id	The dataset mapping that the sub mapping belongs to	Integer	FK to DatasetMapping	Foreign	No
SourceElement_Id	The dataset element sub that will submit the translated value	Integer	FK to DatasetElementSub	Foreign	Yes
PropertyPath	If a source element is not specified, then we need to reference a custom attribute	Nvarchar	Max length MAX	N/A	Yes
Property	If a source element is not specified, then we need to reference a custom attribute	Nvarchar	Max length MAX	N/A	Yes
Entity	DatasetMappingValue				
Description	Contains a list of mapping values that translate a source value to a destination value. These mappings are used to automate the defaulting of the E2B dataset on creation.				
Id	Unique auto-incremented seed for table	Integer	None	Primary	No
SourceValue	The source value that is to be mapped	Nvarchar	Max length 100	N/A	No
DestinationValue	The destination value that is to be mapped to	Nvarchar	Max length 100	N/A	Yes
Active	Is the mapping currently active	Bit		N/A	No
Mapping_Id	The dataset mapping that the mapping value belongs to	Integer	FK to DatasetMapping	Foreign	No
SubMapping_Id	The dataset sub mapping that the mapping value belongs to	Integer	FK to DatasetMappingSub	Foreign	No
Entity	DatasetRule				
Description	Contains a list of implementable rules per dataset				
Id	Unique auto-incremented seed for table	Integer	None	Primary	No
RuleType	The type of rule to be implemented	Integer		N/A	No
	Rule Types ElementCanoOnlyLinkToSingleDataset MandatoryFieldsProminent				
RuleActive	Is this dataset rule currently active	Bit		N/A	No

Dataset_Id	The dataset that the rule belongs to	Integer	FK to Dataset	Foreign	Yes
DatasetElement_Id	The dataset element that the rule belongs to	Integer	FK to DatasetElement	Foreign	Yes
Entity	DatasetXml				
Description	Contains the description of the XML node set that has been attributed to a corresponding dataset. Used to generate the E2B XML data structure.				
Id	Unique auto-incremented seed for table	Integer	None	Primary	No
Description	The description of the XML data structure. For instance, **E2B(R2) XML.**	Nvarchar	Max length 50	N/A	Yes
Created	AUDIT: Date record was created	Datetime	None	N/A	No
LastUpdated	AUDIT: Date record was last updated	Datetime	None	N/A	Yes
CreatedBy_Id	AUDIT: The user that created this record	Integer	FK to User	Foreign	Yes
UpdatedBy_Id	AUDIT: The user that last updated this record	Integer	FK to User	Foreign	Yes
Entity	DatasetXmlAttribute				
Description	Contains a list of XML attributes that correspond to an XML node. Used to generate the E2B XML data structure.				
Id	Unique auto-incremented seed for table	Integer	None	Primary	No
AttributeName	The name of the XML attribute	Nvarchar	Max length 50	N/A	No
AttributeValue	The default value of the attribute	Nvarchar	Max length MAX	N/A	Yes
Created	AUDIT: Date record was created	Datetime	None	N/A	No
LastUpdated	AUDIT: Date record was last updated	Datetime	None	N/A	Yes
CreatedBy_Id	AUDIT: The user that created this record	Integer	FK to User	Foreign	Yes
DatasetElement_Id	The dataset element that the XML attribute represents. The corresponding value of the element is transposed into the generated XML file.	Integer	FK to DatasetElement	Foreign	Yes

ParentNode_Id	The XML node that the attribute belongs to	Integer	FK to DatasetXmlNode	Foreign	No
UpdatedBy_Id	AUDIT: The user that last updated this record	Integer	FK to User	Foreign	Yes
Entity	DatasetXmlNode				
Description	Contains a list of XML nodes that correspond to an XML dataset. Used to generate the E2B XML data structure.				
Id	Unique auto-incremented seed for table	Integer	None	Primary	No
NodeName	The name of the XML node	Nvarchar	Max length 50	N/A	No
NodeType	The type of XML node	Int		N/A	No
NodeType	**Node Types** RootNode = 1 StandardNode = 2 RepeatingNode = 3	Int		N/A	No
NodeValue	The default value of the node	Nvarchar	Max length MAX	N/A	Yes
Created	AUDIT: Date record was created	Datetime	None	N/A	No
LastUpdated	AUDIT: Date record was last updated	Datetime	None	N/A	Yes
ParentNode_Id	Self-referencing, does this node have a parent node. Used to create tree structure.	Integer	FK to DatasetXmlNode	Foreign	Yes
CreatedBy_Id	AUDIT: The user that created this record	Integer	FK to User	Foreign	Yes
DatasetElement_Id	The dataset element that the XML node represents. The corresponding value of the element is transposed into the generated XML file.	Integer	FK to DatasetElement	Foreign	Yes
UpdatedBy_Id	AUDIT: The user that last updated this record	Integer	FK to User	Foreign	Yes
DatasetXml_Id	The XML dataset that the node belongs to.	Integer	FK to DatasetXml	Foreign	Yes
DatasetElementSub_Id	The dataset element that the XML node represents. The corresponding value of the element is transposed into the	Integer	FK to DatasetElementSub	Foreign	Yes

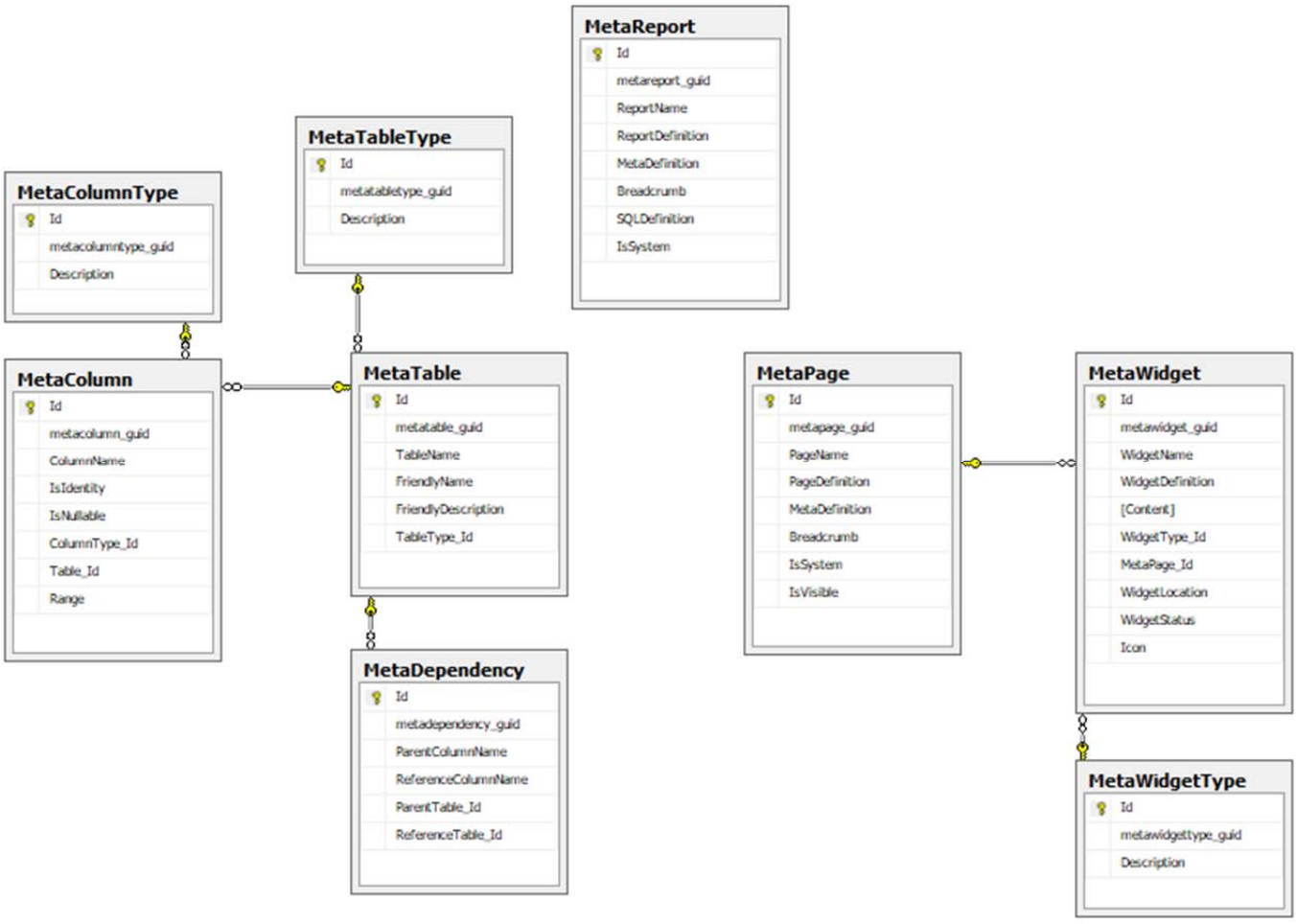
	generated XML file.				
Entity	Field				
Description	Contains the data integrity rules for the associated dataset element				
Id	Unique auto-incremented seed for table	Integer	None	Primary	No
Mandatory	Is the dataset element mandatory? Applicable for all field types.	Bit	None	N/A	No
MaxLength	The maximum length of the element value. Applicable to the AlphaNumericTextbox.	Smallint	None	N/A	Yes
RegEx	Is there a regular expression that should be executed when validating the element? **** not in use ****	Nvarchar	Max length 100	N/A	Yes
Decimals	The number of decimals for a numerical value. Applicable to the NumericTextbox	Smallint	None	N/A	Yes
MaxSize	The maximum value for a numerical value. Applicable to the NumericTextbox	Decimal(18.2)	None	N/A	Yes
MinSize	The minimum value for a numerical value. Applicable to the NumericTextbox	Decimal(18.2)	None	N/A	Yes
Calculation	Element value is defined as per the calculation attribute. For instance, BMI is calculated using the weight and length values.	Nvarchar	Max length 100	N/A	Yes
Image	Clinical value is stored in the form of an image within the database. Can be used to store patient photos etc.	Image	None	N/A	Yes
FileSize	Maximum file size that can be stored.	Smallint	None	N/A	Yes
FileExt	The types of file extensions accepted for storage.	Nvarchar	Max length 100	N/A	Yes
Anonymise	If the element is included in any extract or report, should this field	Bit	None	N/A	No

	be anonymized				
FieldType_Id	The type of field	Integer	FK to FieldType	Foreign	Yes
Entity	FieldValue				
Description	Contains a list of selection values for DropDownList field types				
Id	Unique auto-incremented seed for table	Integer	None	Primary	No
Value	The value of the drop-down list	Nvarchar	Max length 100	N/A	No
Default	Is this the default value for the drop-down list	Bit	None	N/A	No
Other	Drop down list value for OTHER	Bit	None	N/A	No
Unknown	Drop down list value for UNKNOWN	Bit	None	N/A	No
Field_Id	The field that the list of values is linked to	Integer	FK to Field	Foreign	No

2.3 Meta Framework Entities

This section contains all meta related entities that facilitate customization of reports within the reporting portal and content within the information portal.

2.3.1 Entity Relationship Diagram



2.3.2 Reference Entities

Field Name	Field Description	Data Type	Field Constraints	Primary/ Foreign	Nullable
Entity	MetaWidgetType				
Description	Contains a list of meta widget types which defines the type of widgets that can be added to the information portal				
Id	Unique auto-incremented seed for table	Integer	None	Primary	No
Metawidgettype_GUID	A globally unique identifier for this widget type	UniqueIdentifier	None	N/A	No
Description	The name of the meta widget type	Nvarchar	Max length 50	N/A	No
	Meta Widget Types General Wiki ItemList				
Entity	MetaColumnType				
Description	Contains a list of meta column types. A meta column type governs the type of columns that can be defined in the reporting portal				
Id	Unique auto-incremented seed for table	Integer	None	Primary	No
Metacolumntype_GUID	A globally unique identifier for this column type	UniqueIdentifier	None	N/A	No
Description	The name of the meta column type	Nvarchar	Max length 50	N/A	No
	Meta Column Types bigint binary bit char date datetime decimal image int nchar nvarchar smallint				

	time tinyint uniqueidentifier varbinary varchar				
Entity	MetaTableType				
Description	Contains a list of meta table types. A meta table type governs the type of tables that can be defined in the reporting portal				
Id	Unique auto-incremented seed for table	Integer	None	Primary	No
Metacolumntype_GUID	A globally unique identifier for this table type	UniqueIdentifier	None	N/A	No
Description	The name of the meta table type	Nvarchar	Max length 50	N/A	No
	Meta Table Types Core CoreChild Child History Lookup				

2.3.3 Transaction Entities

Field Name	Field Description	Data Type	Field Constraints	Primary/ Foreign	Nullable
Entity	MetaReport				
Description	Contains a list of custom and system defined reports. The reporting portal menu structure is dynamically populated based on the contents of this entity.				
Id	Unique auto-incremented seed for table	Integer	None	Primary	No
Metareport_GUID	A globally unique identifier for this report	UniqueIdentifier	None	N/A	No
ReportName	The name of the report	Nvarchar	Max length 50	N/A	No
ReportDefinition	A detailed description for the report	Nvarchar	Max length 250	N/A	Yes
MetaDefinition	The configuration of the report stored in XML format. Contains a list of entities referenced, where statements and columns.	Nvarchar	Max length MAX	N/A	No
Breadcrumb	The menu structure for the report	Nvarchar	Max length 250	N/A	No
SQLDefinition	The SQL statement needed to generate the report	Nvarchar	Max length MAX	N/A	No
IsSystem	Is this a system defined report? System reports may not be modified or removed.	Bit	None	N/A	No
Entity	MetaTable				
Description	Contains a list of system generated META tables. This list is not administrable at run time but is governed systematically.				
Id	Unique auto-incremented seed for table	Integer	None	Primary	No
Metatable_GUID	A globally unique identifier for this table	UniqueIdentifier	None	N/A	No
TableName	The name of the table	Nvarchar	Max length 50	N/A	No
FriendlyName	A friendly name defining the contents of the table	Nvarchar	Max length 100	N/A	Yes

FriendlyDescription	A friendly description of the table	Nvarchar	Max length 250	N/A	Yes
TableType_Id	The type of table defined	Integer	FK to MetaTableType	Foreign	No
Entity	MetaColumn				
Description	Contains a list of system generated META columns. This list is not administrable at run time but is governed systematically.				
Id	Unique auto-incremented seed for table	Integer	None	Primary	No
Metacolumn_GUID	A globally unique identifier for this column	UniqueIdentifier	None	N/A	No
ColumnName	The name of the column	Nvarchar	Max length 50	N/A	No
IsIdentity	Is this an identity column	Bit	None	N/A	No
IsNullable	Is this column nullable	Bit	None	N/A	No
ColumnType_Id	The type of column defined	Integer	FK to MetaColumnType	Foreign	No
Table_Id	The table the column is associated to	Integer	FK to MetaTable	Foreign	No
Range	Contains a list of selected values for drop down purposes, or contains a source reference to a reference table.	Nvarchar	Max length MAX	N/A	Yes
Entity	MetaDependency				
Description	Defines dependency between all meta generated tables				
Id	Unique auto-incremented seed for table	Integer	None	Primary	No
Metadependency_GUID	A globally unique identifier for this dependency	UniqueIdentifier	None	N/A	No
ParentColumnName	The name of the parent column in the relationship	Nvarchar	Max length 50	N/A	No
ReferenceColumnName	The name of the reference column in the relationship	Nvarchar	Max length 50	N/A	No
ParentTable_Id	The parent table in the relationship	Integer	FK to MetaTable	Foreign	No
ReferenceTable_Id	The reference table in the relationship	Integer	FK to MetaTable	Foreign	No

