DataDictionary

AGNIS Version 2.1.0

2013-03-25

Credits and Resources

AGNIS Development and Management Teams		
Executive Committee	**Technical Committee**	**Software Development**
Dennis Confer, M.D. [1] Mary Horowitz, M.D. [2] Martin Maiers [1] Barbara McGary [2] Douglas Rizzo, M.D. [2] Paul Zyla [1]	Ken Bengtsson [1] Steve Finch [1] Barbara McGary [2] Joel Schneider [9] John Sheets [9] Neil Smeby [1]	Steve Finch [1] Jim Gaudette [1] Barb Kramer [1] Anh Nguyen [1] Kirt Schaper [1] Joel Schneider [9]
Advisory Committee	**Metadata Curation**	**Support (Systems, QA, Documentation)**
Jane Apperley, M.D. [4] Richard Champlin, M.D. [6] Jeremy Chapman, M.D. [5] Margaret MacMillan, M.D. [8] Carlheinz Müller, M.D., Ph.D. [3] Ricardo Pasquini, M.D. [4] Colette Raffoux, M.D. [3] Olle Ringden, M.D., Ph.D. [6] Gerard Socie, M.D., Ph.D. [4] Jeff Szer, M.D. [4] Daniel Weisdorf, M.D. [7] John Wingard, M.D. [7]	Robinette Aley [1] Mary Cooper [10] Tommie Curtis [10] Suzette Czech [2] Jon Iversen [1] Roy Jones, M.D. [11] Thomas Joshua [2] Angela Kummerow [2] Jocelyn Leatherwood [10] Betty Lee [10] Alex Morozoff [2] Michele Nych [2] Dianne Reeves [10] Douglas Rizzo, M.D. [2] Lorenzo Scott [2] Jeremey Sturgill [2] Marishia Qualls-May [2] Wendy Zhang [2]	Robinette Aley [1] Raj Bikkina [1] Jason Brelsford [1] Barb Kramer [1] David Lee [1] Mark MacLennan [1] Ravi Puppala [1] Kirt Schaper [1] Joel Schneider [9] David Zokaites [1]

[1] National Marrow Donor Program
[2] Center for International Blood & Marrow Transplantation Research (CIBMTR)
[3] European Marrow Donor Information System (EMDIS)
[4] European Group for Blood and Marrow Transplantation (EBMT)
[5] Australian Bone Marrow Donor Registry (ABMDR)
[6] International Bone Marrow Transplant Registry / Autologous Blood and Marrow Transplant Registry (IBMTR/ABMTR)
[7] Blood and Marrow Transplant Clinical Trials Network (BMT CTN)
[8] University of Minnesota
[9] McCaa, Webster, & Associates
[10] National Cancer Institute Center for Biomedical Informatics and Information Technology (NCICBIIT)
[11] M.D. Anderson Cancer Center

Other Acknowledgements
caCORE - Project - National Cancer Institute Center for Biomedical Informatics and Information Technology (NCICBIIT)
caDSR - Project - National Cancer Institute Center for Biomedical Informatics and Information Technology (NCICBIIT)
caGrid - Project - National Cancer Institute Center for Biomedical Informatics and Information Technology (NCICBIIT)
FormsNet - Project - National Marrow Donor Program (NMDP)
Globus Toolkit - Project - Globus Alliance

Contacts and Support	
Web Site	http://www.agnis.net/
Mailing List	agnis@googlegroups.com http://groups.google.com/group/agnis
Electronic Mail	agnis@nmdp.org

Table of Contents

Table of Contents

Table of Contents

Table of Contents

Data Dictionary

Purpose

AGNIS® is A Growable Network Information System®.

The purpose of this document is to act as a data dictionary for common elements shared by the AGNIS Repository Database and the AGNIS Staging Database. Additionally, this document includes the XML schema which defines data structures passed between AGNIS clients and the AGNIS server.

Document Revision History

Date	Version	Description	AGNIS Version
2007-12-06	1.00	Initial version.	1.1.0
2008-04-02	1.01	Revise introduction sections at beginning of document, update XML schema to include Question.delete attribute.	1.2.0
2009-06-08	2.00	Revise data structure for AGNIS 2.0. Add value meaning public id, value meaning version, lock string, and lock override attributes, and make changes needed for addition of module class.	2.0.0
2009-06-29	2.01	Remove Module.delete attribute.	2.0.0
2010-03-26	2.02	Revise data structure to support enhanced (originator mode) staging client. Modify InformationProcess class to add process type, process status, processing rank, start time, end time, subscriber unique name, originator unique name, publisher unique name, retrieval sequence number, and service URL attributes. Modify ProcessingStatus class to add processing rank, attempt number, output form revision, and parent processing status attributes. Change length of alternate metadata columns to 255 in FORM, MODULE, QUESTION, and TEST_QUESTION tables. Change length of status (or value) columns to 30 in RETRIEVAL, FORM_REVISION, and PROCESSING_STATUS tables. Add LOCK_OVERRIDE_FLAG column to FORM_REVISION table. Fix problem that caused data model diagram to appear blurry. Revise layout of class and data model diagrams for improved consistency and readability. Add Microsoft SQL Server and Sybase DDL.	2.0.2
2010-05-13	2.03	Add FormRevision diagnosticMessage attribute to UML class diagram and XML schema.	2.1.0

Trademark Information

A Growable Network Information System and AGNIS are registered trademarks of the National Marrow Donor Program.

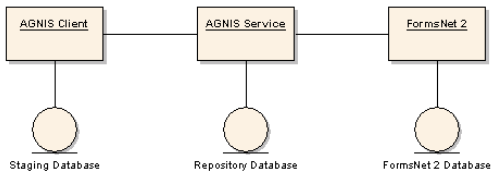
National Marrow Donor Program and NMDP are registered trademarks of the National Marrow Donor Program.

FormsNet is a trademark of the National Marrow Donor Program.

Other product or company names mentioned herein are the trademarks of their respective owners.

Overview

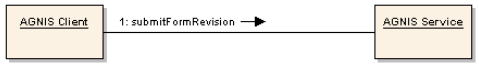
An AGNIS web service can act as a channel for electronic transmission of forms data between several end points. The AGNIS system at the National Marrow Donor Program (NMDP) interacts with the internal FormsNet 3 system and with external AGNIS clients such as transplant centers and the Center for International Blood & Marrow Transplantation Research (CIBMTR) in Milwaukee:



There are three different possible roles which may be assumed by an AGNIS client: originator, publisher, and subscriber. The client's interaction with the AGNIS service is slightly different for each of these roles.

Originator Role

In the originator role, the client transmits forms data to the AGNIS service for processing by a back end system such as FormsNet 3.



A transplant center acts as an originator when electronically transmitting data to FormsNet 3 via AGNIS.

Publisher Role

In the publisher role, the client transmits forms data to the AGNIS service for storage in the AGNIS repository, a database attached to the AGNIS service.

The FormsNet 3 system acts as a publisher when sending completed forms data to AGNIS.

The AGNIS service stores the published forms data in a database which follows the schema described in this document.

Subscriber Role

In the subscriber role, the client retrieves forms data from the AGNIS service.

CIBMTR Milwaukee acts as a subscriber when retrieving forms data from AGNIS.

The currently available AGNIS software provides a StagingClient program capable of retrieving data from the AGNIS service and storing them in a local staging database which follows the schema described in this document.

Web Service Operations

The table below divides AGNIS web service operations into three categories associated with the three client roles.