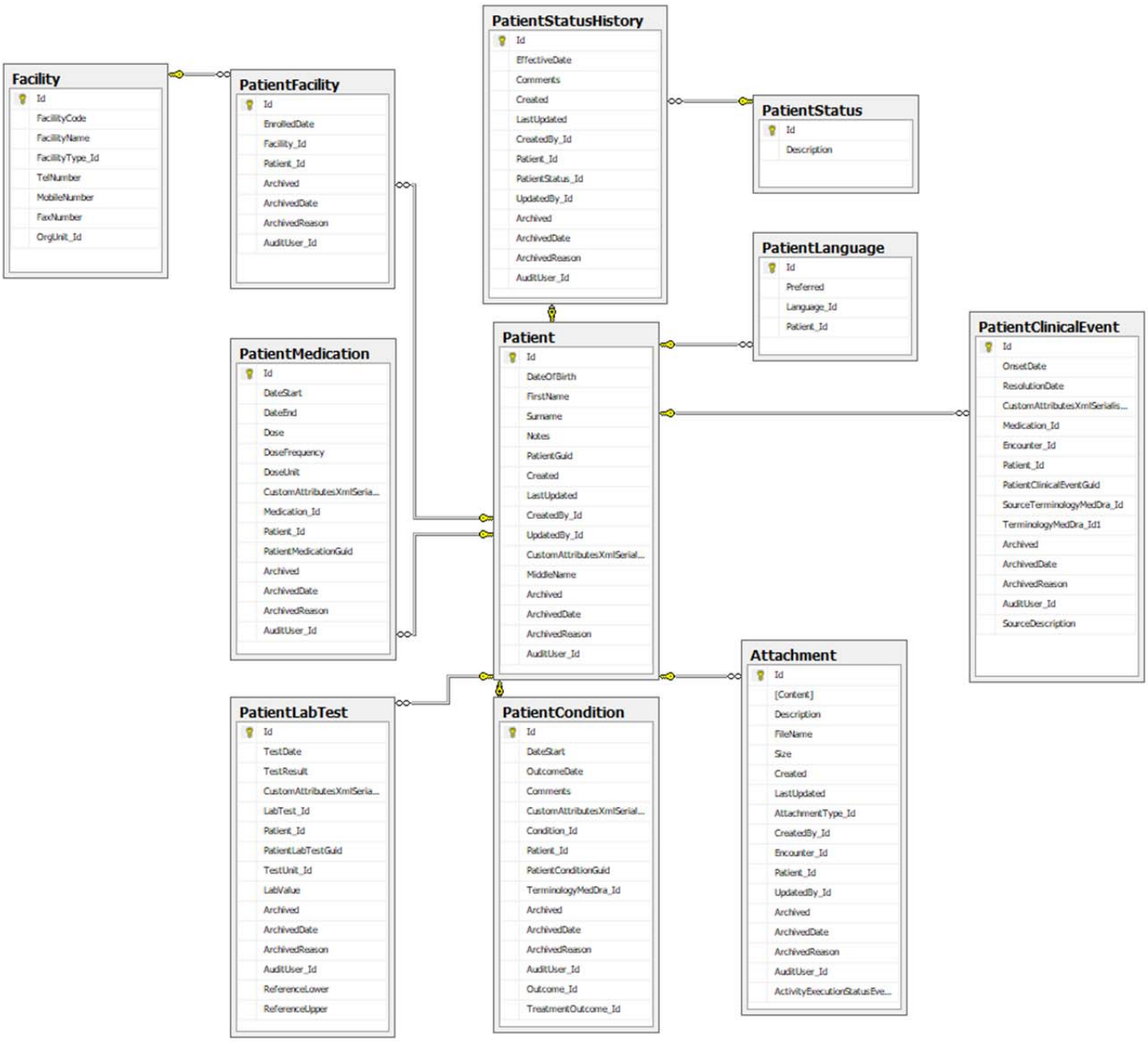
Entity	MetaPage				
Description	Contains a list of META pages for publication within the information portal				
Id	Unique auto-incremented seed for table	Integer	None	Primary	No
Metapage_GUID	A globally unique identifier for this page	UniqueIdentifier	None	N/A	No
PageName	The name of the publication page	Nvarchar	Max length 50	N/A	No
PageDefinition	A detailed description for the page	Nvarchar	Max length 250	N/A	Yes
MetaDefinition	The configuration of the page	Nvarchar	Max length MAX	N/A	No
Breadcrumb	The menu structure for the report	Nvarchar	Max length 250	N/A	No
IsSystem	Is this a system defined page? System pages may not be adjusted.	Bit	None	N/A	No
IsVisible	Is this page visible in the menu structure of the information portal	Bit	None	N/A	No
Entity	MetaWidget				
Description	Contains a list of META widgets that have been created for a page within the information portal				
Id	Unique auto-incremented seed for table	Integer	None	Primary	No
Metawidget_GUID	A globally unique identifier for this widget	UniqueIdentifier	None	N/A	No
WidgetName	The name of the publication widget	Nvarchar	Max length 50	N/A	No
WidgetDefinition	A detailed description for the widget	Nvarchar	Max length 250	N/A	Yes
Content	The content of the widget	Nvarchar	Max length MAX	N/A	No
WidgetType_Id	The type of widget that has been added	Integer	FK to MetaWidgetType	N/A	No
MetaPage_Id	The page the widget is associated to	Integer	FK to MetaPage	N/A	No

WidgetLocation	The location of the widget within the page. A widget can only exist in one of these locations and a location can only contain one widget.	Integer		N/A	No
	Widget Locations TopLeft = 1 TopRight = 2 MiddleLeft = 3 MiddleRight = 4 BottomLeft = 5 BottomRight = 6 Unassigned = 7				
WidgetStatus	The status of the widget.	Integer		N/A	No
	Widget Status Published = 1 Unpublished = 2 Unpublished widgets are not rendered to the user.				
Icon	The icon that should be displayed in the header of the widget. (http://fontawesome.com)	Nvarchar	Max length MAX	N/A	Yes

2.4 Patient Entities

This section contains all patient related entities

2.4.1 Entity Relationship Diagram



2.4.2 Reference Entities

Field Name	Field Description	Data Type	Field Constraints	Primary/ Foreign	Nullable
Entity	PatientStatus				
Description	Contains a list of patient statuses				
Id	Unique auto-incremented seed for table	Integer	None	Primary	No
Description	The name of the patient status	Nvarchar	Max length 50	N/A	No
	Patient Statuses • Active • Suspended • Stopped ART • Investigation • LTF • Stopped PMTCT • Transferred Out • Died				
Entity	Facility				
Description	Contains a list of facilities				
Id	Unique auto-incremented seed for table	Integer	None	Primary	No
FacilityCode	A unique code for the facility	Nvarchar	Max length 10	N/A	No
FacilityName	A unique name for the facility	Nvarchar	Max length 100	N/A	No
FacilityType_Id	The type of facility	Integer	FK to FacilityType	Foreign	Yes
TelNumber	The contact number for the facility	Nvarchar	Max length 30	N/A	Yes
MobileNumber	The mobile number for the facility	Nvarchar	Max length 30	N/A	Yes
FaxName	A fax number for the facility	Nvarchar	Max length 30	N/A	Yes
OrgUnit_Id	**** not used ****	Integer	FK to OrgUnit	Foreign	Yes

2.4.3 Transaction Entities

Field Name	Field Description	Data Type	Field Constraints	Primary/ Foreign	Nullable
Entity	Patient				
Description	Core patient demographics table containing a single unique record per patient				
Id	Unique auto-incremented seed for table	Integer	None	Primary	No
DateOfBirth	Patient date of birth	Date	None	N/A	No
FirstName	Patient first name	Nvarchar	Max length MAX Allow A-Z, a-z, space	N/A	No
Surname	Patient surname	Nvarchar	Max length MAX Allow A-Z, a-z, space	N/A	No
Notes	Generic patient notes	Nvarchar	Max length MAX	N/A	Yes
PatientGUID	A globally unique identifier for this patient	UniqueIdentifier	None	N/A	No
Created	AUDIT: Date record was created	Datetime	None	N/A	No
LastUpdated	AUDIT: Date record was last updated	Datetime	None	N/A	Yes
CreatedBy_Id	AUDIT: The user that created this record	Integer	FK to User	Foreign	Yes
UpdatedBy_Id	AUDIT: The user that last updated this record	Integer	FK to User	Foreign	Yes
CustomAttributesXmlSerialised	XML data structure	XML	None	N/A	Yes

	containing all custom attributes				
MiddleName	The middle name of the patient	Nvarchar	Max length MAX Allow A-Z, a-z, space	N/A	No
Archived	ARCHIVE: Is this record archived	Bit	None	N/A	No
ArchivedDate	ARCHIVE: The date the record was archived	DateTime	None	N/A	Yes
ArchivedReason	ARCHIVE: The reason the record was archived	Nvarchar	Max length 200	N/A	Yes
AuditUser_Id	ARCHIVE: Who archived the record	Integer	FK to User	Foreign	Yes
Entity	PatientCondition				
Description	A list of concomitant patient conditions				
Id	Unique auto-incremented seed for table	Integer	None	Primary	No
DateStart	The start date of the condition	Date	None	N/A	Yes
OutcomeDate	The end date of the condition	Date	None	N/A	Yes
Comments	Generic condition notes	Nvarchar	Max length 500	N/A	Yes
CustomAttributesXmlSerialised	XML data structure	XML	None	N/A	Yes

	containing all custom attributes				
Patient_Id	The patient the condition is associated to	Integer	FK to Patient	Foreign	No
PatientConditionGUID	A globally unique identifier for this patient condition	UniqueIdentifier	None	N/A	No
TerminologyMedDRA_Id	The MedDRA term for the condition	Integer	FK to TerminologyMedDRA	Foreign	No
Archived	ARCHIVE: Is this record archived	Bit	None	N/A	No
ArchivedDate	ARCHIVE: The date the record was archived	DateTime	None	N/A	Yes
ArchivedReason	ARCHIVE: The reason the record was archived	Nvarchar	Max length 200	N/A	Yes
AuditUser_Id	ARCHIVE: Who archived the record	Integer	FK to User	Foreign	Yes
Outcome_Id	The outcome of the condition	Integer	FK to Outcome	Foreign	No
TreatmentOutcome_Id	The treatment outcome of the condition	Integer	FK to TreatmentOutcome	Foreign	No
Entity	PatientMedication				
Description	A comprehensive list of medications associated to the patient				

Id	Unique auto-incremented seed for table	Integer	None	Primary	No
DateStart	The start date of the medication	Date	None	N/A	No
DateEnd	The end date of the medication	Date	None	N/A	Yes
Dose	The dose of the medication	Nvarchar	Max length 30	N/A	Yes
DoseFrequency	The frequency of the dose	Nvarchar	Max length 30	N/A	Yes
DoseUnit	The unit of the dose	Nvarchar	Max length 10	N/A	Yes
CustomAttributesXmlSerialised	XML data structure containing all custom attributes	XML	None	N/A	Yes
Medication_Id	The medication the patient is taking	Integer	FK to Medication	Foreign	No
Patient_Id	The patient the medication is associated to	Integer	FK to Patient	Foreign	No
PatientMedicationGUID	A globally unique identifier for this patient medication	UniqueIdentifier	None	N/A	No
Archived	ARCHIVE: Is this record archived	Bit	None	N/A	No

ArchivedDate	ARCHIVE: The date the record was archived	DateTime	None	N/A	Yes
ArchivedReason	ARCHIVE: The reason the record was archived	Nvarchar	Max length 200	N/A	Yes
AuditUser_Id	ARCHIVE: Who archived the record	Integer	FK to User	Foreign	Yes

Entity	PatientClinicalEvent				
Description	A list of adverse events associated to the patient. These events form the basis for pharmacovigilance activities conducted in the analytical portal.				
Id	Unique auto-incremented seed for table	Integer	None	Primary	No
OnsetDate	The onset date of the clinical event	Date	None	N/A	Yes
ResolutionDate	The resolution date of the clinical event	Date	None	N/A	Yes
CustomAttributesXmlSerialised	XML data structure containing all custom attributes	XML	None	N/A	Yes
Patient_Id	The patient the adverse event is associated to	Integer	FK to Patient	Foreign	No
PatientClinicalEventGuid	A globally unique identifier for this patient adverse event	UniqueIdentifier	None	N/A	No
SourceTerminologyMedDRA_Id	The MedDRA term for the clinical event that is selected by the clinician	Integer	FK to TerminologyMedDRA	Foreign	No
Archived	ARCHIVE: Is this record archived	Bit	None	N/A	No
ArchivedDate	ARCHIVE: The	DateTime	None	N/A	Yes

	date the record was archived				
ArchivedReason	ARCHIVE: The reason the record was archived	Nvarchar	Max length 200	N/A	Yes
AuditUser_Id	ARCHIVE: Who archived the record	Integer	FK to User	Foreign	Yes
SourceDescription	The adverse event description as per the patient or reporter	Nvarchar	Max length 500	N/A	Yes
Entity	PatientLabTest				
Description	A list of lab tests associated to the patient				
Id	Unique auto-incremented seed for table	Integer	None	Primary	No
TestDate	The test date of the lab test	Date	None	N/A	No
TestResult	The result of the test	Nvarchar	Max length 50	N/A	Yes
CustomAttributesXmlSerialised	XML data structure containing all custom attributes	XML	None	N/A	Yes
LabTest_Id	The lab test conducted	Integer	FK to LabTest	Foreign	No
Patient_Id	The patient the lab test is associated to	Integer	FK to Patient	Foreign	No

PatientLabTestGuid	A globally unique identifier for this patient lab test	UniqueIdentifier	None	N/A	No
TestUnit_Id	The unit of the test	Integer	FK to LabTestUnit	Foreign	Yes
LabValue	The value of the lab test	Nvarchar(20)	None	N/A	Yes
Archived	ARCHIVE: Is this record archived	Bit	None	N/A	No
ArchivedDate	ARCHIVE: The date the record was archived	DateTime	None	N/A	Yes
ArchivedReason	ARCHIVE: The reason the record was archived	Nvarchar	Max length 200	N/A	Yes
AuditUser_Id	ARCHIVE: Who archived the record	Integer	FK to User	Foreign	Yes
ReferenceLower	The lower range of the lab test result	Nvarchar(20)	None	N/A	Yes
ReferenceUpper	The upper range of the lab test result	Nvarchar(20)	None	N/A	Yes
Entity	PatientFacility				
Description	Facility history associated to the patient. The latest facility being the current facility the patient has been assigned to.				
Id	Unique auto-incremented seed for table	Integer	None	Primary	No
EnrolledDate	The date the	Date	None	N/A	No

	patient was enrolled into the facility				
Facility_Id	The facility the patient is linked to	Integer	FK to Facility	Foreign	No
Patient_Id	The patient the facility is associated to	Integer	FK to Patient	Foreign	No
Archived	ARCHIVE: Is this record archived	Bit	None	N/A	No
ArchivedDate	ARCHIVE: The date the record was archived	DateTime	None	N/A	Yes
ArchivedReason	ARCHIVE: The reason the record was archived	Nvarchar	Max length 200	N/A	Yes
AuditUser_Id	ARCHIVE: Who archived the record	Integer	FK to User	Foreign	Yes
Entity	PatientLanguage				
Description	A list of languages associated to the patient				
Id	Unique auto-incremented seed for table	Integer	None	Primary	No
Preferred	Is this the preferred language of the patient	Bit	None	N/A	No
Language_Id	The language associated to the patient	Integer	FK to Language	Foreign	No
Patient_Id	The patient	Integer	FK to Patient	Foreign	No

	the language is associated to				
Entity	Attachment				
Description	A list of attachments associated to the patient				
Id	Unique auto-incremented seed for table	Integer	None	Primary	No
Content	The content (in binary form) of the attachment	Varbinary	Max length Max	N/A	No
Description	A friendly description of the attachment	Nvarchar	Max length 100	N/A	Yes
FileName	The filename of the attachment	Nvarchar	Max length 50	N/A	No
Size	The size of the attachment	Bigint	None	N/A	No
Created	AUDIT: Date record was created	Datetime	None	N/A	No
LastUpdated	AUDIT: Date record was last updated	Datetime	None	N/A	Yes
AttachmentType_Id	The type of the attachment	Integer	FK to AttachmentType	Foreign	Yes
CreatedBy_Id	AUDIT: The user that created this record	Integer	FK to User	Foreign	Yes
Encounter_Id	The encounter	Integer	FK to Patient	Foreign	Yes

	the attachment is associated to				
Patient_Id	The patient the attachment is associated to	Integer	FK to Patient	Foreign	Yes
UpdatedBy_Id	AUDIT: The user that last updated this record	Integer	FK to User	Foreign	Yes
Archived	ARCHIVE: Is this record archived	Bit	None	N/A	No
ArchivedDate	ARCHIVE: The date the record was archived	DateTime	None	N/A	Yes
ArchivedReason	ARCHIVE: The reason the record was archived	Nvarchar	Max length 200	N/A	Yes
AuditUser_Id	ARCHIVE: Who archived the record	Integer	FK to User	Foreign	Yes
ActivityExecutionStatusEvent_Id	The work flow activity the attachment is associated to	Integer	FK to ActivityExecutionStatusEvent	Foreign	Yes
Entity	PatientStatusHistory				
Description	A list of status updates associated to the patient				
Id	Unique auto-incremented seed for table	Integer	None	Primary	No
EffectiveDate	The effective date of the	Datetime	None	N/A	No