Role	Task	Associated AGNIS Web Service Operations
Publisher	publish a completed form	`publishCompletedFormRevision`
Originator	submit a form electronically	`submitFormRevision`

Subscriber	retrieve multiple forms	`acknowledgeRetrieval`
		`getNewlyCompletedFormRevisionQuantity`
		`retrieveNewlyCompletedFormRevisions`

	retrieve a specific form	`acknowledgeSingleRetrievedFormRevision` `retrieveSingleCompletedFormRevision`

The `publishCompletedFormRevision`, `submitFormRevision` operations are relatively self explanatory. The retrieval process is slightly more complicated in that the AGNIS service requires an explicit acknowledgement before it considers a retrieval to be completed.

For further detail on each operation, refer to the AGNIS Service document, or the Web Service Description Language (WSDL) associated with the AGNIS web service.

Metadata Repository

Forms data transmitted via AGNIS are defined as Case Report Forms (CRFs) in a metadata repository operated by the National Cancer Institute Center for Biomedical Informatics and Information Technology (NCICBIIT), known as the Cancer Data Standards Registry and Repository (caDSR). The caDSR is an ISO/IEC 11179 compliant metadata repository, containing detailed information about data elements used within the forms. The CRF definitions in the caDSR are intended to act as a standard for transmitting this data.

The caDSR form definitions can be viewed using the publicly accessible web-based Form Builder tool found here:

- https://formbuilder.nci.nih.gov/

Guest users of Form Builder can log in with the username "guest" and password shown on the login page. Forms used by the NMDP hosted AGNIS service are listed under the National Heart, Lung and Blood Institute (NHLBI) context.

Model Driven Design

The classes and database tables for AGNIS were initially developed using model driven tools provided by the cancer Common Ontologic Representation Environment (caCORE) SDK and the caGrid project, as a step toward the eventual goal of having AGNIS qualify as a silver or gold level cancer Biomedical Informatics Grid (caBIG) compliant system.

Links to additional information on caCORE and caGrid:

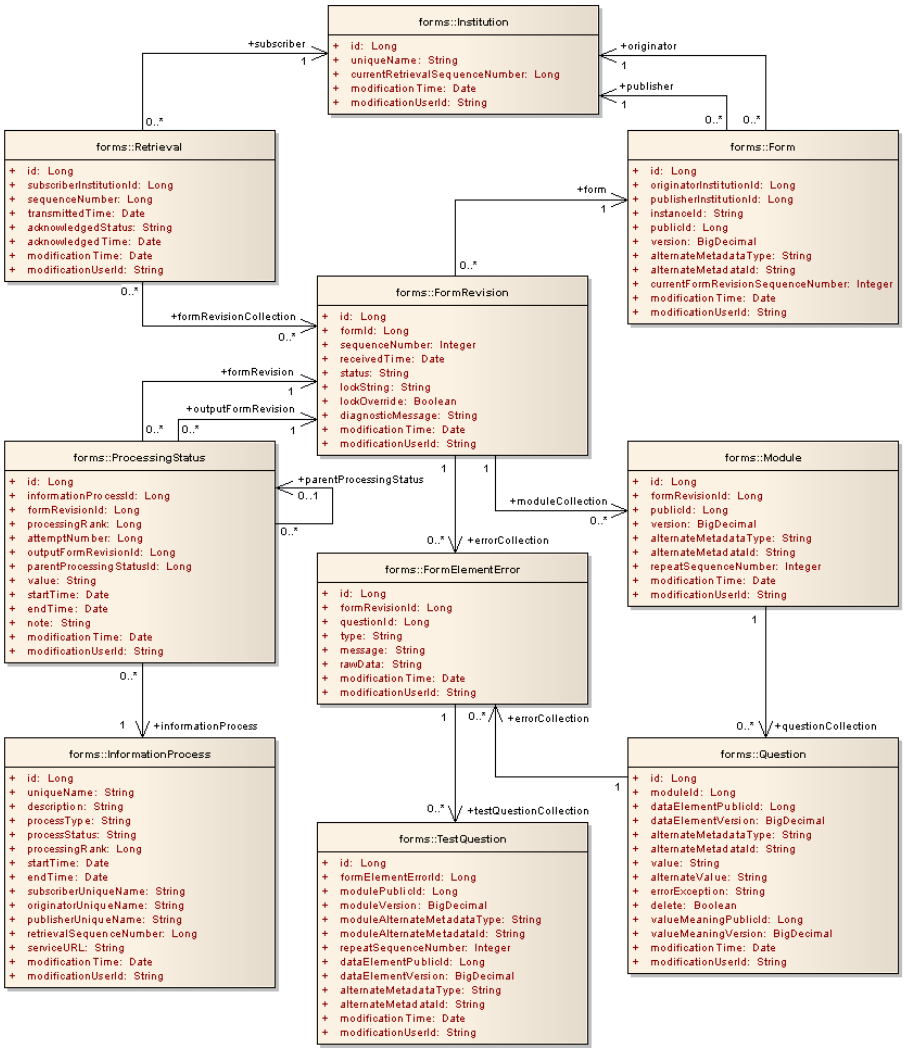
- http://ncicb.nci.nih.gov/NCICB/infrastructure/cacore_overview
- https://cabig.nci.nih.gov/workspaces/Architecture/caGrid

Note: The AGNIS data structures described in this document have not yet been curated in the caDSR metadata repository. Although forms data passing through AGNIS are comprised of CDEs, the AGNIS web service itself has not been certified as a caBIG compliant system.

Local Fields

Each database table described here contains a number of local fields, such as the table's automatically generated numeric identifier and modification time, which are meaningful only within the local database. These fields are normally omitted from inter-system AGNIS communications in XML format. Refer to the notes section on each table for details.

UML Class Diagram



Data Model

XML Schema

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"
 xmlns="gme://forms.AGNIS/2.0/net.agnis.forms"
 targetNamespace="gme://forms.AGNIS/2.0/net.agnis.forms"
 elementFormDefault="qualified"
 version="2.0">
  <xs:element name="Question" type="Question" />
  <xs:complexType name="Question">
    <xs:sequence>
      <xs:element name="errorCollection" type="FormElementError"
       minOccurs="0" maxOccurs="unbounded" />
      <xs:element name="value" type="xs:string"
       minOccurs="0" maxOccurs="1" />
      <xs:element name="alternateValue" type="xs:string"
       minOccurs="0" maxOccurs="1" />
    </xs:sequence>
    <xs:attribute name="id" type="xs:long" />
    <xs:attribute name="moduleId" type="xs:long" />
    <xs:attribute name="dataElementPublicId" type="xs:long" />
    <xs:attribute name="dataElementVersion" type="xs:decimal" />
    <xs:attribute name="alternateMetadataType" type="xs:string" />
    <xs:attribute name="alternateMetadataId" type="xs:string" />
    <xs:attribute name="errorException" type="xs:string" />
    <xs:attribute name="delete" type="xs:boolean" />
    <xs:attribute name="valueMeaningPublicId" type="xs:long" />
    <xs:attribute name="valueMeaningVersion" type="xs:decimal" />
    <xs:attribute name="modificationTime" type="xs:dateTime" />
    <xs:attribute name="modificationUserId" type="xs:string" />
  </xs:complexType>
  <xs:element name="Module" type="Module" />
  <xs:complexType name="Module">
    <xs:sequence>
      <xs:element name="questionCollection" type="Question"
       minOccurs="0" maxOccurs="unbounded" />
    </xs:sequence>
    <xs:attribute name="id" type="xs:long" />
    <xs:attribute name="formRevisionId" type="xs:long" />
    <xs:attribute name="publicId" type="xs:long" />
    <xs:attribute name="version" type="xs:decimal" />
    <xs:attribute name="alternateMetadataType" type="xs:string" />
    <xs:attribute name="alternateMetadataId" type="xs:string" />
    <xs:attribute name="repeatSequenceNumber" type="xs:integer" />
    <xs:attribute name="modificationTime" type="xs:dateTime" />
    <xs:attribute name="modificationUserId" type="xs:string" />
  </xs:complexType>
  <xs:element name="Institution" type="Institution" />
  <xs:complexType name="Institution">
    <xs:sequence />
    <xs:attribute name="id" type="xs:long" />
    <xs:attribute name="uniqueName" type="xs:string" />
    <xs:attribute name="currentRetrievalSequenceNumber" type="xs:long" />
    <xs:attribute name="modificationTime" type="xs:dateTime" />
    <xs:attribute name="modificationUserId" type="xs:string" />
  </xs:complexType>
  <xs:element name="FormRevision" type="FormRevision" />
  <xs:complexType name="FormRevision">
```



```
  <xs:sequence>
    <xs:element name="form" type="Form"
     minOccurs="0" maxOccurs="1" />
    <xs:element name="errorCollection" type="FormElementError"
     minOccurs="0" maxOccurs="unbounded" />
    <xs:element name="moduleCollection" type="Module"
     minOccurs="0" maxOccurs="unbounded" />
  </xs:sequence>
  <xs:attribute name="id" type="xs:long" />
  <xs:attribute name="formId" type="xs:long" />
  <xs:attribute name="sequenceNumber" type="xs:integer" />
  <xs:attribute name="receivedTime" type="xs:dateTime" />
  <xs:attribute name="status" type="xs:string" />
  <xs:attribute name="lockString" type="xs:string" />
  <xs:attribute name="lockOverride" type="xs:boolean" />
  <xs:attribute name="diagnosticMessage" type="xs:string" />
  <xs:attribute name="modificationTime" type="xs:dateTime" />
  <xs:attribute name="modificationUserId" type="xs:string" />
</xs:complexType>
<xs:element name="Retrieval" type="Retrieval" />
<xs:complexType name="Retrieval">
  <xs:sequence>
    <xs:element name="subscriber" type="Institution"
     minOccurs="0" maxOccurs="1" />
    <xs:element name="formRevisionCollection" type="FormRevision"
     minOccurs="0" maxOccurs="unbounded" />
  </xs:sequence>
  <xs:attribute name="id" type="xs:long" />
  <xs:attribute name="subscriberInstitutionId" type="xs:long" />
  <xs:attribute name="sequenceNumber" type="xs:long" />
  <xs:attribute name="transmittedTime" type="xs:dateTime" />
  <xs:attribute name="acknowledgedStatus" type="xs:string" />
  <xs:attribute name="acknowledgedTime" type="xs:dateTime" />
  <xs:attribute name="modificationTime" type="xs:dateTime" />
  <xs:attribute name="modificationUserId" type="xs:string" />
</xs:complexType>
<xs:element name="TestQuestion" type="TestQuestion" />
<xs:complexType name="TestQuestion">
  <xs:sequence />
  <xs:attribute name="id" type="xs:long" />
  <xs:attribute name="formElementErrorId" type="xs:long" />
  <xs:attribute name="modulePublicId" type="xs:long" />
  <xs:attribute name="moduleVersion" type="xs:decimal" />
  <xs:attribute name="moduleAlternateMetadataType" type="xs:string" />
  <xs:attribute name="moduleAlternateMetadataId" type="xs:string" />
  <xs:attribute name="repeatSequenceNumber" type="xs:integer" />
  <xs:attribute name="dataElementPublicId" type="xs:long" />
  <xs:attribute name="dataElementVersion" type="xs:decimal" />
  <xs:attribute name="alternateMetadataType" type="xs:string" />
  <xs:attribute name="alternateMetadataId" type="xs:string" />
  <xs:attribute name="modificationTime" type="xs:dateTime" />
  <xs:attribute name="modificationUserId" type="xs:string" />
</xs:complexType>
<xs:element name="InformationProcess" type="InformationProcess" />
<xs:complexType name="InformationProcess">
  <xs:sequence />
  <xs:attribute name="id" type="xs:long" />
  <xs:attribute name="uniqueName" type="xs:string" />
  <xs:attribute name="description" type="xs:string" />
```



```
  <xs:attribute name="processType" type="xs:string" />
  <xs:attribute name="processStatus" type="xs:string" />
  <xs:attribute name="processingRank" type="xs:long" />
  <xs:attribute name="startTime" type="xs:dateTime" />
  <xs:attribute name="endTime" type="xs:dateTime" />
  <xs:attribute name="subscriberUniqueName" type="xs:string" />
  <xs:attribute name="originatorUniqueName" type="xs:string" />
  <xs:attribute name="publisherUniqueName" type="xs:string" />
  <xs:attribute name="retrievalSequenceNumber" type="xs:long" />
  <xs:attribute name="serviceURL" type="xs:string" />
  <xs:attribute name="modificationTime" type="xs:dateTime" />
  <xs:attribute name="modificationUserId" type="xs:string" />
</xs:complexType>
<xs:element name="Form" type="Form" />
<xs:complexType name="Form">
  <xs:sequence>
    <xs:element name="publisher" type="Institution"
     minOccurs="0" maxOccurs="1" />
    <xs:element name="originator" type="Institution"
     minOccurs="0" maxOccurs="1" />
  </xs:sequence>
  <xs:attribute name="id" type="xs:long" />
  <xs:attribute name="originatorInstitutionId" type="xs:long" />
  <xs:attribute name="publisherInstitutionId" type="xs:long" />
  <xs:attribute name="instanceId" type="xs:string" />
  <xs:attribute name="publicId" type="xs:long" />
  <xs:attribute name="version" type="xs:decimal" />
  <xs:attribute name="alternateMetadataType" type="xs:string" />
  <xs:attribute name="alternateMetadataId" type="xs:string" />
  <xs:attribute name="currentFormRevisionSequenceNumber"
   type="xs:integer" />
  <xs:attribute name="modificationTime" type="xs:dateTime" />
  <xs:attribute name="modificationUserId" type="xs:string" />
</xs:complexType>
<xs:element name="ProcessingStatus" type="ProcessingStatus" />
<xs:complexType name="ProcessingStatus">
  <xs:sequence>
    <xs:element name="informationProcess" type="InformationProcess"
     minOccurs="0" maxOccurs="1" />
    <xs:element name="formRevision" type="FormRevision"
     minOccurs="0" maxOccurs="1" />
    <xs:element name="outputFormRevision" type="FormRevision"
     minOccurs="0" maxOccurs="1" />
    <xs:element name="parentProcessingStatus" type="ProcessingStatus"
     minOccurs="0" maxOccurs="1" />
  </xs:sequence>
  <xs:attribute name="id" type="xs:long" />
  <xs:attribute name="informationProcessId" type="xs:long" />
  <xs:attribute name="formRevisionId" type="xs:long" />
  <xs:attribute name="parentProcessingStatusId" type="xs:long" />
  <xs:attribute name="processingRank" type="xs:long" />
  <xs:attribute name="attemptNumber" type="xs:long" />
  <xs:attribute name="outputFormRevisionId" type="xs:long" />
  <xs:attribute name="value" type="xs:string" />
  <xs:attribute name="startTime" type="xs:dateTime" />
  <xs:attribute name="endTime" type="xs:dateTime" />
  <xs:attribute name="note" type="xs:string" />
  <xs:attribute name="modificationTime" type="xs:dateTime" />
  <xs:attribute name="modificationUserId" type="xs:string" />
```