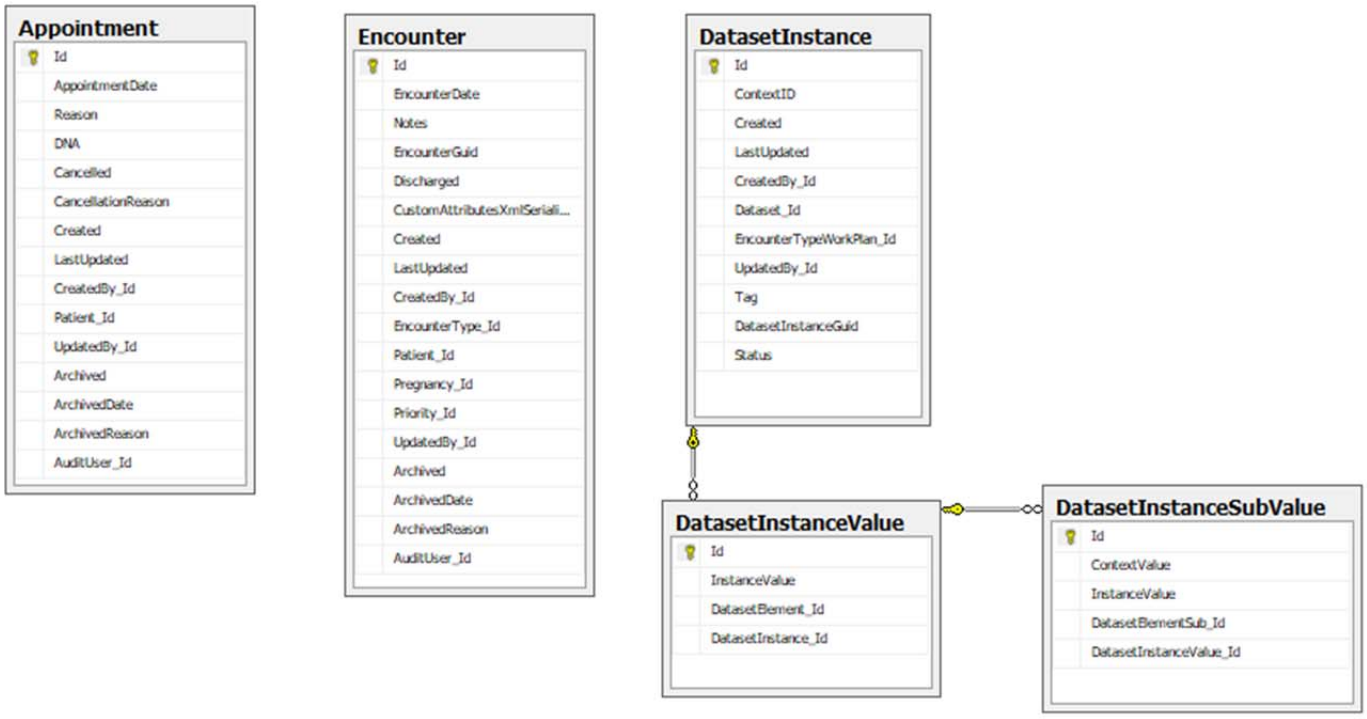
	status change				
Comments	Details associated to the status change	Nvarchar	Max length 100	N/A	Yes
Created	AUDIT: Date record was created	Datetime	None	N/A	No
LastUpdated	AUDIT: Date record was last updated	Datetime	None	N/A	Yes
CreatedBy_Id	AUDIT: The user that created this record	Integer	FK to User	Foreign	Yes
Patient_Id	The patient the status change is associated to	Integer	FK to Patient	Foreign	No
PatientStatus_Id	The status change of the patient	Integer	FK to PatientStatus	Foreign	Yes
UpdatedBy_Id	AUDIT: The user that last updated this record	Integer	FK to User	Foreign	Yes
Archived	ARCHIVE: Is this record archived	Bit	None	N/A	No
ArchivedDate	ARCHIVE: The date the record was archived	DateTime	None	N/A	Yes
ArchivedReason	ARCHIVE: The reason the record was archived	Nvarchar	Max length 200	N/A	Yes
AuditUser_Id	ARCHIVE:	Integer	FK to User	Foreign	Yes

	Who archived the record				

2.5 Encounter Entities

This section contains all entities that relate to clinical data collected in context of a patient encounter. A patient's longitudinal clinical history is effectively made up of a series of encounters.

2.5.1 Entity Relationship Diagram



2.5.2 Transaction Entities

Field Name	Field Description	Data Type	Field Constraints	Primary/ Foreign	Nullable
Entity	Appointment				
Description	A list of appointments associated to the patient				
Id	Unique auto-incremented seed for table	Integer	None	Primary	No
AppointmentDate	The date of the appointment	Datetime	None	N/A	No
Reason	The reason for the	Nvarchar	Max length 250	N/A	No

	appointment				
DNA	The patient did not arrive for their appointment	Bit	None	N/A	No
CancellationReason	The reason the appointment was cancelled	Nvarchar	Max length 250	N/A	Yes
Created	AUDIT: Date record was created	Datetime	None	N/A	No
LastUpdated	AUDIT: Date record was last updated	Datetime	None	N/A	Yes
CreatedBy_Id	AUDIT: The user that created this record	Integer	FK to User	Foreign	Yes
Patient_Id	The patient the appointment is associated to	Integer	FK to Patient	Foreign	No
UpdatedBy_Id	AUDIT: The user that last updated this record	Integer	FK to User	Foreign	Yes
Archived	ARCHIVE: Is this record archived	Bit	None	N/A	No
ArchivedDate	ARCHIVE: The date the record was archived	DateTime	None	N/A	Yes
ArchivedReason	ARCHIVE: The reason the record was archived	Nvarchar	Max length 200	N/A	Yes
AuditUser_Id	ARCHIVE: Who archived the record	Integer	FK to User	Foreign	Yes
Entity	Encounter				
Description	A list of encounters associated to the patient				
Id	Unique auto-incremented seed for table	Integer	None	Primary	No

EncounterDate	The date of the encounter	Datetime	None	N/A	No
Notes	General notes for the encounter	Nvarchar	Max length MAX	N/A	Yes
EncounterGUID	A globally unique identifier for this encounter	UniqueIdentifier	None	N/A	No
Discharged **** not in use ****	Is the encounter closed	Bit	None	N/A	No
CustomAttributesXmlSerialised	XML data structure containing all custom attributes	XML	None	N/A	Yes
Created	AUDIT: Date record was created	Datetime	None	N/A	No
LastUpdated	AUDIT: Date record was last updated	Datetime	None	N/A	Yes
CreatedBy_Id	AUDIT: The user that created this record	Integer	FK to User	Foreign	Yes
EncounterType_Id	The type of encounter	Integer	FK to EncounterType	Foreign	No
Patient_Id	The patient the encounter is associated to	Integer	FK to Patient	Foreign	No
Priority_Id	The priority of the encounter	Integer	FK to Priority	Foreign	Yes
Pregnancy_Id **** not in use ****	Is the patient pregnant	Integer	FK to Pregnancy	Foreign	Yes
UpdatedBy_Id	AUDIT: The user that last updated this record	Integer	FK to User	Foreign	Yes
Archived	ARCHIVE: Is this record archived	Bit	None	N/A	No
ArchivedDate	ARCHIVE: The date the record was	DateTime	None	N/A	Yes

	archived				
ArchivedReason	ARCHIVE: The reason the record was archived	Nvarchar	Max length 200	N/A	Yes
AuditUser_Id	ARCHIVE: Who archived the record	Integer	FK to User	Foreign	Yes
Entity	DatasetInstance				
Description	A list of dataset instances. A dataset instance is a physical instantiation of a dataset and contains clinical data collected.				
Id	Unique auto-incremented seed for table	Integer	None	Primary	No
ContextID	The unique id of the associated entity.	Integer	None	N/A	No
Created	AUDIT: Date record was created	Datetime	None	N/A	No
LastUpdated	AUDIT: Date record was last updated	Datetime	None	N/A	Yes
CreatedBy_Id	AUDIT: The user that created this record	Integer	FK to User	Foreign	Yes
Dataset_Id	The dataset associated to the instance. This link defines what data can be expected to be collected within the instance.	Integer	FK to Dataset	Foreign	No
EncounterTypeWorkPlan_Id	The encounter type work plan that resulted in the instantiation of the dataset	Integer	FK to EncounterTypeWorkPlan	Foreign	No
UpdatedBy_Id	AUDIT: The user	Integer	FK to User	Foreign	Yes

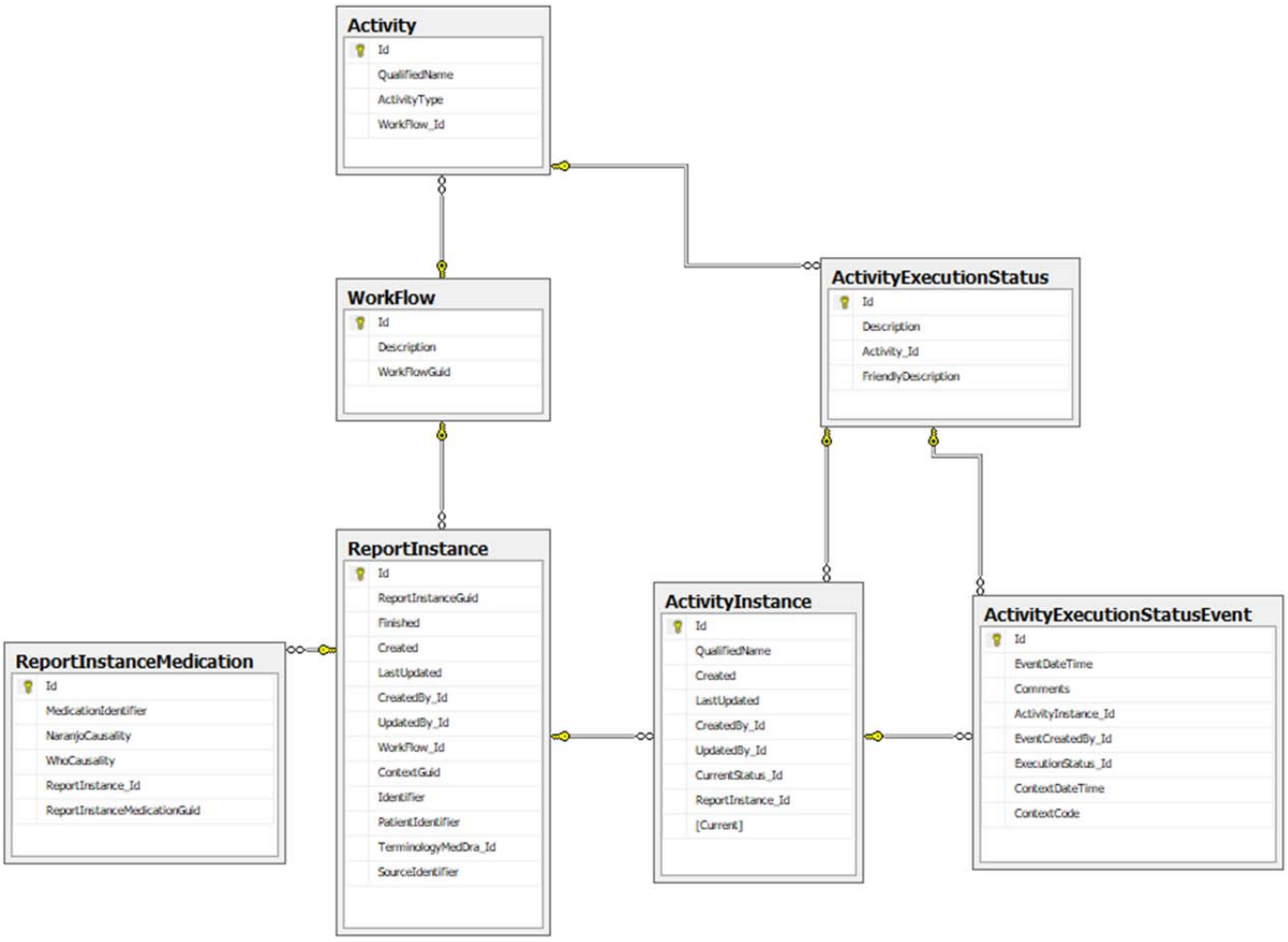
	that last updated this record				
Tag	Active for active reporting and Spontaneous for spontaneous reporting	Nvarchar	Max length MAX	N/A	Yes
DatasetInstanceGUID	A globally unique identifier for this dataset instance	UniqueIdentifier	None	N/A	No
Status	The current status of the dataset instance Incomplete = 1, Complete = 2	Integer		N/A	No
Entity	DatasetInstanceValue				
Description	A list of clinical values collected per dataset instance				
Id	Unique auto-incremented seed for table	Integer	None	Primary	No
InstanceValue	The value of the associated clinical element	Nvarchar	Max length 10	N/A	No
DatasetElement_Id	The dataset element that defines the corresponding value	Integer	FK to DatasetElement	Foreign	No
DatasetInstance_Id	The dataset instance that contains the collection of instance values	Integer	FK to DatasetInstance	Foreign	No
Entity	DatasetInstanceSubValue				
Description	A list of clinical values per dataset element table				
Id	Unique auto-incremented seed for table	Integer	None	Primary	No

ContextValue	A globally unique identifier for this sub table row	UniqueIdentifier	None	N/A	No
InstanceValue	The value of the associated clinical element	Nvarchar	Max length 10	N/A	No
DatasetElementSub_Id	The dataset sub-element that defines the corresponding value	Integer	FK to DatasetElementSub	Foreign	No
DatasetInstanceValue_Id	The dataset instance value that contains the collection of sub instance values	Integer	FK to DatasetInstanceValue	Foreign	No

2.6 Workflow Entities

This section contains a list of all work flow entities that govern the processing of workflow-based pharmacovigilance activities for newly created adverse event reports.