```
  </xs:complexType>
  <xs:element name="FormElementError" type="FormElementError" />
  <xs:complexType name="FormElementError">
    <xs:sequence>
      <xs:element name="testQuestionCollection" type="TestQuestion"
       minOccurs="0" maxOccurs="unbounded" />
      <xs:element name="rawData" type="xs:string"
       minOccurs="0" maxOccurs="1" />
    </xs:sequence>
    <xs:attribute name="id" type="xs:long" />
    <xs:attribute name="formRevisionId" type="xs:long" />
    <xs:attribute name="questionId" type="xs:long" />
    <xs:attribute name="type" type="xs:string" />
    <xs:attribute name="message" type="xs:string" />
    <xs:attribute name="modificationTime" type="xs:dateTime" />
    <xs:attribute name="modificationUserId" type="xs:string" />
  </xs:complexType>
</xs:schema>
```


FORM Table

Description

Represents an instance of a form which has Case Report Form (CRF) metadata defined in the caDSR metadata repository.

Notes

- **Local Fields:** FORM_ID, PUBLISHER_INSTITUTION_ID, ORIGINATOR_INSTITUTION_ID, MODIFICATION_TIME, and MODIFICATION_USER are meaningful only within the local AGNIS (repository or staging) database. For inter-system AGNIS communications in XML format, these fields are normally omitted.

- **Semi-Local Field:** For some circumstances, CURRENT_FORM_REVISION_SEQ_NUM might also be best omitted from inter-system AGNIS communications.

Columns

Column	**Data Type**	**Description**
FORM_ID	64 bit integer	numeric identifier generated by database
PUBLISHER_INSTITUTION_ID	64 bit integer	form publisher identifier (foreign key)
INSTANCE_ID	varchar(255)	individual form identifier, e.g. NMDP
		sequence number
ORIGINATOR_INSTITUTION_ID	64 bit integer	form creator identifier (foreign key)
PUBLIC_ID	64 bit integer	partial caDSR identifier for CRF Form
		metadata
VERSION	numeric(4,2)	partial caDSR identifier for CRF Form
		metadata
ALTERNATE_METADATA_TYPE	varchar(255)	partial identifier for alternate (non-caDSR)
		metadata
ALTERNATE_METADATA_ID	varchar(255)	partial identifier for alternate (non-caDSR)
		metadata
CURRENT_FORM_REVISION_SEQ_NUM	32 bit integer	REVISION_SEQ_NUM of most recent record inserted into FORM_REVISION
		table for this form
MODIFICATION_TIME	timestamp	database time this record was modified
		(set by trigger?)
MODIFICATION_USER_ID	varchar(30)	database user who modified this record
		(set by trigger?)

Indexes

Index Type	Columns
Primary Key	FORM_ID
Non Unique	PUBLISHER_INSTITUTION_ID, INSTANCE_ID
Non Unique	PUBLISHER_INSTITUTION_ID
Non Unique	ORIGINATOR_INSTITUTION_ID

Foreign Keys

Column	Foreign Table	Foreign Column
PUBLISHER_INSTITUTION_ID	INSTITUTION	INSTITUTION_ID
ORIGINATOR_INSTITUTION_ID	INSTITUTION	INSTITUTION_ID

Object Relational Mapping

Type	Column	Class	Attribute/Association
attr	FORM_ID	Form	id
attr	PUBLISHER_INSTITUTION_ID	Form	publisherInstitutionId
attr	INSTANCE_ID	Form	instanceId
attr	ORIGINATOR_INSTITUTION_ID	Form	originatorInstitutionId
attr	PUBLIC_ID	Form	publicId
attr	VERSION	Form	version
attr	ALTERNATE_METADATA_TYPE	Form	alternateMetadataType
attr	ALTERNATE_METADATA_ID	Form	alternateMetadataId
attr	CURRENT_FORM_REVISION_SEQ_NUM	Form	currentFormRevisionSequenceNumber
attr	MODIFICATION_TIME	Form	modificationTime
attr	MODIFICATION_USER_ID	Form	modificationUser
assn	PUBLISHER_INSTITUTION_ID	Form	publisher
assn	ORIGINATOR_INSTITUTION_ID	Form	originator

(attr = mapped-attributes, assn = implements-association)

MySQL DDL

```
CREATE TABLE `FORM` (
  `FORM_ID` bigint(20) NOT NULL auto_increment,
  `PUBLISHER_INSTITUTION_ID` bigint(20) default NULL,
  `INSTANCE_ID` varchar(255) default NULL,
  `ORIGINATOR_INSTITUTION_ID` bigint(20) default NULL,
  `PUBLIC_ID` bigint(20) default NULL,
  `VERSION` decimal(4,2) default NULL,
  `ALTERNATE_METADATA_TYPE` varchar(255) default NULL,
  `ALTERNATE_METADATA_ID` varchar(255) default NULL,
  `CURRENT_FORM_REVISION_SEQ_NUM` int(11) default NULL,
  `MODIFICATION_TIME` datetime default NULL,
  `MODIFICATION_USER_ID` varchar(30) default NULL,
  PRIMARY KEY  (`FORM_ID`),
  KEY `PUBLISHER_INSTITUTION_ID` (`PUBLISHER_INSTITUTION_ID`,`INSTANCE_ID`),
  KEY `ORIGINATOR_INSTITUTION_ID` (`ORIGINATOR_INSTITUTION_ID`),
  KEY `PUBLISHER_INSTITUTION_ID_2` (`PUBLISHER_INSTITUTION_ID`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
```

Oracle DDL

```
CREATE SEQUENCE "FORM_ID_SEQ"

CREATE TABLE "FORM"
(
   "FORM_ID" NUMBER(18,0) NOT NULL ENABLE,
   "PUBLISHER_INSTITUTION_ID" NUMBER(18,0),
   "INSTANCE_ID" VARCHAR2(255),
   "ORIGINATOR_INSTITUTION_ID" NUMBER(18,0),
   "PUBLIC_ID" NUMBER,
   "VERSION" NUMBER(4,2),
   "CURRENT_FORM_REVISION_SEQ_NUM" NUMBER(8,0),
   "MODIFICATION_TIME" TIMESTAMP (6),
   "MODIFICATION_USER_ID" VARCHAR2(30),
   "ALTERNATE_METADATA_TYPE" VARCHAR2(255),
   "ALTERNATE_METADATA_ID" VARCHAR2(255),
    CONSTRAINT "FORM_PK" PRIMARY KEY ("FORM_ID") ENABLE
)

CREATE INDEX "FORM_IDX1"
 ON  "FORM" ("PUBLISHER_INSTITUTION_ID", "INSTANCE_ID")

CREATE INDEX "FORM_IDX2"
 ON "FORM" ("ORIGINATOR_INSTITUTION_ID")

CREATE INDEX "FORM_IDX3"
 ON "FORM" ("PUBLISHER_INSTITUTION_ID")

CREATE OR REPLACE TRIGGER "BI_FORM"
  before insert on "FORM"
  for each row
begin
    if :NEW.FORM_ID IS NULL then
        select "FORM_ID_SEQ".nextval into :NEW.FORM_ID from dual;
    end if;
```



```
end;

ALTER TRIGGER "BI_FORM" ENABLE
```

(Microsoft) SQL Server DDL

```
CREATE TABLE FORM
(
  FORM_ID bigint NOT NULL IDENTITY CONSTRAINT FORM_ID PRIMARY KEY,
  PUBLISHER_INSTITUTION_ID bigint default NULL,
  INSTANCE_ID varchar(255) default NULL,
  ORIGINATOR_INSTITUTION_ID bigint default NULL,
  PUBLIC_ID bigint default NULL,
  VERSION decimal(4,2) default NULL,
  ALTERNATE_METADATA_TYPE varchar(255) default NULL,
  ALTERNATE_METADATA_ID varchar(255) default NULL,
  CURRENT_FORM_REVISION_SEQ_NUM int default NULL,
  MODIFICATION_TIME datetime default NULL,
  MODIFICATION_USER_ID varchar(30) default NULL
)

CREATE INDEX PUBLISHER_INSTITUTION_ID
ON FORM (PUBLISHER_INSTITUTION_ID,INSTANCE_ID)

CREATE INDEX ORIGINATOR_INSTITUTION_ID
ON FORM (ORIGINATOR_INSTITUTION_ID)

CREATE INDEX PUBLISHER_INSTITUTION_ID_2
ON FORM (PUBLISHER_INSTITUTION_ID)
```

Sybase DDL

```
CREATE TABLE FORM
(
  FORM_ID bigint IDENTITY CONSTRAINT FORM_ID PRIMARY KEY,
  PUBLISHER_INSTITUTION_ID bigint NULL,
  INSTANCE_ID varchar(255) NULL,
  ORIGINATOR_INSTITUTION_ID bigint NULL,
  PUBLIC_ID bigint NULL,
  VERSION decimal(4,2) NULL,
  ALTERNATE_METADATA_TYPE varchar(255) NULL,
  ALTERNATE_METADATA_ID varchar(255) NULL,
  CURRENT_FORM_REVISION_SEQ_NUM int NULL,
  MODIFICATION_TIME datetime NULL,
  MODIFICATION_USER_ID varchar(30) NULL
)

CREATE INDEX PUBLISHER_INSTITUTION_ID
ON FORM (PUBLISHER_INSTITUTION_ID,INSTANCE_ID)

CREATE INDEX ORIGINATOR_INSTITUTION_ID
ON FORM (ORIGINATOR_INSTITUTION_ID)

CREATE INDEX PUBLISHER_INSTITUTION_ID_2
ON FORM (PUBLISHER_INSTITUTION_ID)
```


FORM_ELEMENT_ERROR Table

Description

Represents an error associated either with a form revision, or with a specific question within a form revision.

Notes

- **Local Fields:** FORM_ELEMENT_ERROR_ID, FORM_REVISION_ID, QUESTION_ID, MODIFICATION_TIME, and MODIFICATION_USER_ID are meaningful only within the local AGNIS (repository or staging) database. For inter-system AGNIS communications in XML format, these fields are normally omitted.

Columns

Column	Data Type	Description
FORM_ELEMENT_ERROR_ID	64 bit integer	numeric identifier generated by database
FORM_REVISION_ID	64 bit integer	for form level error, identifies associated form revision (foreign key); otherwise NULL
QUESTION_ID	64 bit integer	for question level error, identifies associated question (foreign key); otherwise NULL
ERROR_TYPE	varchar(255)	code specifying error type
ERROR_MESSAGE	varchar(255)	text error message
ERROR_RAW_DATA	text	extended error message
MODIFICATION_TIME	timestamp	database time this record was modified (set by trigger?)
MODIFICATION_USER_ID	varchar(30)	database user who modified this record (set by trigger?)

Indexes

Index Type	Columns
Primary Key	FORM_ELEMENT_ERROR_ID
Non Unique	FORM_REVISION_ID
Non Unique	QUESTION_ID

Foreign Keys

Column	Foreign Table	Foreign Column
FORM_REVISION_ID	FORM_REVISION	FORM_REVISION_ID
QUESTION_ID	QUESTION	QUESTION_ID

Object Relational Mapping

Mapping Type	Column	Class	Attribute
attr	FORM_ELEMENT_ERROR_ID	FormElementError	id
attr	FORM_REVISION_ID	FormElementError	formRevisionId
attr	QUESTION_ID	FormElementError	questionId
attr	ERROR_TYPE	FormElementError	type
attr	ERROR_MESSAGE	FormElementError	message
attr	ERROR_RAW_DATA	FormElementError	rawData
attr	MODIFICATION_TIME	FormElementError	modificationTime
attr	MODIFICATION_USER_ID	FormElementError	modificationUserId
assn	FORM_REVISION_ID	FormElementError	formRevision
assn	QUESTION_ID	FormElementError	question

(attr = mapped-attributes, assn = implements-association)

MySQL DDL

```
CREATE TABLE `FORM_ELEMENT_ERROR` (
  `FORM_ELEMENT_ERROR_ID` bigint(20) NOT NULL auto_increment,
  `FORM_REVISION_ID` bigint(20) default NULL,
  `QUESTION_ID` bigint(20) default NULL,
  `ERROR_TYPE` varchar(255) default NULL,
  `ERROR_MESSAGE` varchar(255) default NULL,
  `ERROR_RAW_DATA` text,
  `MODIFICATION_TIME` datetime default NULL,
  `MODIFICATION_USER_ID` varchar(30) default NULL,
  PRIMARY KEY  (`FORM_ELEMENT_ERROR_ID`),
  KEY `FORM_REVISION_ID` (`FORM_REVISION_ID`),
  KEY `QUESTION_ID` (`QUESTION_ID`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
```


Oracle DDL

```
CREATE SEQUENCE "FORM_ELEMENT_ERROR_ID_SEQ"

CREATE TABLE "FORM_ELEMENT_ERROR"
(
   "FORM_ELEMENT_ERROR_ID" NUMBER(18,0) NOT NULL ENABLE,
   "FORM_REVISION_ID" NUMBER(18,0),
   "QUESTION_ID" NUMBER(18,0),
   "ERROR_TYPE" VARCHAR2(255),
   "ERROR_MESSAGE" VARCHAR2(255),
   "ERROR_RAW_DATA" VARCHAR2(4000),
   "MODIFICATION_TIME" TIMESTAMP (6),
   "MODIFICATION_USER_ID" VARCHAR2(30),
    CONSTRAINT "FORM_ELEMENT_ERROR_PK" PRIMARY KEY ("FORM_ELEMENT_ERROR_ID") ENABLE
)

CREATE INDEX "FORM_ELEMENT_ERROR_IDX1"
 ON "FORM_ELEMENT_ERROR" ("FORM_REVISION_ID")

CREATE INDEX "FORM_ELEMENT_ERROR_IDX2"
 ON "FORM_ELEMENT_ERROR" ("QUESTION_ID")

CREATE OR REPLACE TRIGGER "BI_FORM_ELEMENT_ERROR"
  before insert on "FORM_ELEMENT_ERROR"
  for each row
begin
    if :NEW.FORM_ELEMENT_ERROR_ID IS NULL then
        select "FORM_ELEMENT_ERROR_ID_SEQ".nextval into :NEW.FORM_ELEMENT_ERROR_ID from dual;
    end if;
end;