2.6.1 Entity Relationship Diagram



2.6.2 Reference Entities

Field Name	Field Description	Data Type	Field Constraints	Primary/ Foreign	Nullable
Entity	Activity				
Description	Contains a list of available activities to be conducted by the pharmacovigilance unit per report type.				
Id	Unique auto-incremented seed for table	Integer	None	Primary	No
QualifiedName	Qualified name of the workflow activity	Nvarchar	Max length 50	N/A	No
	Activities Confirm Report Data Set MedDRA and Causality Extract E2B				
ActivityType	Type of activity	Integer		N/A	No
	Activity Types UserDrivenActivity = 1, SystemDrivenActivity = 2, PublicationActivity = 3				
Workflow_Id	The workflow the activity is associated to	Integer	FK to WorkFlow	Foreign	No
Entity	ActivityExecutionStatus				
Description	Contains a list of available statuses per activity				
Id	Unique auto-incremented seed for table	Integer	None	Primary	No
Description	The name of the status	Nvarchar	Max length 50	N/A	No
	Activities UNCONFIRMED CONFIRMED DELETED NOTSET MEDDRASET CAUSALITYSET NOTGENERATED E2BINITIATED E2BGENERATED E2BSUBMITTED				

FriendlyDescription	A friendly description of the activity status	Nvarchar	Max length 100	N/A	Yes
Activity_Id	The activity the status is associated to	Integer	FK to Activity	Foreign	No
Entity	WorkFlow				
Description	Contains a list of available work flows for implementation				
Id	Unique auto-incremented seed for table	Integer	None	Primary	No
Description	Workflow name	Nvarchar	Max length 100	N/A	No
	Workflows New Active Surveillance Report New Spontaneous Surveillance Report				
WorkFlowGUID	A globally unique identifier for this workflow	UniqueIdentifier	None	N/A	No

2.6.3 Transaction Entities

Field Name	Field Description	Data Type	Field Constraints	Primary/ Foreign	Nullable
Entity	ActivityExecutionStatusEvent				
Description	A series of timestamped status changes logged per report activity. Track progress on workflow.				
Id	Unique auto-incremented seed for table	Integer	None	Primary	No
EventDateTime	The date and time of the activity status change	Datetime		N/A	No
Comments	Comments noted by the user when changing status	Nvarchar	Max length MAX	N/A	Yes
ActivityInstance_Id	The activity instance the status change is associated to	Integer	FK to ActivityInstance	Foreign	No
EventCreatedBy_Id	The user that generated the status change	Integer	FK to User	Foreign	Yes
ExecutionStatus_Id	The new activity status	Integer	FK to ActivityExecutionStatus	Foreign	No
ContextDateTime	Contextual date and time based on the status change	Datetime		N/A	Yes
ContextCode	Contextual code based on the status change	Nvarchar	Max length 20	N/A	Yes
Entity	ActivityInstance				
Description	Physical instantiation of a work flow activity. A work flow will be comprised of multiple activities as they move from one state to the next				
Id	Unique auto-incremented seed for table	Integer	None	Primary	No
QualifiedName	Qualified name of	Nvarchar	Max length 50	N/A	No

	the workflow activity				
Created	AUDIT: Date record was created	Datetime	None	N/A	No
LastUpdated	AUDIT: Date record was last updated	Datetime	None	N/A	Yes
CreatedBy_Id	AUDIT: The user that created this record	Integer	FK to User	Foreign	Yes
UpdatedBy_Id	AUDIT: The user that last updated this record	Integer	FK to User	Foreign	Yes
CurrentStatus_Id	The current event for this activity	Integer	FK to ActivityExecutionStatusEvent	Foreign	Yes
ReportInstance_Id	The report instance the activity is associated to	Integer	FK to ReportInstance	Foreign	No
Current	Is this activity the current activity for the report instance	Bit		N/A	No
Entity	ReportInstance				
Description	Physical instantiation of a work flow for a newly logged adverse event report				
Id	Unique auto-incremented seed for table	Integer	None	Primary	No
ReportInstanceGUID	A globally unique identifier for this report instance	UniqueIdentifier	None	N/A	No
Finished	The date of completions	Datetime		N/A	Yes
Created	AUDIT: Date record was created	Datetime	None	N/A	No
LastUpdated	AUDIT: Date	Datetime	None	N/A	Yes

	record was last updated				
CreatedBy_Id	AUDIT: The user that created this record	Integer	FK to User	Foreign	Yes
UpdatedBy_Id	AUDIT: The user that last updated this record	Integer	FK to User	Foreign	Yes
WorkFlow_Id	The workflow the report instance is associated to	Integer	FK to WorkFlow	Foreign	No
ContextGUID	The unique identifier of the adverse event report	UniqueIdentifier	None	N/A	No
Identifier	A unique identifier for the adverse event report	Nvarchar	Max length MAX	N/A	No
PatientIdentifier	A unique patient identifier for the adverse event report	Nvarchar	Max length MAX	N/A	No
TerminologyMedDra_Id	The MedDRA terminology set by the pharmacovigilance unit for this event	Integer	FK to TerminologyMedDra	Foreign	Yes
SourceIdentifier	A description of the source event	Nvarchar	Max length MAX	N/A	No
Entity	ReportInstanceMedication				
Description	Medications associated to a physical instantiation of a work flow for a newly logged adverse event report Medications who overlap the onset date of the adverse event will be automatically linked to the report instance on creation.				
Id	Unique auto-incremented seed for table	Integer	None	Primary	No
MedicationIdentifier	A unique medication	Nvarchar	Max length MAX	N/A	Yes

	identifier for the adverse event report				
NaranjoCausality	Naranjo causality setting for medication	Nvarchar	Max length 30	N/A	Yes
WHOCausality	WHO causality setting for medication	Nvarchar	Max length 30	N/A	Yes
ReportInstance_Id	The report instance the medication is associated to	Integer	FK to ReportInstance	Foreign	No
ReportInstanceMedicationGUID	A globally unique identifier for this medication	UniqueIdentifier	None	N/A	No

2.7 Other Entities

This section contains a list of all remaining entities.

2.7.1 Reference Entities

Field Name	Field Description	Data Type	Field Constraints	Primary/ Foreign	Nullable
Entity	AttachmentType				
Description	Contains a list of attachment types				
Id	Unique auto-incremented seed for table	Integer	None	Primary	No
Description	The name of the attachment type	Nvarchar	Max length 50	N/A	No
Key	The name of the attachment type	Nvarchar	Max length 4	N/A	No
	Attachment Types MS Word 2003-2007 Document MS Excel 2003-2007 Document MS Word Document MS Excel Document Portable Document Format Image | JPEG Image | JPEG Image | PNG Image | BMP				
Entity	Condition				
Description	Contains a list of concomitant conditions				
Id	Unique auto-incremented seed for table	Integer	None	Primary	No
Description	The description of the condition	Nvarchar	Max length 50	N/A	No
Chronic	Is this condition chronic	Bit		N/A	No
	Conditions HIV Malaria TB				
Entity	ContextType				
Description	Contains a list of context types. A context type is defined as the type of entity a dataset will reference				

Id	Unique auto-incremented seed for table	Integer	None	Primary	No
Description	The name of the context type	Nvarchar	Max length 50	N/A	No
	Context Types Encounter Patient Pregnancy Global PatientClinicalEvent DatasetInstance				
Entity	FacilityType				
Description	Contains a list of facility types				
Id	Unique auto-incremented seed for table	Integer	None	Primary	No
Description	The name of the facility type	Nvarchar	Max length 50	N/A	No
	Facility Types Unknown Hospital CHC PHC				
Entity	Holiday				
Description	Contains a list of holidays administered within the system				
Id	Unique auto-incremented seed for table	Integer	None	Primary	No
HolidayDate	The date of the holiday	Datetime		N/A	No
Description	The name of the facility type	Nvarchar	Max length 100	N/A	No
Entity	LabResult				
Description	Contains a list of laboratory results				
Id	Unique auto-incremented seed for table	Integer	None	Primary	No
Description	The name of the laboratory result	Nvarchar	Max length 50	N/A	No
	Laboratory Results Abnormal Borderline Improved Inconclusive				

	Negative Normal Positive Progressed Seronegative Seropositive Stable				
Entity	LabTest				
Description	Contains a list of lab tests				
Id	Unique auto-incremented seed for table	Integer	None	Primary	No
Active	Is the lab test currently active for selection	Bit	None	N/A	No
Description	The name of the lab test	Nvarchar	Max length 50	N/A	No
	Lab Tests AFB Smear Result Albumin Alkaline phosphatase ALT (SGPT) Amylase AST (SGOT) Audiometry Blood Glucose CD4 Count Chest X-Ray Creatinine Clearance Culture Results ESR Glucose Haemoglobin HBC IgM HBsAg HCV Ab Hearing test Heart Rate Hematocrit Hepatitis B Virus Hepatitis C Virus HIV Antibody Lactic acid Lipase				

	Neurological Exam Platelet Count PR Interval Pregnancy Test QRS Duration QT Duration QT Interval (Fredericia) QT Interval (uncorrected) QTc interval RBC Count Serum Calcium (Ca++) Serum Creatinine (SCr) Serum Magnesium (Mg++) Serum Potassium (K+) Smear Total Bilirubin Total WBC TSH Visual acuity WBC Count				
Entity	LabTestUnit				
Description	Contains a list of lab test units				
Id	Unique auto-incremented seed for table	Integer	None	Primary	No
Active	Is the lab test unit currently active for selection	Bit	None	N/A	No
Description	The name of the lab test unit	Nvarchar	Max length 50	N/A	No
	Lab Test Unit % % hearing loss left ear % hearing loss right ear µg/dL µg/L beats per minute breaths per minute cavities cells/mm 3 Eye, Left (OS) Eye, Right (OD) g/dL g/L				

	IU/L kg/m 2 mEq/L mg/24 hr mg/dL Milliseconds min mL/min mm Hg mm/h mmol/kg mmol/L mOsm/kg ms N/A ng/dL ng/L ng/mL ng/mL/hr nmol/L pg/mL pH pmol/L sec U/L X 10 3 /mm 3 X 10 6 /mm 3 X 10 9 /L µg/dL µg/L µmol/L µU/L µU/mL				
Entity	Language				
Description	Contains a list of languages. **** not in use ******				
Id	Unique auto-incremented seed for table	Integer	None	Primary	No
Description	The name of the language	Nvarchar	Max length 20	N/A	No
Entity	Medication				
Description	Contains a list of medications				

Id	Unique auto-incremented seed for table	Integer	None	Primary	No
DrugName	The name of the drug	Nvarchar	Max length 100	N/A	No
Active	Is the medication active and available for use in the system	Bit	None	N/A	No
PackSize	The pack size of the medication	Integer	None	N/A	No
Strength	The strength of the medication	Nvarchar	Max length 40	N/A	No
CatalogNo	The catalog number of the medication	Nvarchar	Max length 10	N/A	Yes
MedicationForm_Id	The form of the medication	Integer	FK to MedicationForm	Foreign	Yes
Entity	MedicationForm				
Description	Contains a list of medication forms				
Id	Unique auto-incremented seed for table	Integer	None	Primary	No
Description	The name of the medication form	Nvarchar	Max length 50	N/A	No
	Medication Forms Ampoule Bottle Cartridge Condom Cream Disc for lab testing Disposable Each Ear drops Elixir Enema Gas Gel Granules Inhaler IUD Liquid Lotion Nasal drops Nasal spray Net Ointment				

	Ophthalmic drops Ophthalmic ointment Ophthalmic strips Oral drops Oral gel Pessary Powder Rectal tube Respiratory solution Rod Shampoo Solid Solution Spray Suppository Suspension Syringe Syrup Tablet or capsule Test Tincture Transdermal patch Unknown Vial				
Entity	Outcome				
Description	Contains a list of condition outcomes				
Id	Unique auto-incremented seed for table	Integer	None	Primary	No
Description	The description of the outcome	Nvarchar	Max length 50	N/A	No
	Outcomes Recovered/Resolved Recovered/Resolved With Sequelae Recovering/Resolving Not Recovered/Not Resolved Fatal Unknown				
Entity	Priority				
Description	Contains a list of priorities				
Id	Unique auto-incremented seed for	Integer	None	Primary	No

	table				
Description	The description of the priority	Nvarchar	Max length 50	N/A	No
	Priorities Urgent High Medium Low				
Entity	Role				
Description	Contains a list of security roles				
Id	Unique auto-incremented seed for table	Integer	None	Primary	No
Name	The name of the role	Nvarchar	Max length 30	N/A	No
Key	The unique key for this role	Nvarchar	Max length 30	N/A	No
	Roles Administrator Registration Clerk Data Capturer Clinician Analytics Reporter Publisher Reporter Administrator Publisher Administrator				
Entity	TerminologyMedDRA				
Description	Contains a list of all MedDRA definitions				
Id	Unique auto-incremented seed for table	Integer	None	Primary	No
MedDraTerm	The MedDRA term	Nvarchar	Max length 100	N/A	No
MedDraCode	The MedDRA code	Nvarchar	Max length 10	N/A	No
MedDraTermType	The MedDRA term type	Nvarchar	Max length 4	N/A	No
	MedDRA Term Types LLT HLGT SOC PT HLT				
Parent_Id	The parent of this MedDRA term	Integer	FK to	Foreign	Yes

			TerminologyMedDRA		
Entity	TreatmentOutcome				
Description	Contains a list of treatment outcomes				
Id	Unique auto-incremented seed for table	Integer	None	Primary	No
Description	The description of the treatment outcome	Nvarchar	Max length 50	N/A	No
	Treatment Outcomes Cured Treatment Completed Treatment Failed Died Lost to Follow-up Not Evaluated				
Entity	User				
Description	Contains a list of system users.				
Id	Unique auto-incremented seed for table	Integer	None	Primary	No
Email	The email address of the user	Nvarchar	Max length 256	N/A	Yes
EmailConfirmed	Is email address confirmed	Bit	None	N/A	No
PasswordHash	Encrypted user password	Nvarchar	Max length MAX	N/A	Yes
SecurityStamp	Invalidate existing sign ins if user details changed	Nvarchar	Max length MAX	N/A	Yes
PhoneNumber	The phone number of the user	Nvarchar	Max length MAX	N/A	Yes
PhoneNumberConfirmed	Is phone number confirmed	Bit	None	N/A	No
TwoFactorEnabled	Two forms of security validation	Bit	None	N/A	No
LockoutEndDateUtc	Locked out date	Datetime	None	N/A	Yes
LockoutEnabled	User has been locked out of account	Bit	None	N/A	No
AccessFailedCount	Number of times user has failed security validations	Integer	None	N/A	No
UserName	The user name of the user	Nvarchar	Max length 256	N/A	No
FirstName	The first name of the user	Nvarchar	Max length MAX	N/A	Yes
LastName	The last name of the user	Nvarchar	Max length MAX	N/A	Yes

Active	User is currently active	Bit	None	N/A	No
CurrentContext	The portal the user is currently logged into	Nvarchar	Max length MAX	N/A	Yes
EulaAcceptanceDate	The date the user accepted the EULA on first login	Datetime	None	N/A	Yes
AllowDatasetDownload	Does the user have the ability to download data from the analytical portal	Bit	None	N/A	No

2.7.2 Custom Attribute Entities

Field Name	Field Description	Data Type	Field Constraints	Primary/ Foreign	Nullable
Entity	CustomAttributeConfiguration				
Description	Contains a list of custom attributes that extend core system entities				
Id	Unique auto-incremented seed for table	Integer	None	Primary	No
ExtendableTypeName	The entity that the custom attribute belongs to	Nvarchar	Max length MAX	N/A	Yes
CustomAttributeType	The type of custom attribute	Integer		N/A	No
	Custom Attribute Types None = 0 Numeric = 1 String = 2 Selection = 3 DateTime = 4 FirstClassProperty = 5				
Category	The category that the attribute is allocated to	Nvarchar	Max length MAX	N/A	Yes
AttributeKey	The unique name of the attribute	Nvarchar	Max length MAX	N/A	Yes
IsRequired	Is this attribute mandatory	Bit	None	N/A	No
StringMaxLength	Maximum length of the string attribute	Integer	None	N/A	Yes
NumericMinValue	Minimum value of the numeric attribute	Integer	None	N/A	Yes
NumericMaxValue	Maximum value of the numeric attribute	Integer	None	N/A	Yes
FutureDateOnly	For date attributes, must the attribute be in the future	Bit	None	N/A	No
PastDateOnly	For date attributes, must the attribute be in the past	Bit	None	N/A	No
IsSearchable	Include this attribute in the patient or encounter search functions	Bit	None	N/A	No
AttributeDetail	Additional help for the attribute	Nvarchar	Max length 150	N/A	Yes
Entity	SelectionDataItem				

Description	A list of selection values for a selection custom attribute				
Id	Unique auto-incremented seed for table	Integer	None	Primary	No
AttributeKey	The unique name of the attribute	Nvarchar	Max length MAX	N/A	Yes
SelectionKey	A unique value identifying the selection value	Nvarchar	Max length MAX	N/A	Yes
Value	A description of the selection value	Nvarchar	Max length MAX	N/A	Yes

2.7.3 Bridging Entities

Field Name	Field Description	Data Type	Field Constraints	Primary/ Foreign	Nullable
Entity	ConditionLabTest				
Description	A list of lab tests associated to the corresponding condition				
Id	Unique auto-incremented seed for table	Integer	None	Primary	No
Condition_Id	The condition that the lab test is associated to	Integer	FK to Condition	N/A	No
LabTest_Id	The lab test that has been associated to the condition	Integer	FK to LabTest	N/A	No
Entity	ConditionMeddra				
Description	A list of MedDra terms associated to the corresponding condition				
Id	Unique auto-incremented seed for table	Integer	None	Primary	No
Condition_Id	The condition that the MedDra term is associated to	Integer	FK to Condition	N/A	No
TerminologyMedDra_Id	The MedDra term that has been associated to the condition	Integer	FK to TerminologyMedDra	N/A	No
Entity	ConditionMedication				
Description	A list of medications associated to the corresponding condition				
Id	Unique auto-incremented seed for table	Integer	None	Primary	No
Condition_Id	The condition that the medication is associated to	Integer	FK to Condition	N/A	No
Medication_Id	The medication that has been	Integer	FK to Medication	N/A	No

	associated to the condition				
Entity	UserFacility				
Description	A list of facilities that the user has access to				
Id	Unique auto-incremented seed for table	Integer	None	Primary	No
Facility_Id	The facility that has been linked to the user	Integer	FK to Facility	N/A	No
User_Id	The user that is been administered	Integer	FK to User	N/A	No
Entity	UserRole				
Description	A list of roles that the user has access to				
Id	Unique auto-incremented seed for table	Integer	None	Primary	No
Role_Id	The role that has been linked to the user	Integer	FK to Role	N/A	No
User_Id	The user that is been administered	Integer	FK to User	N/A	No

2.7.4 Miscellaneous Entities

Field Name	Field Description	Data Type	Field Constraints	Primary/ Foreign	Nullable
Entity	AuditLog				
Description	A detailed audit log of PViMS activity				
Id	Unique auto-incremented seed for table	Integer	None	Primary	No
AuditType	The type of activity being audited	Integer	None	N/A	No
	Audit Types InvalidSubscriberAccess = 1 ValidSubscriberAccess = 2 InValidSubscriberPost = 3 ValidSubscriberPost = 4 UserLogin = 5 InValidMedDRAImport = 6 ValidMedDRAImport = 7				
ActionDate	The date and time of the activity	DateTime	None	N/A	No
Details	Additional details describing the activity	Nvarchar	Max length MAX	N/A	Yes

User_Id	The user that conducted the activity	Integer	FK to User	N/A	Yes
Log	Additional details logged during the activity	Nvarchar	Max length MAX	N/A	Yes
Entity	CohortGroup				
Description	A list of cohorts that have been configured in the system				
Id	Unique auto-incremented seed for table	Integer	None	Primary	No
CohortName	The name of the cohort	Nvarchar	Max length 50	N/A	No
CohortCode	A unique code identifying the cohort	Nvarchar	Max length 5	N/A	No
LastPatientNo	**** not in use ****	Integer	None	N/A	No
StartDate	The start date of the cohort. Used as the minimum start date when generating the population set when doing cohort analysis	Datetime	None	N/A	No
FinishDate	The end date of the cohort. Used as the maximum start date when generating the population set when doing cohort analysis	Datetime	None	N/A	No
MinEnrolment	**** not in use ****	Integer	None	N/A	No
MaxEnrolment	**** not in use ****	Integer	None	N/A	No
Condition_Id	Is this cohort targeting a specific condition?	Integer	FK to Condition	N/A	Yes
Entity	CohortGroupEnrolment				
Description	A list of patients enrolled into the cohort				
Id	Unique auto-incremented seed for table	Integer	None	Primary	No
EnroledDate	The date the patient was enrolled into the cohort	Datetime	None	N/A	No
CohortGroup_Id	The cohort the patient has been enrolled into	Integer	FK to CohortGroup	N/A	Yes
Patient_Id	The patient that is being enrolled into the cohort	Integer	FK to Patient	N/A	Yes
DeenroledDate	The date the patient was de-enrolled from the cohort	Datetime	None	N/A	Yes

Archived	ARCHIVE: Is this record archived	Bit	None	N/A	No
ArchivedDate	ARCHIVE: The date the record was archived	DateTime	None	N/A	Yes
ArchivedReason	ARCHIVE: The reason the record was archived	Nvarchar	Max length 200	N/A	Yes
AuditUser_Id	ARCHIVE: Who archived the record	Integer	FK to User	Foreign	Yes
Entity	Config				
Description	A list of configurable PViMS values				
Id	Unique auto-incremented seed for table	Integer	None	Primary	No
ConfigType	The type of configuration	Integer	None	N/A	No
	Config Types E2BVersion = 1 WebServiceSubscriberList = 2 AssessmentScale = 3 MedDRAVersion = 4 ReportInstanceNewAlertCount = 5 MedicationOnsetCheckPeriodWeeks = 6 MetaDataLastUpdated = 7				
ConfigValue	The value of the configuration	Nvarchar	Max length 100	N/A	No
Created	AUDIT: Date record was created	Datetime	None	N/A	No
LastUpdated	AUDIT: Date record was last updated	Datetime	None	N/A	Yes
CreatedBy_Id	AUDIT: The user that created this record	Integer	FK to User	Foreign	Yes
UpdatedBy_Id	AUDIT: The user that last updated this record	Integer	FK to User	Foreign	Yes
Entity	PostDeployment				
Description	A list of post implementation scripts that need to be executed against the PViMS database. Scripts must be stored in the PostDeploymentScripts folder.				
Id	Unique auto-incremented seed for table	Integer	None	Primary	No
ScriptGUID	A globally unique identifier for this script	UniqueIdentifier	None	N/A	No
ScriptFileName	The name of the script to be	Nvarchar	Max length 200	N/A	No

	executed				
ScriptDescription	**A description of the script to be executed**	Nvarchar	Max length 200	N/A	No
RunDate	The date the script was run	Datetime	None	N/A	Yes
StatusCode	The status of the script execution	Integer	None	N/A	Yes
	Script Status 200 successful 404 not found				
StatusMessage	Status message, File not found	Nvarchar	Max length 200	N/A	Yes
RunRank	Order the script should be executed in	Integer	None	N/A	No
Entity	RiskFactor				
Description	Risk factors that have been configured for use in the analytical portal.				
Id	Unique auto-incremented seed for table	Integer	None	Primary	No
FactorName	A unique name describing the risk factor	Nvarchar	Max length 50	N/A	No
Criteria	The SQL query that is incorporated into the analyser stored procedure	Nvarchar	Max length MAX	N/A	No
Display	The name that is displayed in the analytical portal	Nvarchar	Max length 20	N/A	Yes
IsSystem	Is this a system defined risk factor	Bit	None	N/A	No
Active	Is this risk factor active for selection	Bit	None	N/A	No
Entity	RiskFactorOption				
Description	Sub options per risk factor				
Id	Unique auto-incremented seed for table	Integer	None	Primary	No
OptionName	A unique name describing the risk factor option	Nvarchar	Max length 50	N/A	No
Criteria	The criteria that is consumed by the risk factor to determine if the patient is successfully matched to the risk factor	Nvarchar	Max length 250	N/A	No
Display	The name that is displayed in the analytical portal	Nvarchar	Max length 30	N/A	Yes

RiskFactor_Id	The risk factor that the option is associated to	Integer	FK to RiskFactor	N/A	Yes
Entity	SiteContactDetail				
Description	List of contact details that are used to populate the E2B extract				
Id	Unique auto-incremented seed for table	Integer	None	Primary	No
ContactType	The type of address	Integer	None	N/A	No
	Contact Types RegulatoryAuthority = 1 ReportingAuthority = 2				
ContactFirstName	The first name of the primary contact	Nvarchar	Max length 30	N/A	No
ContactSurname	The surname of the primary contact	Nvarchar	Max length 30	N/A	No
StreetAddress	Contact details	Nvarchar	Max length 100	N/A	No
City		Nvarchar	Max length 50	N/A	No
State		Nvarchar	Max length 50	N/A	Yes
PostCode		Nvarchar	Max length 20	N/A	Yes
ContactNumber		Nvarchar	Max length 50	N/A	Yes
ContactEmail		Nvarchar	Max length 50	N/A	Yes
CountryCode		Nvarchar	Max length 10	N/A	Yes
OrganisationName	The name of the organisation	Nvarchar	Max length 50	N/A	Yes
Created	AUDIT: Date record was created	Datetime	None	N/A	No
LastUpdated	AUDIT: Date record was last updated	Datetime	None	N/A	Yes
CreatedBy_Id	AUDIT: The user that created this record	Integer	FK to User	Foreign	Yes
UpdatedBy_Id	AUDIT: The user that last updated this record	Integer	FK to User	Foreign	Yes

3 Database Security

This section details administrative and operational best practises that should be performed from a security perspective when implementing the PViMS database.

3.1 Surface Area Reduction

It is best practise to only install the SQL Server options that are required for PViMS to operate, thereby reducing the attack surface. The following best practise should be observed to reduce the attack service:

- Install only those components that you will immediately use. Additional components can always be installed as needed.
- Enable only the optional features that you will immediately use.
- Review optional feature usage before doing an in-place upgrade and disable unneeded features either before or after the upgrade.
- Develop a policy with respect to permitted network connectivity choices. Use SQL Server Policy-Based Management to standardize this policy.
- Develop a policy for the usage of optional features. Use SQL Server Policy-Based Management to standardize optional feature enabling. Document any exceptions to the policy on a per-instance basis.
- Turn off unneeded services by setting the service to either *manual startup* or *disabled*.
- Configure only those server network interfaces that you will actually use.

3.2 Service Account Selection and Management

SQL Server 2008 executes as a set of windows services which can be configured to use its own account. When choosing service accounts, consider the principle of least privilege which dictates that the service account has the exact permissions needed to execute the service and nothing more. Account should also be implemented in isolation so that service account should not only be different from one another, they should not be used by any other services on the same server. The following best practise should be observed when selecting service accounts:

- Always use SQL Server Configuration Manager to change service accounts.
- If you use a user or domain account, change the service account password at regular intervals.
- Use CREDENTIALs to execute job steps that require specific privileges rather than adjusting the privilege to the SQL Server Agent service account.
- If a user needs to execute a job that requires different Windows credentials, assign them a proxy account that has just enough permissions to get the task done.

3.3 SQL Server Best Practises Analyser

SQL Server 2008 R2 Best Practices Analyzer (BPA) gathers data from Microsoft Windows and SQL Server configuration settings and uses a predefined list of SQL Server 2008/2008 R2 recommendations and best practices to determine if there are potential security issues in the PViMS database environment.

- Run SQL Server Best Practices Analyzer against SQL Server 2008/2008 R2.
- Use Microsoft Security Compliance Manager to provide centralized security baseline management.
- Consider running antivirus software on the SQL Server machine if appropriate.

3.4 Patching and Automated Windows Updates

The best way to ensure the security of the PViMS database server is to install security hotfixes and service packs as soon as possible. Use manual updates on an operating system basis by using Windows Update or Microsoft Update as updates should be tested before they are applied to the live PViMS server. All hotfixes should be installed immediately, and service packs should be tested and installed as soon as possible.

- Always stay as current as possible.
- Enable automatic updates whenever feasible but test them before applying to production systems.

3.5 Encryption

Best practices for data encryption

- Encrypt high-value and sensitive data.
- Use symmetric keys to encrypt data, and asymmetric keys or certificates to protect the symmetric keys.
- Password-protect keys and remove master key encryption for the most secure configuration.
- Do not delete pre-provisioned system certificates in the **master** database
- Always back up the service master key, database master keys, and certificates by using the key-specific DDL statements.
- Always back up your database to back up your symmetric and asymmetric keys.
- TDE is recommended for encrypting existing applications or for performance sensitive applications.
- Cell-level encryption can be used for defense in depth both for a database encrypted by TDE and for limited access control through the use of passwords.
- Use EKM with both database-level and cell-level encryption for more comprehensive key management and hardware-based cryptography.

3.6 Administrator permissions

Best practices for administrator privileges

- Use administrator privileges only when needed.

- Minimize the number of administrators.
- Have multiple distinct administrators if more than one is needed.
- Avoid dependency on the builtin\administrators Windows group.

3.7 Database Ownership

A SQL Server instance can contain multiple user databases. Each user database has a specific owner; the owner defaults to the database creator. By definition, members of the sysadmin server role are database owners (DBOs) in every user database. In addition, there is a database role, db_owner, in every user database. Members of the db_owner role have approximately the same privileges as the dbo user.

Therefore, several best practices should be implemented regarding these special ownerships:

- Minimize the number of users that have the db_owner role for each database.
- Have distinct owners for databases; not all databases should be owned by SA or by any other user in the sysadmin server role.
- Use user-defined server roles as an alternative to assigning server-level privileges to individual users.

3.8 Lockdown of System Stored Procedures

SQL Server ships with various system stored procedures such as xp_cmdshell or sp_send_dbmail that interact with the operating system or execute code outside of a normal SQL Server permission and may constitute additional security risks.

Therefore, several best practices should be implemented regarding these stored procedures:

- Disable **xp_cmdshell** unless it is absolutely needed
- Disable COM components once all COM components have been converted to SQLCLR
- Disable both mail procedures (Database Mail and SQL Mail) unless you need to send mail from SQL Server
- Use Policy-Based Management to enforce a standard policy for extended procedure usage
- Document each exception to the standard policy
- Do not remove the system stored procedures by dropping them
- Do not modify the default permissions on system objects
- Do not DENY all users/administrators access to the extended procedures

3.9 Authorisation

Authorisation is the process of granting permissions on securables to users. In SQL Server, securables are database objects. SQL Server principals include both instance-level principals, such as Windows logins, Windows group logins, SQL Server logins, and server roles and database-level principals, such as users, database roles, and application roles. Except for a few objects that are instance-scoped, most database objects, such as tables, views, and procedures are schema-scoped. This means that authorization is usually granted to database-level principals.