ALTER TRIGGER "BI_FORM_ELEMENT_ERROR" ENABLE
```

(Microsoft) SQL Server DDL

```
CREATE TABLE FORM_ELEMENT_ERROR
(
  FORM_ELEMENT_ERROR_ID bigint NOT NULL IDENTITY CONSTRAINT FORM_ELEMENT_ERROR_ID PRIMARY KEY,
  FORM_REVISION_ID bigint default NULL,
  QUESTION_ID bigint default NULL,
  ERROR_TYPE varchar(255) default NULL,
  ERROR_MESSAGE varchar(255) default NULL,
  ERROR_RAW_DATA text,
  MODIFICATION_TIME datetime default NULL,
  MODIFICATION_USER_ID varchar(30) default NULL
)

CREATE INDEX FORM_REVISION_ID
ON FORM_ELEMENT_ERROR (FORM_REVISION_ID)

CREATE INDEX QUESTION_ID
ON FORM_ELEMENT_ERROR (QUESTION_ID)
```


Sybase DDL

```
CREATE TABLE FORM_ELEMENT_ERROR
(
  FORM_ELEMENT_ERROR_ID bigint IDENTITY CONSTRAINT FORM_ELEMENT_ERROR_ID PRIMARY KEY,
  FORM_REVISION_ID bigint NULL,
  QUESTION_ID bigint NULL,
  ERROR_TYPE varchar(255) NULL,
  ERROR_MESSAGE varchar(255) NULL,
  ERROR_RAW_DATA text NULL,
  MODIFICATION_TIME datetime NULL,
  MODIFICATION_USER_ID varchar(30) NULL
)

CREATE INDEX FORM_REVISION_ID
ON FORM_ELEMENT_ERROR (FORM_REVISION_ID)

CREATE INDEX QUESTION_ID
ON FORM_ELEMENT_ERROR (QUESTION_ID)
```


FORM_REVISION Table

Description

Represents a specific revision of a form instance.

Notes

- **Local Fields:** FORM_REVISION_ID, FORM_ID, MODIFICATION_TIME, and MODIFICATION_USER_ID are meaningful only within the local AGNIS (repository or staging) database. For inter-system AGNIS communications in XML format, these fields are normally omitted.

- **Semi-Local Field:** For some circumstances, FORM_REVISION_SEQ_NUM might also be best omitted from inter-system AGNIS communications.

- **Volatile Field:** The FormRevision diagnosticMessage attribute is included in inter-system AGNIS communications in XML format, but is not mapped to a database column here, and is therefore not normally persisted to the database.

- **Status Field:** STATUS is meaningful as an indication of whether a submitted form revision was accepted by FormsNet, but may not be useful at the AGNIS repository level.

- **Concurrency Control:** LOCK_STRING and LOCK_OVERRIDE are used as part of the AGNIS optimistic locking strategy, e.g. when submitting forms data stored in a local staging database.

Columns

Column	Data Type	Description
FORM_REVISION_ID	64 bit integer	numeric identifier generated by database
FORM_ID	64 bit integer	form identifier (foreign key)
FORM_REVISION_SEQ_NUM	32 bit integer	partial identifier (with FORM_ID) of this form revision
RECEIVED_TIME	timestamp	time when this form revision was received
STATUS	varchar(30)	status (used externally but not actually needed here?)
LOCK_STRING	varchar(255)	"version" indicator, used by optimistic locking mechanism to detect collisions during data submission
LOCK_OVERRIDE	8 bit integer	pseudo-boolean "lock override" flag, e.g. to disable usage of optimistic locking in originator mode
MODIFICATION_TIME	timestamp	database time this record was modified (set by trigger?)
MODIFICATION_USER_ID	varchar(30)	database user who modified this record (set by trigger?)

Indexes

Index Type	Columns
Primary Key	FORM_REVISION_ID
Unique	FORM_ID, FORM_REVISION_SEQ_NUM
Non Unique	FORM_ID

Foreign Keys

Column	Foreign Table	Foreign Column
FORM_ID	FORM	FORM_ID

Object Relational Mapping

Type	Column	Class	Attribute/Association
attr	FORM_REVISION_ID	FormRevision	id
attr	FORM_ID	FormRevision	formId
attr	FORM_REVISION_SEQ_NUM	FormRevision	sequenceNumber
attr	RECEIVED_TIME	FormRevision	receivedTime
attr	STATUS	FormRevision	status
attr	LOCK_STRING	FormRevision	lockString
attr	LOCK_OVERRIDE	FormRevision	lockOverride
attr	MODIFICATION_TIME	FormRevision	modificationTime
attr	MODIFICATION_USER_ID	FormRevision	modificationUserId

assn	FORM_ID	FormRevision	form

(attr = mapped-attributes, assn = implements-association)

MySQL DDL

```
CREATE TABLE `FORM_REVISION` (
  `FORM_REVISION_ID` bigint(20) NOT NULL auto_increment,
  `FORM_ID` bigint(20) NOT NULL,
  `FORM_REVISION_SEQ_NUM` int(11) NOT NULL,
  `RECEIVED_TIME` datetime default NULL,
  `STATUS` varchar(30) default NULL,
  `LOCK_STRING` varchar(255) default NULL,
  `LOCK_OVERRIDE_FLAG` tinyint(1) default NULL,
  `MODIFICATION_TIME` datetime default NULL,
  `MODIFICATION_USER_ID` varchar(30) default NULL,
  PRIMARY KEY  (`FORM_REVISION_ID`),
  UNIQUE KEY `FORM_ID` (`FORM_ID`,`FORM_REVISION_SEQ_NUM`),
  KEY `FORM_ID_2` (`FORM_ID`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
```

Oracle DDL

```
CREATE SEQUENCE "FORM_REVISION_ID_SEQ"

CREATE TABLE "FORM_REVISION"
(
   "FORM_REVISION_ID" NUMBER(18,0) NOT NULL ENABLE,
   "FORM_ID" NUMBER(18,0) NOT NULL ENABLE,
   "FORM_REVISION_SEQ_NUM" NUMBER(8,0) NOT NULL ENABLE,
   "RECEIVED_TIME" TIMESTAMP (6),
   "STATUS" VARCHAR2(30),
   "LOCK_STRING" VARCHAR2(255),
   "LOCK_OVERRIDE_FLAG" NUMBER(1,0),
   "MODIFICATION_TIME" TIMESTAMP (6),
   "MODIFICATION_USER_ID" VARCHAR2(30),
    CONSTRAINT "FORM_REVISION_PK" PRIMARY KEY ("FORM_REVISION_ID") ENABLE
)

CREATE UNIQUE INDEX "FORM_REVISION_IDX1"
 ON "FORM_REVISION" ("FORM_ID","FORM_REVISION_SEQ_NUM")

CREATE INDEX "FORM_REVISION_IDX2"
 ON "FORM_REVISION" ("FORM_ID")

CREATE OR REPLACE TRIGGER "BI_FORM_REVISION"
  before insert on "FORM_REVISION"
  for each row
begin
    if :NEW.FORM_REVISION_ID IS NULL then
        select "FORM_REVISION_ID_SEQ".nextval into :NEW.FORM_REVISION_ID from dual;
    end if;
end;