Best practices for database object authorization

- Manage permissions via database roles or Windows groups
- Use permission granularity to implement the principle of least privilege
- Do not enable **guest** access in any database
- Use users without logins instead of application roles

3.10 Authentication

SQL Server has two authentication modes: Windows Authentication and Mixed Mode Authentication. In Windows Authentication mode, specific Windows user and group accounts are trusted to log in to SQL Server. Windows credentials are used in the process; that is, either Kerberos or NTLM authentication credentials. Windows accounts use a series of encrypted messages to authenticate to SQL Server; no passwords are passed across the network during the authentication process. In Mixed Mode Authentication, both Windows accounts and SQL Server-specific accounts (known as SQL logins) are permitted. When SQL logins are used, SQL login passwords are passed across the network for authentication. This makes SQL logins less secure than Windows logins.

Best practices for authentication mode and logins

- Always use Windows Authentication mode if possible
- Use Mixed Mode Authentication only for legacy applications, non-Windows users, and users from untrusted domains
- Use the standard login DDL statements instead of the compatibility system procedures
- It the **sa** account is not going to be used, you should disable it. Change the **sa** account password to a known value if you might ever need to use it. Always use a strong password for the **sa** account and change the **sa** account password periodically
- Do not manage SQL Server by using the **sa** login account; assign **sysadmin** privilege to a knows user or group
- Rename the **sa** account to a different account name to prevent attacks on the **sa** account by name
- Do not delete internal built-in logins
- Use Windows Logins rather than Windows Group to control access to SQL Server and use care when using Windows Group logins to prevent group overlap for a particular user
- Use login triggers for more granular control of the login process

3.11 Network Security

A standard network protocol is required to connect to the SQL Server database. There are no internal connections that bypass the network.

SQL Server has introduced an abstraction for managing any connectivity channel—entry points into a SQL Server instance are all represented as endpoints. Endpoints exist for the following network client connectivity protocols:

- Shared Memory

- Named Pipes
- TCP/IP
- Dedicated administrator connection

Best practices for network connectivity

- Enable Windows Firewall and limit the network protocols supported
- Do not enable network protocols unless they are needed
- Disable NETBIOS and SMB protocol unless specifically needed
- Do not expose a server that is running SQL Server to the public Internet
- Configure named instances of SQL Server to use specific port assignments for TCP/IP rather than dynamic ports
- Grant CONNECT permission only on endpoints to logins that need to use them. Explicitly deny CONNECT permission to endpoints that are not needed by users or groups

4 Database Maintenance

4.1 Database Backups

SQL Server backup and restore operations occur within the context of the recovery model of the database. Recovery models are designed to control transaction log maintenance. A recovery model is a database property that controls how transactions are logged, whether the transaction log requires (and allows) backing up, and what kinds of restore operations are available. Three recovery models exist: simple, full, and bulk-logged. Typically, a database uses the full recovery model or simple recovery model. A database can be switched to another recovery model at any time.

Recovery model	Description	Work loss exposure	Recover to point in time?
Simple	No log backups. Automatically reclaims log space to keep space requirements small, essentially eliminating the need to manage the transaction log space.	Changes since the most recent backup are unprotected. In the event of a disaster, those changes must be redone.	Can recover only to the end of a backup.
Full	Requires log backups. No work is lost due to a lost or damaged data file. Can recover to an arbitrary point in time (for example, prior to application or user error).	Normally none. If the tail of the log is damaged, changes since the most recent log backup must be redone.	Can recover to a specific point in time, assuming that your backups are complete up to that point in time. Note: If you have two or more full-recovery-model databases that must be logically consistent, you may have to implement special procedures to make sure the recoverability of these databases.

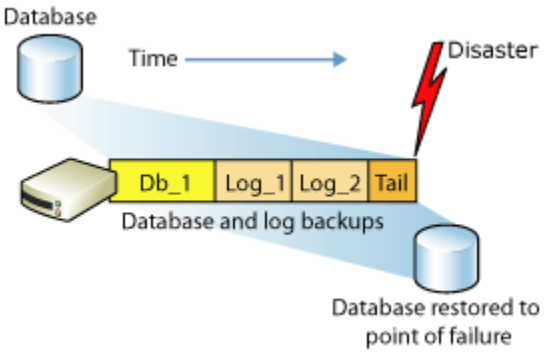
Recovery model	Description	Work loss exposure	Recover to point in time?
Bulk logged	Requires log backups. An adjunct of the full recovery model that permits high-performance bulk copy operations. Reduces log space usage by using minimal logging for most bulk operations.	If the log is damaged or bulk-logged operations occurred since the most recent log backup, changes since that last backup must be redone. Otherwise, no work is lost.	Can recover to the end of any backup. Point-in-time recovery is not supported.

4.1.1 Full Backups

The full recovery model uses log backups to prevent data loss in the broadest range of failure scenarios and backing and restoring the transaction log (log backups) is required.

The advantage of using log backups is that they let you restore a database to any point of time that is contained within a log backup (point-in-time recovery). You can use a series of log backups to roll a database forward to any point in time that is contained in one of the log backups. Be aware that to minimize your restore time, you can supplement each full back up with a series of differential backups of the same data.

Assuming you can back up the active log after a disaster occurs, you can restore the database up to the point of failure without data loss. The disadvantages of using log backups are that they require storage space and increase restore time and complexity.



The transaction log is a serial record of all the transactions that have been performed against the database since the

transaction log was last backed up. With transaction log backups, you can recover the database to a specific point in time (for example, prior to entering unwanted data), or to the point of failure.

Minimally, you must have created at least one full back up before you can create any log backups. After that, the transaction log can be backed up at any time unless the log is already being backed up. We recommend that you take log backups frequently, both to minimize work loss exposure and to truncate the transaction log. Typically, a database administrator creates a full database backup occasionally, such as weekly, and, optionally, creates a series of differential database backup at a shorter interval, such as daily. Independently of the database backups, the database administrator backs up the transaction log at frequent intervals, such as every 10 minutes. For a given type of backup, the optimal interval depends on factors such as the importance of the data, the size of the database, and the workload of the server.

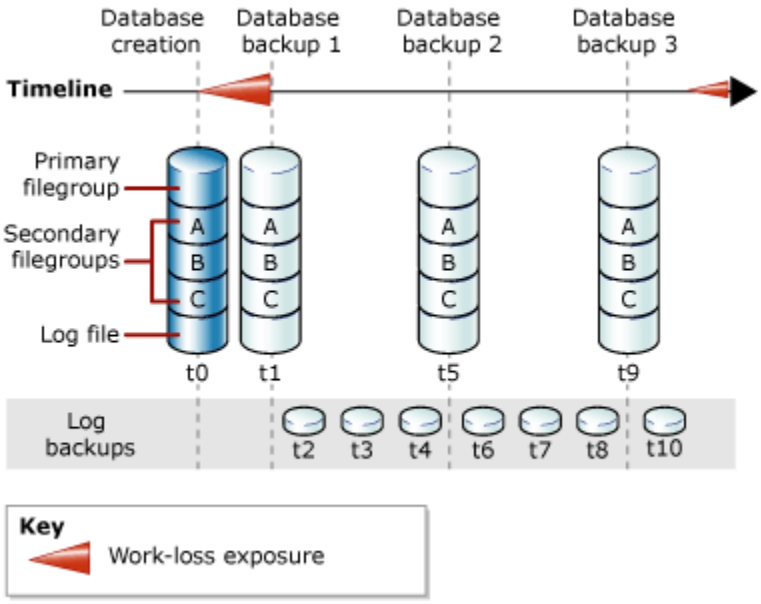
The sequence of transaction log backups is independent of the database backups. You make one sequence of transaction log backups, and then make periodic database backups that are used to start a restore operation. For example, assume the following sequence of events.

Time	Event
8:00 A.M.	Back up database
Noon	Back up transaction log
4:00 P.M.	Back up transaction log
6:00 P.M.	Back up database
8:00 P.M.	Back up transaction log
10:00 P.M.	Failure occurs

Data Server	System Failure	Secure equipment room, backup server, UPS	Switch over to backup server, validate that UPS is running	Fix/replace primary server, fail back to primary server
	Corrupted DB	To minimize data corruption, it is recommended to use a db analyse utility regularly. Prevent power outages. Ensure that there is enough disk space	You need to determine the extent of the corruption so that you can figure out what your options are in terms of restore or repair (or potentially just fail over and deal with the corruption offline).	Restore or repair DB

	Data Loss	Implement RAID system, fault tolerance, and ensure that entire array is available. Check logs, activate automatic notifications so that alerts can be monitored	Restore application data and VM from backups	In the event that a disk has failed, replace disk so that entire array is available

For databases that use full and bulk-logged recovery, database backups are necessary but not sufficient. Transaction log backups are also required. The following illustration shows the least complex backup strategy that is possible under the full recovery model.



4.2 Maintenance Plans

Maintenance plans can be created to perform the following tasks:

- Reorganize the data on the data and index pages by rebuilding indexes with a new fill factor. Rebuilding indexes with a new fill factor makes sure that database pages contain an equally distributed amount of data and free space. It also enables faster growth in the future
- Compress data files by removing empty database pages

- Update index statistics to make sure the query optimizer has current information about the distribution of data values in the tables. This enables the query optimizer to make better judgments about the best way to access data, because it has more information about the data stored in the database
- Perform internal consistency checks of the data and data pages within the database to make sure that a system or software problem has not damaged data
- Back up the database and transaction log files. Database and log backups can be retained for a specified period. This lets you create a history of backups to be used if you have to restore the database to a time earlier than the last database backup. You can also perform differential backups

4.2.1 Preparing the Maintenance Plan

All maintenance plans are implemented using Ola Hallengren's SQL Server Maintenance Solution which is licensed under the MIT license, a populate widely used open source license.

Please note, this section assumes you are using MS SQL Express which does not have an inbuilt SQL Server Agent to execute jobs created by the MaintenanceSolution automatically. If you are using the Standard or Enterprise Editions of SQL Server, you will not need to create a window task to execute these maintenance plans.

- Create two folders on the SQL server – **C:\AppData\Backups** and **C:\AppData\Reports** (these folders do not need to be created on the c drive, this can be adjusted as necessary)
- Download MaintenanceSolution.sql, which is a script that creates all objects needed to facilitate the backup process.
- Save this file in the folder **X:\AppData**
- Open this file using SQL Server Management Studio (log in as an administrator)
- Locate the following lines of code near the top of the script (around line 32)

```
SET @CreateJobs           = 'Y'
SET @BackupDirectory      = NULL
SET @CleanupTime          = NULL
SET @OutputFileDirectory  = NULL
SET @LogToTable           = 'Y'
```

- Amend these variables as follows:

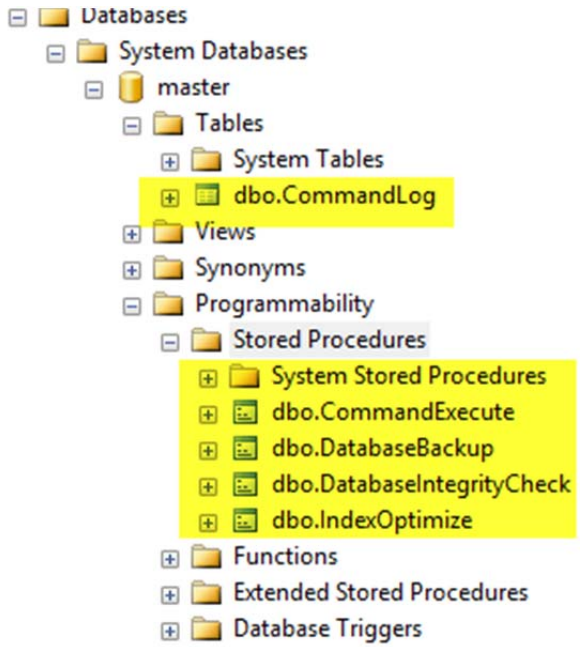
Variable	Modification
CreateJobs	Change to **'N'** if you are using SQL Server Express else leave as **'Y'.** **Please note,** ensure the SQL Server Agent is started before executing this stored procedure if you have set this variable to **'Y'**
BackupDirectory	Replace the value of the BackupDirectory variable (NULL) with the preferred backup path, **C:\AppData\Backups**. The provided path will be used as a root directory for storing backup files
CleanupTime	Set the integer value for the CleanupTime. This value specifies how long (in hours) will each backup file be kept on the drive. After the specified time passes, the file will be deleted automatically. If default NULL value is left in the script, the backup files will never be deleted automatically. Set this value to **120 (5 days).**
OutputFileDirectory	Replace the value of the OutputFileDirectory variable (NULL) with the preferred backup path, **C:\AppData\Reports. T**his variable specifies the path for the log files that will be created after each job is run.

LogToTable	Default value is set to Yes, and leaving this value as is will log the results of each created job in the CommandLog table

- Execute the MaintenanceSolution.sql against the server. The script creates one table (dbo.CommandLog), one function, and four stored procedures in the master database. It also creates 11 pre-created SQL Server Agent jobs if you have set the CreateJobs parameter to **'Y'**

Please ensure you receive a Command(s) completed successfully message once you have executed the step above

The following table and stored procedures are created



Object	Description
CommandLog Table	Used for logging
CommandExecute Stored Procedure	Core component of the Maintenance Solution as it is used by all other procedures and jobs
DatabaseBackup	Manages all backup tasks
DatabaseIntegrityCheck	Manages all integrity tasks
IndexOptimise	Manages index optimization tasks

4.2.2 Preparing the Database Backup Task (System Databases)

This section uses the Database Backup Stored Procedure created in section 4.2.1.

Please note, this section assumes you are using MS SQL Express which does not have an inbuilt SQL Server Agent to execute jobs created by the Maintenance Solution. If you are using the Standard or Enterprise Editions of SQL Server, you will not need to create a windows task to execute these maintenance plans, instead schedule the appropriate Database Backup job using SQL Server Agent. If you are using the SQL Server Express Edition, please follow the steps outlined below.

Please ensure you have access to the Core.PViMS.Backup.zip file provided as part of system implementation. Extract contents of this zip file to the **C:\AppData** folder on the SQL Server.

STEP 1: Prepare the SQL Script

Please note, the file **backup_all_sys_dbs.sql** Is used to execute the DatabaseBackup stored procedure and contains all the configuration parameters needed to backup the Master Database as part of this process.

Variables

@Databases	SYSTEM_DATABASES	All system databases (master, msdb, and model)
@Directory	C:\AppData\Backups	Backup to the following directory
@BackupType	FULL	Do a full backup
@Verify	Y	Verify the backup after it has completed
@CheckSum	Y	Enable backup checksums
@CleanupTime	120	Set for 5 days
@CleanupMode	AFTER_BACKUP	Remove old backup files after successful completion of the backup

- Browse to the **C:\AppData** folder and open **backup_all_sys_dbs.sql** using SQL Server Management Studio (log in as an administrator)
- Amend these variables as follows:

Variable	Modification
BackupDirectory	Replace the value of the BackupDirectory variable with the preferred backup path, **C:\AppData\Backups**. The provided path will be used as a root directory for storing backup files
CleanupTime	Set the integer value for the CleanupTime. This value specifies how long (in hours) will each backup file be kept on the drive. After the specified time passes, the file will be deleted automatically. If default NULL value is left in the script, the backup files will

	never be deleted automatically. Set this value to **120 (5 days).**

STEP 2: Prepare the Command File

Please note, the file **full_weekly_backup.cmd** Is used to connect to the SQL database using the SQLCMD program and execute the **backup_all_sys_dbs.sql** script.

SQLCmd Parameters

-E	Use a trusted SQL connection
-S	C:\AppData\Backups
-d	The name of the database (master)
-i	The input file name **(backup_all_sys_dbs.sql**)
-o	The output file name

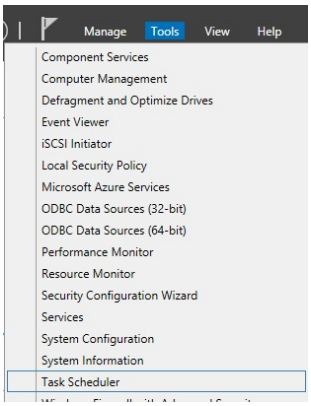
- Browse to the **C:\AppData** folder and open **full_weekly_backup.cmd** using notepad
- Amend these variables as follows:

Variable	Modification
SERVERANDINSTANCENAME	Specify the name of the Windows Server, followed by a backslash followed by the name of the SQL Server Instance and then specify the port number if required **Example** .\SQLEXPRESS PVIMSSERVER\SQLEXPRESS .\SQLEXPRESS,1433
OUTPUTFILENAME	The name of the output file. Reports\FullWeeklyBackup_Log.txt

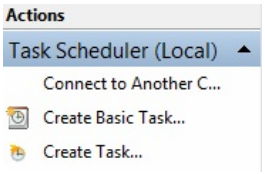
STEP 3: Create a Weekly Task

Please note, this task must be created on the same server as the files that have been edited in step 1 and 2. Please ensure you are logged in to the server as an administrator.

Open Server Manager → Tools → Task Scheduler

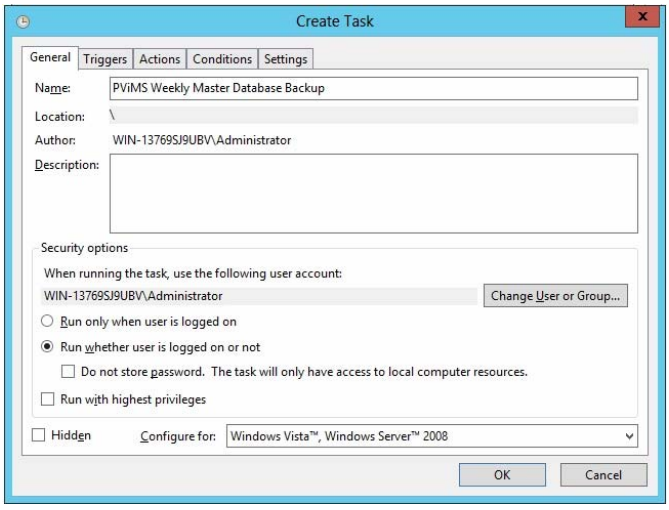


Create a new task



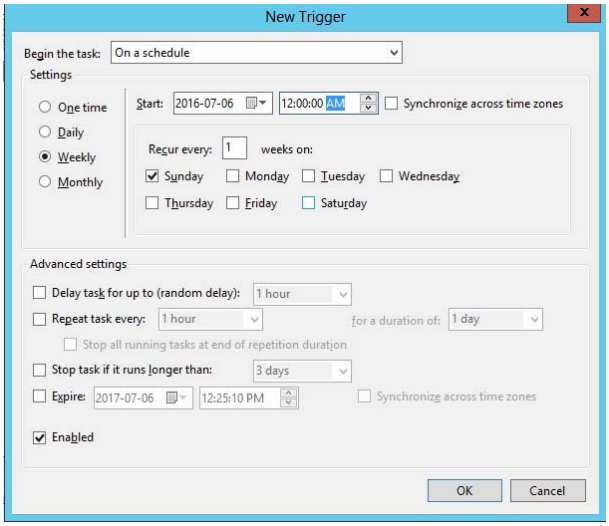
General Settings

- Enter a task name | **PViMS Weekly Master Database Backup**
- Ensure Run when user is logged on or not is selected

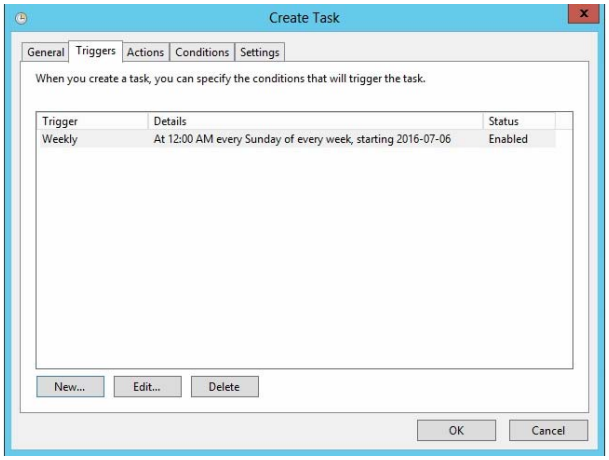


Trigger Settings

- Add a new trigger
  - Ensure task is configured to run weekly every Sunday
  - Ensure task is configured to start on the current date plus one day
  - Ensure task is configured to start at 12:00 AM
  - Under advanced settings, ensure task is enabled
  - Click OK to add trigger

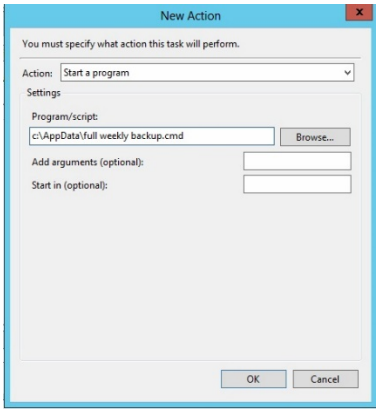


Ensure trigger appears in trigger list

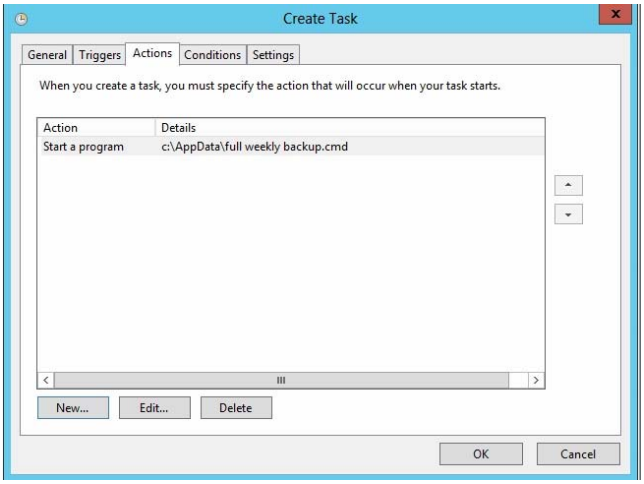


Action Settings

- Add a new action
  - Ensure Start a Program is select
  - Browse to the location of the **full_ weekly_backup.cmd** file in **C:\AppData**
  - Click OK to add action



Ensure action appears in the list
Click OK to add this task to the server. You may be prompted to confirm your administrator user name and password to add the task.



4.2.3 Preparing the Database Backup Task (PViMS Database)

This section uses the Database Backup Stored Procedure created in section 4.2.1.

Please note, this section assumes you are using MS SQL Express which does not have an inbuilt SQL Server Agent to execute jobs created by the Maintenance Solution. If you are using the Standard or Enterprise Editions of SQL Server, you will not need to create a windows task to execute these maintenance plans, instead schedule the appropriate Database Backup job using SQL Server Agent. If you are using the SQL Server Express Edition, please follow the steps outlined below.

Please ensure you have access to the Core.PViMS.Backup.zip file provided as part of system implementation. Extract contents of this zip file to the **C:\AppData** folder on the SQL Server.

STEP 1: Prepare the SQL Script

Please note, the file **backup_pvims_db.sql** Is used to execute the DatabaseBackup stored procedure and contains all the configuration parameters needed to backup the PVIMS Database as part of this process.

Variables

@Databases	PViMS	Backup the PViMS database (specify USER_DATABASES to backup all databases)
@Directory	C:\AppData\Backups	Backup to the following directory
@BackupType	FULL	Do a full backup
@Verify	Y	Verify the backup after it has completed
@CheckSum	Y	Enable backup checksums
@CleanupTime	120	Set for 5 days
@CleanupMode	AFTER_BACKUP	Remove old backup files after successful completion of the backup

- Browse to the **C:\AppData** folder and open **backup_pvims _db.sql** using SQL Server Management Studio (log in as an administrator)
- Amend these variables as follows:

Variable	Modification
Databases	Replace the value of the Databases variable with the correct database name if the default name has been changed
BackupDirectory	Replace the value of the BackupDirectory variable (NULL) with the preferred backup path, **C:\AppData\Backups**. The provided path will be used as a root directory for storing backup files

CleanupTime	Set the integer value for the CleanupTime. This value specifies how long (in hours) will each backup file be kept on the drive. After the specified time passes, the file will be deleted automatically. If default NULL value is left in the script, the backup files will never be deleted automatically. Set this value to **120 (5 days).**

STEP 2: Prepare the Command File

Please note, the file **full_daily_backup.cmd** Is used to connect to the SQL database using the SQLCMD program and execute the **backup_pvims_db.sql** script.

SQLCmd Parameters

-E	Use a trusted SQL connection
-S	C:\AppData\Backups
-d	The name of the database (master)
-i	The input file name **(backup_pvims_db.sql**)
-o	The output file name

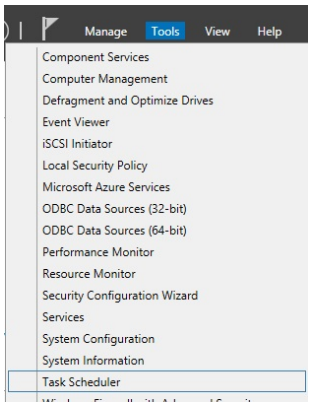
- Browse to the **C:\AppData** folder and open **full_daily_backup.cmd** using notepad
- Amend these variables as follows:

Variable	Modification
SERVERANDINSTANCENAME	Specify the name of the Windows Server, followed by a backslash followed by the name of the SQL Server Instance and then specify the port number if required **Example** .\SQLEXPRESS PVIMSSERVER\SQLEXPRESS .\SQLEXPRESS,1433
OUTPUTFILENAME	The name of the output file. Reports\FullDailyBackup_Log.txt

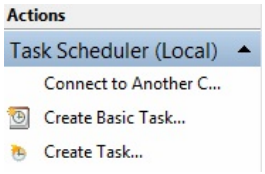
STEP 3: Create a Daily Task

Please note, this task must be created on the same server as the files that have been edited in step 1 and 2. Please ensure you are logged in to the server as an administrator.

Open Server Manager → Tools → Task Scheduler

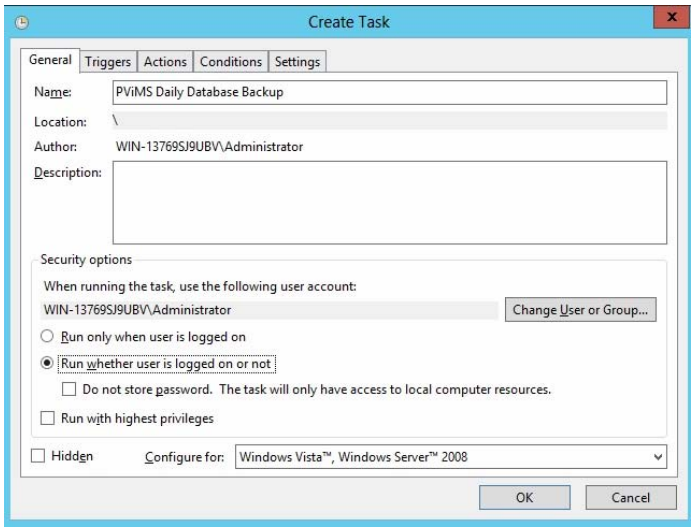


Create a new task



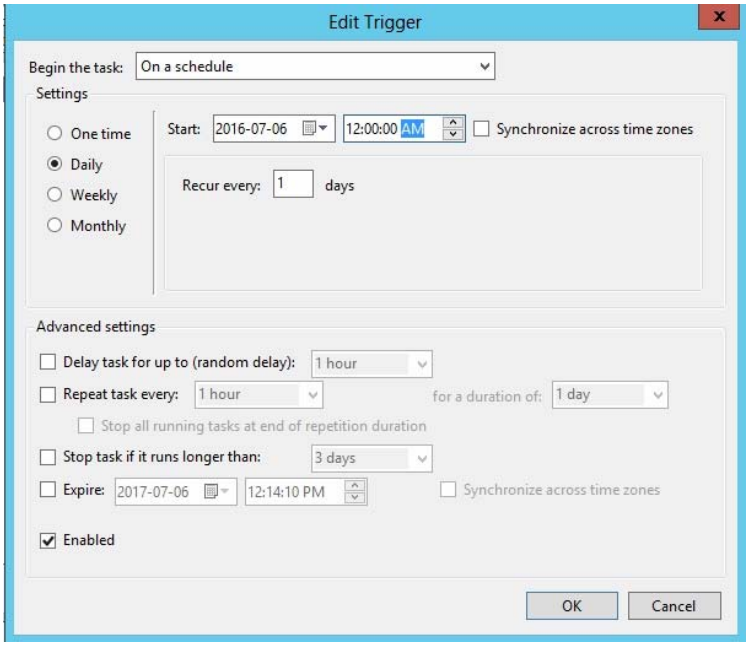
General Settings

- Enter a task name | **PViMS Daily Database Backup**
- Ensure Run when user is logged on or not is selected

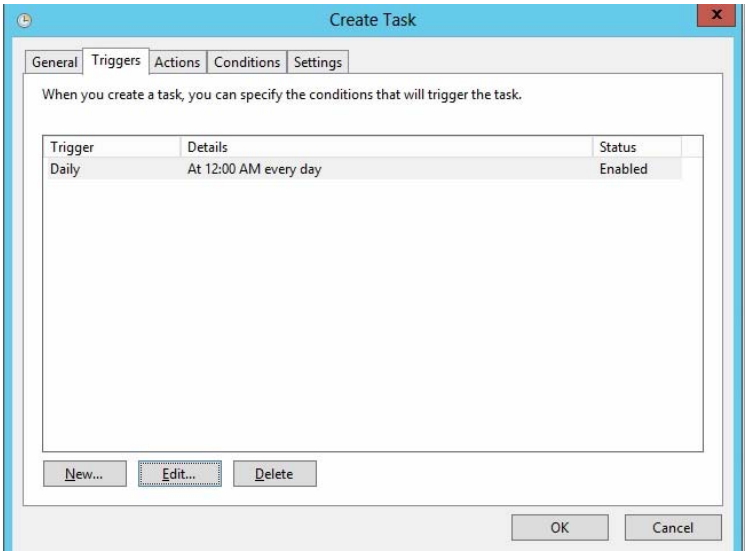


Trigger Settings

- Add a new trigger
  - Ensure task is configured to run daily
  - Ensure task is configured to start on the current date plus one day
  - Ensure task is configured to start at 12:00 AM
  - Under advanced settings, ensure task is enabled
  - Click OK to add trigger

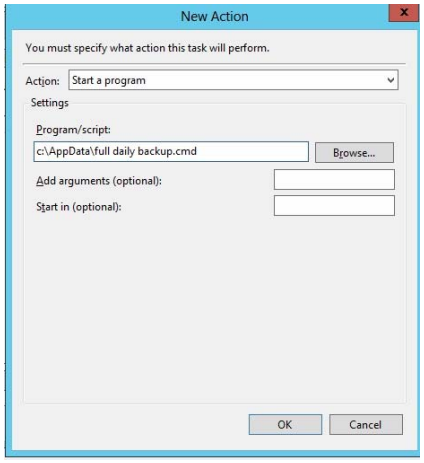


Ensure trigger appears in trigger list

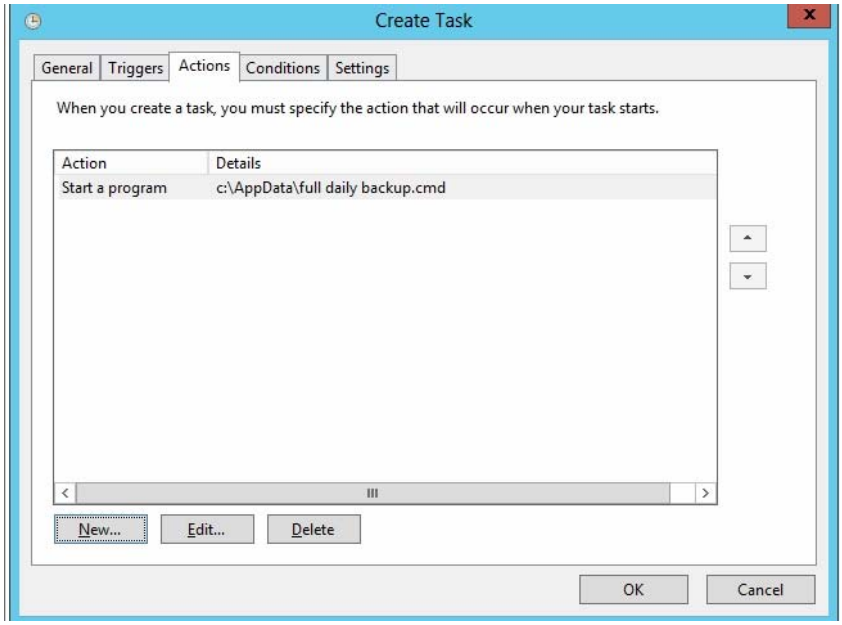


Action Settings

- Add a new action
    - Ensure Start a Program is select
    - Browse to the location of the **full_daily_backup.cmd** file in **C:\AppData**
    - Click OK to add action



Ensure action appears in the list
Click OK to add this task to the server. You may be prompted to confirm your administrator user name and password to add the task.