ALTER TRIGGER "BI_FORM_REVISION" ENABLE
```


(Microsoft) SQL Server DDL

```
CREATE TABLE FORM_REVISION (
  FORM_REVISION_ID bigint NOT NULL IDENTITY CONSTRAINT FORM_REVISION_ID PRIMARY KEY,
  FORM_ID bigint NOT NULL,
  FORM_REVISION_SEQ_NUM int NOT NULL,
  RECEIVED_TIME datetime default NULL,
  STATUS varchar(30) default NULL,
  LOCK_STRING varchar(255) default NULL,
  LOCK_OVERRIDE_FLAG tinyint default NULL,
  MODIFICATION_TIME datetime default NULL,
  MODIFICATION_USER_ID varchar(30) default NULL
)

CREATE UNIQUE INDEX FORM_ID
ON FORM_REVISION (FORM_ID,FORM_REVISION_SEQ_NUM)

CREATE INDEX FORM_ID_2
ON FORM_REVISION (FORM_ID)
```

Sybase DDL

```
CREATE TABLE FORM_REVISION (
  FORM_REVISION_ID bigint IDENTITY CONSTRAINT FORM_REVISION_ID PRIMARY KEY,
  FORM_ID bigint NOT NULL,
  FORM_REVISION_SEQ_NUM int NOT NULL,
  RECEIVED_TIME datetime NULL,
  STATUS varchar(30) NULL,
  LOCK_STRING varchar(255) NULL,
  LOCK_OVERRIDE_FLAG tinyint NULL,
  MODIFICATION_TIME datetime NULL,
  MODIFICATION_USER_ID varchar(30) NULL
)

CREATE UNIQUE INDEX FORM_ID
ON FORM_REVISION (FORM_ID,FORM_REVISION_SEQ_NUM)

CREATE INDEX FORM_ID_2
ON FORM_REVISION (FORM_ID)
```


INFORMATION_PROCESS Table

Description

Represents a process, such as forms submission or extract, having to do with AGNIS data.

Notes

- **Local Fields:** INFORMATION_PROCESS_ID, MODIFICATION_TIME, and MODIFICATION_USER_ID are meaningful only within the local AGNIS (repository or staging) database. For inter-system AGNIS communications in XML format, these fields are normally omitted.

Columns

Column	Data Type	Description
INFORMATION_PROCESS_ID	64 bit integer	numeric identifier generated by database
UNIQUE_NAME	varchar(255)	unique name assigned to the process
DESCRIPTION	varchar(255)	brief text description of the process
PROCESS_TYPE	varchar(30)	process type indicator, e.g. RETRIEVE, SUBMIT, PUBLISH
PROCESS_STATUS	varchar(30)	process status indicator, e.g. READY, COMPLETED, etc.
PROCESSING_RANK	64 bit integer	numeric rank of process; e.g. staging client processes these records in ascending order by rank
START_TIME	timestamp	processing start time
END_TIME	timestamp	processing end time
SUBSCRIBER_UNIQUE_NAME	varchar(255)	name of subscriber, e.g. for RETRIEVE process
ORIGINATOR_UNIQUE_NAME	varchar(255)	name of originator, e.g. for SUBMIT process
PUBLISHER_UNIQUE_NAME	varchar(255)	name of publisher, e.g. for PUBLISH process
RETRIEVAL_SEQ_NUM	64 bit integer	partial identifier of RETRIEVAL record from a RETRIEVE process
SERVICE_URL	varchar(255)	URL of AGNIS web service used for this process
MODIFICATION_TIME	timestamp	database time this record was modified (set by trigger?)
MODIFICATION_USER_ID	varchar(30)	database user who modified this record (set by trigger?)

Indexes

Index Type	Columns
Primary Key	INFORMATION_PROCESS_ID
Unique	UNIQUE_NAME

Foreign Keys

None.

Object Relational Mapping

Type	Column	Class	Attribute/Association
attr	INFORMATION_PROCESS_ID	InformationProcess	id
attr	UNIQUE_NAME	InformationProcess	uniqueName
attr	DESCRIPTION	InformationProcess	description
attr	PROCESS_TYPE	InformationProcess	processType
attr	PROCESS_STATUS	InformationProcess	processStatus
attr	PROCESSING_RANK	InformationProcess	processingRank
attr	START_TIME	InformationProcess	startTime
attr	END_TIME	InformationProcess	endTime
attr	SUBSCRIBER_UNIQUE_NAME	InformationProcess	subscriberUniqueName
attr	ORIGINATOR_UNIQUE_NAME	InformationProcess	originatorUniqueName
attr	PUBLISHER_UNIQUE_NAME	InformationProcess	publisherUniqueName
attr	RETRIEVAL_SEQ_NUM	InformationProcess	retrievalSequenceNumber
attr	SERVICE_URL	InformationProcess	serviceURL
attr	MODIFICATION_TIME	InformationProcess	modificationTime
attr	MODIFICATION_USER_ID	InformationProcess	modificationUserId

(attr = mapped-attributes, assn = implements-association)

MySQL DDL

```
CREATE TABLE `INFORMATION_PROCESS` (
  `INFORMATION_PROCESS_ID` bigint(20) NOT NULL auto_increment,
  `UNIQUE_NAME` varchar(255) NOT NULL,
  `DESCRIPTION` varchar(255) default NULL,
  `PROCESS_TYPE` varchar(30) default NULL,
  `PROCESS_STATUS` varchar(30) default NULL,
  `PROCESSING_RANK` bigint(20) default NULL,
  `START_TIME` datetime default NULL,
  `END_TIME` datetime default NULL,
  `SUBSCRIBER_UNIQUE_NAME` varchar(255) default NULL,
```



```
  `ORIGINATOR_UNIQUE_NAME` varchar(255) default NULL,
  `PUBLISHER_UNIQUE_NAME` varchar(255) default NULL,
  `RETRIEVAL_SEQ_NUM` bigint(20) default NULL,
  `SERVICE_URL` varchar(255) default NULL,
  `MODIFICATION_TIME` datetime default NULL,
  `MODIFICATION_USER_ID` varchar(30) default NULL,
  PRIMARY KEY  (`INFORMATION_PROCESS_ID`),
  UNIQUE KEY `UNIQUE_NAME` (`UNIQUE_NAME`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
```

Oracle DDL

```
CREATE SEQUENCE "INFORMATION_PROCESS_ID_SEQ"

CREATE TABLE "INFORMATION_PROCESS"
(
   "INFORMATION_PROCESS_ID" NUMBER(18,0),
   "UNIQUE_NAME" VARCHAR2(255),
   "DESCRIPTION" VARCHAR2(255),
   "PROCESS_TYPE" VARCHAR2(30),
   "PROCESS_STATUS" VARCHAR2(30),
   "PROCESSING_RANK" NUMBER(18,0),
   "START_TIME" TIMESTAMP (6),
   "END_TIME" TIMESTAMP (6),
   "SUBSCRIBER_UNIQUE_NAME" VARCHAR2(255),
   "ORIGINATOR_UNIQUE_NAME" VARCHAR2(255),
   "PUBLISHER_UNIQUE_NAME" VARCHAR2(255),
   "RETRIEVAL_SEQ_NUM" NUMBER(18,0),
   "SERVICE_URL" VARCHAR2(255),
   "MODIFICATION_TIME" TIMESTAMP (6),
   "MODIFICATION_USER_ID" VARCHAR2(30),
    CONSTRAINT "INFORMATION_PROCESS_PK" PRIMARY KEY ("INFORMATION_PROCESS_ID") ENABLE
)

CREATE UNIQUE INDEX "INFORMATION_PROCESS_IDX1"
 ON "INFORMATION_PROCESS" ("UNIQUE_NAME")