4.2.4 Preparing the Database Integrity Task (PViMS Database)

This section uses the Database Integrity Check Stored Procedure created in section 4.2.1.

Please note, this section assumes you are using MS SQL Express which does not have an inbuilt SQL Server Agent to execute jobs created by the Maintenance Solution. If you are using the Standard or Enterprise Editions of SQL Server, you will not need to create a windows task to execute these maintenance plans, instead schedule the appropriate Database Backup job using SQL Server Agent. If you are using the SQL Server Express Edition, please follow the steps outlined below.

Please ensure you have access to the Core.PViMS.Backup.zip file provided as part of system implementation. Extract contents of this zip file to the **C:\AppData** folder on the SQL Server.

STEP 1: Prepare the SQL Script

Please note, the file **integrity_pvims_db.sql** Is used to execute the Database Integrity Check stored procedure and contains all the configuration parameters needed to integrity check the PVIMS Database as part of this process.

Variables

@Databases	PViMS	Verify the integrity of the PViMS database (specify USER_DATABASES to verify all user databases)
@CheckCommands	CHECKDB	Check the integrity of the whole database

- Browse to the **C:\AppData** folder and open **integrity_pvims_db.sql** using SQL Server Management Studio (log in as an administrator)
- Amend these variables as follows:

Variable	Modification
Databases	Replace the value of the Databases variable with the correct database name if the default name has been changed

STEP 2: Prepare the Command File

Please note, the file **full_weekly_integrity.cmd** Is used to connect to the SQL database using the SQLCMD program and execute the **integrity_pvims_db.sql** script.

SQLCmd Parameters

-E	Use a trusted SQL connection
-S	C:\AppData\Backups

-d	The name of the database (master)
-i	The input file name **(integrity_pvims_db.sql**)
-o	The output file name

- Browse to the **C:\AppData** folder and open **full_weekly_integrity.cmd** using notepad
- Amend these variables as follows:

Variable	**Modification**
SERVERANDINSTANCENAME	Specify the name of the Windows Server, followed by a backslash followed by the name of the SQL Server Instance and then specify the port number if required **Example** .\SQLEXPRESS PVIMSSERVER\SQLEXPRESS .\SQLEXPRESS,1433
OUTPUTFILENAME	The name of the output file. Reports\FullWeeklyIntegrity_Log.txt

STEP 3: Create a Weekly Task

Please note, this task must be created on the same server as the files that have been edited in step 1 and 2. Please ensure you are logged in to the server as an administrator.

Please see step 3 for preparing the Database Backup Task (System Databases) for steps on how to create a weekly task to execute this task.

4.2.5 Preparing the Index Optimise Task (PViMS Database)

This section uses the Index Optimize Stored Procedure created in section 4.2.1.

Please note, this section assumes you are using MS SQL Express which does not have an inbuilt SQL Server Agent to execute jobs created by the Maintenance Solution. If you are using the Standard or Enterprise Editions of SQL Server, you will not need to create a windows task to execute these maintenance plans, instead schedule the appropriate Database Backup job using SQL Server Agent. If you are using the SQL Server Express Edition, please follow the steps outlined below.

Please ensure you have access to the Core.PViMS.Backup.zip file provided as part of system implementation. Extract contents of this zip file to the **C:\AppData** folder on the SQL Server.

STEP 1: Prepare the SQL Script

Please note, the file **optimise_pvims_db.sql** Is used to execute the Index Optimise stored procedure and contains all the configuration parameters needed to optimise the PVIMS Database as part of this process.

Variables

@Databases	PViMS	Optimise the PViMS database (specify USER_DATABASES to optimise all user databases)
@FragmentationLow	NULL	Do not defragment indexes with low fragmentation
@FragmentationMedium	INDEX_REORGANIZE INDEX_REBUILD_ONLINE INDEX_REBUILD_OFFLINE	Reorganise and rebuild indexes with medium fragmentation
@FragmentationHigh	INDEX_REBUILD_ONLINE INDEX_REBUILD_OFFLINE	Reorganise and rebuild indexes with high fragmentation
@FragmentationLevel1	5	Set the lower limit, as a percentage, for medium fragmentation. The default is 5 percent. This is based on Microsoft's recommendation
@FragmentationLevel2	30	Set the lower limit, as a percentage, for high fragmentation. The default is 30 percent. This is based on Microsoft's recommendation
@UpdateStatistics	ALL	Update index and column statistics
@OnlyModifiedStatistics	Y	Update statistics only if any rows have been modified since the most recent statistics update

- Browse to the **C:\AppData** folder and open **optimise_pvims _db.sql** using SQL Server Management Studio (log in as an administrator)
- Amend these variables as follows:

Variable	Modification
Databases	Replace the value of the Databases variable with the correct database name if the default name has been changed

STEP 2: Prepare the Command File

Please note, the file **full_weekly_optimise.cmd** Is used to connect to the SQL database using the SQLCMD program and execute the **optimise_pvims_db.sql** script.

SQLCmd Parameters

-E	Use a trusted SQL connection
-S	C:\AppData\Backups
-d	The name of the database (master)
-i	The input file name **(optimise_pvims_db.sql**)
-o	The output file name

- Browse to the **C:\AppData** folder and open **full_weekly_optimise.cmd** using notepad
- Amend these variables as follows:

Variable	Modification
SERVERANDINSTANCENAME	Specify the name of the Windows Server, followed by a backslash followed by the name of the SQL Server Instance and then specify the port number if required **Example** .\SQLEXPRESS PVIMSSERVER\SQLEXPRESS .\SQLEXPRESS,1433
OUTPUTFILENAME	The name of the output file. Reports\FullWeeklyOptimise_Log.txt

STEP 3: Create a Weekly Task

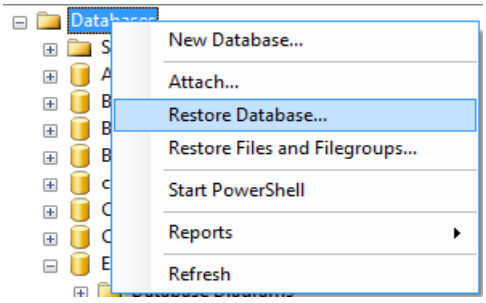
Please note, this task must be created on the same server as the files that have been edited in step 1 and 2. Please ensure you are logged in to the server as an administrator.

Please see step 3 for preparing the Database Backup Task (System Databases) for steps on how to create a weekly task to execute this task.

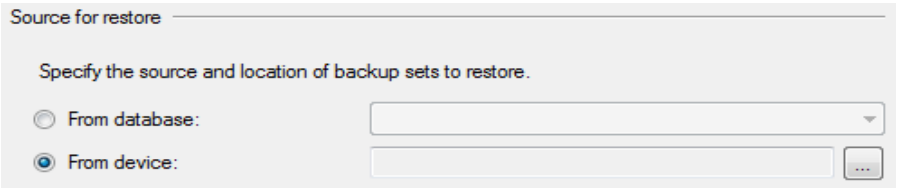
5 Database Restore

This section describes how to restore a database backup onto the PViMS SQL Server.

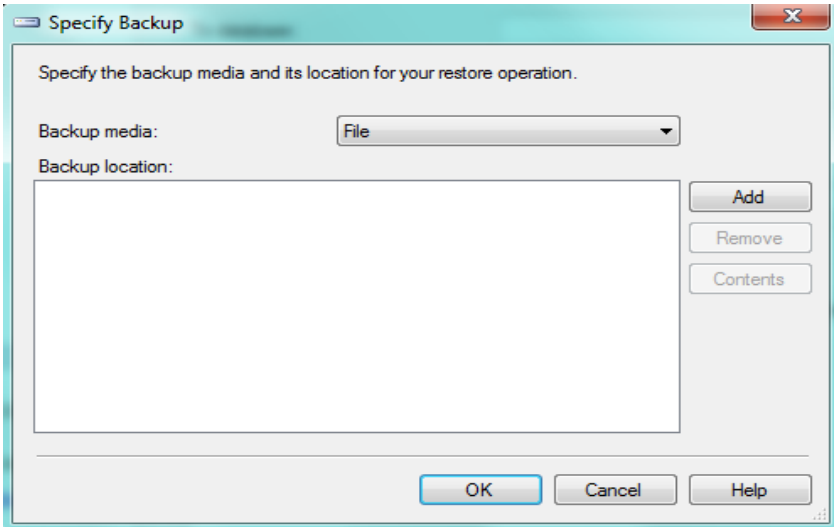
Open SQL Server Management Studio and login as an administrator. Once logged in, right click on the Databases node and select 'Restore Database'.



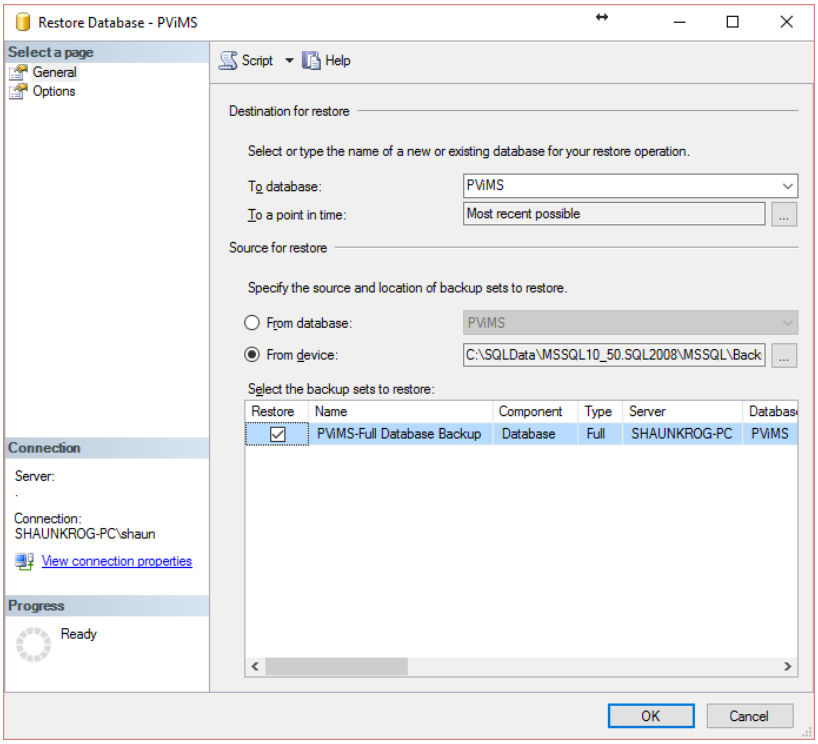
- Click the ellipses button next to 'From device' under the 'Source for restore' section.



- Set 'File' as the backup media and then click 'Add'.

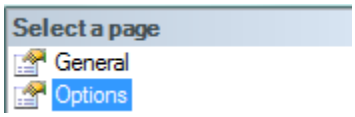


- Browse to the SQL backup (BAK) file you want to restore.
- In the Restore Database dialog, type or select the name of the database you want this backup restored to.
- If you select an existing database, this database will be replaced by the contents of the backup
- If you type a database name which does not currently exist in your SQL Server installation, a new database will be created.
- Next, select the restore point you want to use. Since a SQL backup file can hold multiple backups you may see more than one restore point listed.



At this point, enough information has been entered for the database to be restored. However, SQL backup files store information about where data files are copied so if there are any file system problems such as the destination directory not existing or conflicting data file names an error will occur. These problems are common when restoring a backup created on a different SQL Server installation.

To review and change the file system settings, click the Options page on the left in the Restore Database dialog.

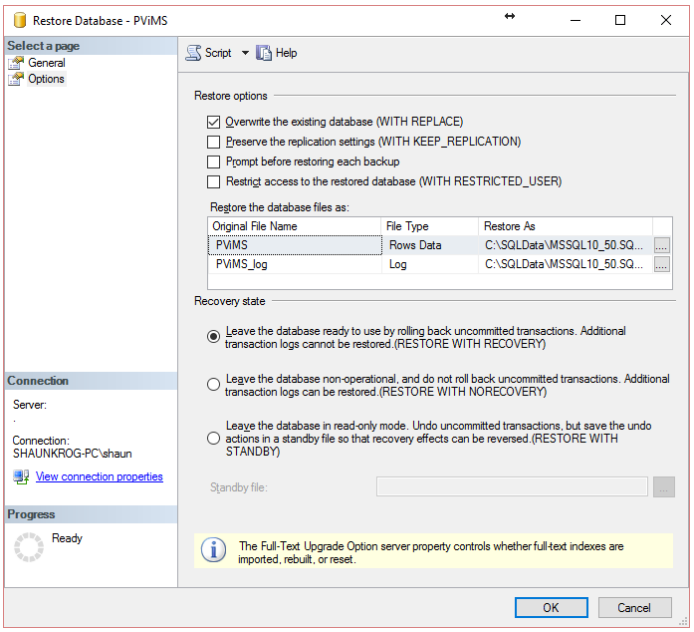


On the options page, you will want to make sure the 'Restore As' column points to valid folder locations (you can change them as needed). These files do not have to exist; however, the folder path must exist. If the respective files do exist, SQL Server follows a simple set of rules:

If the 'To database' (from the General page) matches the restore database backup (i.e. restoring to matching databases), the respective files will be overwritten as part of the restore.

If the 'To database' does not match the restore database backup (i.e. restoring to a different database), the 'Overwrite the existing database' will need to be checked for the restore process to complete. Use this function with caution as you can potentially restore database backup information on top of data files from a completely different database.

Generally, you can tell the databases differ based on the 'Original File Name' which is the internal name SQL Server uses to reference the respective files.



Once restore options are set, click Ok and the database will be restored.