CREATE OR REPLACE TRIGGER "BI_INFORMATION_PROCESS"
  before insert on "INFORMATION_PROCESS"
  for each row
begin
    if :NEW.INFORMATION_PROCESS_ID IS NULL then
        select "INFORMATION_PROCESS_ID_SEQ".nextval into :NEW.INFORMATION_PROCESS_ID from dual;
    end if;
end;

ALTER TRIGGER "BI_INFORMATION_PROCESS" ENABLE
```


(Microsoft) SQL Server DDL

```
CREATE TABLE INFORMATION_PROCESS (
  INFORMATION_PROCESS_ID bigint NOT NULL IDENTITY CONSTRAINT INFORMATION_PROCESS_ID PRIMARY KEY,
  UNIQUE_NAME varchar(255) NOT NULL,
  DESCRIPTION varchar(255) default NULL,
  PROCESS_TYPE varchar(30) default NULL,
  PROCESS_STATUS varchar(30) default NULL,
  PROCESSING_RANK bigint default NULL,
  START_TIME datetime default NULL,
  END_TIME datetime default NULL,
  SUBSCRIBER_UNIQUE_NAME varchar(255) default NULL,
  ORIGINATOR_UNIQUE_NAME varchar(255) default NULL,
  PUBLISHER_UNIQUE_NAME varchar(255) default NULL,
  RETRIEVAL_SEQ_NUM bigint default NULL,
  SERVICE_URL varchar(255) default NULL,
  MODIFICATION_TIME datetime default NULL,
  MODIFICATION_USER_ID varchar(30) default NULL
)

CREATE UNIQUE INDEX UNIQUE_NAME
ON INFORMATION_PROCESS (UNIQUE_NAME)
```

Sybase DDL

```
CREATE TABLE INFORMATION_PROCESS (
  INFORMATION_PROCESS_ID bigint IDENTITY CONSTRAINT INFORMATION_PROCESS_ID PRIMARY KEY,
  UNIQUE_NAME varchar(255) NOT NULL,
  DESCRIPTION varchar(255) NULL,
  PROCESS_TYPE varchar(30) NULL,
  PROCESS_STATUS varchar(30) NULL,
  PROCESSING_RANK bigint NULL,
  START_TIME datetime NULL,
  END_TIME datetime NULL,
  SUBSCRIBER_UNIQUE_NAME varchar(255) NULL,
  ORIGINATOR_UNIQUE_NAME varchar(255) NULL,
  PUBLISHER_UNIQUE_NAME varchar(255) NULL,
  RETRIEVAL_SEQ_NUM bigint NULL,
  SERVICE_URL varchar(255) NULL,
  MODIFICATION_TIME datetime NULL,
  MODIFICATION_USER_ID varchar(30) NULL
)

CREATE UNIQUE INDEX UNIQUE_NAME
ON INFORMATION_PROCESS (UNIQUE_NAME)
```


INSTITUTION Table

Description

Represents an institution. The institution could be a subscriber authorized to retrieve forms data from AGNIS, a publisher (e.g. the NMDP) of forms data within AGNIS, or the originator of a specific form.

Notes

- **Local Fields:** INSTITUTION_ID, MODIFICATION_TIME, and MODIFICATION_USER_ID are meaningful only within the local AGNIS (repository or staging) database. For inter-system AGNIS communications in XML format, these fields are normally omitted.

- **Semi-Local Field:** For some circumstances, CURRENT_RETRIEVAL_SEQ_NUM might also be best omitted from inter-system AGNIS communications.

Columns

Column	Data Type	Description
INSTITUTION_ID	64 bit integer	numeric identifier generated by database
UNIQUE_NAME	varchar(255)	unique name assigned to this institution
CURRENT_RETRIEVAL_SEQ_NUM	64 bit integer	RETRIEVAL_SEQ_NUM of most recent record inserted into RETRIEVAL table for this institution (subscriber)
MODIFICATION_TIME	timestamp	database time this record was modified (set by trigger?)
MODIFICATION_USER_ID	varchar(30)	database user who modified this record (set by trigger?)

Indexes

Index Type	Columns
Primary Key	INSTITUTION_ID
Unique	UNIQUE_NAME

Foreign Keys

None.

Object Relational Mapping

Type	Column	Class	Attribute
attr	INSTITUTION_ID	Institution	id
attr	UNIQUE_NAME	Institution	uniqueName
attr	CURRENT_RETRIEVAL_SEQ_NUM	Institution	currentRetrievalSequenceNumber
attr	MODIFICATION_TIME	Institution	modificationTime
attr	MODIFICATION_USER_ID	Institution	modificationUserId

(attr = mapped-attributes, assn = implements-association)

MySQL DDL

```
CREATE TABLE `INSTITUTION` (
  `INSTITUTION_ID` bigint(20) NOT NULL auto_increment,
  `UNIQUE_NAME` varchar(255) NOT NULL,
  `CURRENT_RETRIEVAL_SEQ_NUM` bigint(20) default NULL,
  `MODIFICATION_TIME` datetime default NULL,
  `MODIFICATION_USER_ID` varchar(30) default NULL,
  PRIMARY KEY  (`INSTITUTION_ID`),
  UNIQUE KEY `UNIQUE_NAME` (`UNIQUE_NAME`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
```

Oracle DDL

```
CREATE SEQUENCE "INSTITUTION_ID_SEQ"

CREATE TABLE "INSTITUTION"
(
   "INSTITUTION_ID" NUMBER(18,0),
   "UNIQUE_NAME" VARCHAR2(255),
   "CURRENT_RETRIEVAL_SEQ_NUM" NUMBER(18,0),
   "MODIFICATION_TIME" TIMESTAMP (6),
   "MODIFICATION_USER_ID" VARCHAR2(30),
    CONSTRAINT "INSTITUTION_PK" PRIMARY KEY ("INSTITUTION_ID") ENABLE
)

CREATE UNIQUE INDEX "INSTITUTION_IDX1"
 ON "INSTITUTION" ("UNIQUE_NAME")

CREATE OR REPLACE TRIGGER "BI_INSTITUTION"
  before insert on "INSTITUTION"
  for each row
begin
    if :NEW.INSTITUTION_ID IS NULL then
        select "INSTITUTION_ID_SEQ".nextval into :NEW.INSTITUTION_ID from dual;
    end if;
end;

ALTER TRIGGER  "BI_INSTITUTION" ENABLE
```


(Microsoft) SQL Server DDL

```
CREATE TABLE INSTITUTION (
  INSTITUTION_ID bigint NOT NULL IDENTITY CONSTRAINT INSTITUTION_ID PRIMARY KEY,
  UNIQUE_NAME varchar(255) NOT NULL,
  CURRENT_RETRIEVAL_SEQ_NUM bigint default NULL,
  MODIFICATION_TIME datetime default NULL,
  MODIFICATION_USER_ID varchar(30) default NULL
)

CREATE UNIQUE INDEX UNIQUE_NAME
ON INSTITUTION (UNIQUE_NAME)
```

Sybase DDL

```
CREATE TABLE INSTITUTION (
  INSTITUTION_ID bigint IDENTITY CONSTRAINT INSTITUTION_ID PRIMARY KEY,
  UNIQUE_NAME varchar(255) NOT NULL,
  CURRENT_RETRIEVAL_SEQ_NUM bigint NULL,
  MODIFICATION_TIME datetime NULL,
  MODIFICATION_USER_ID varchar(30) NULL
)

CREATE UNIQUE INDEX UNIQUE_NAME
ON INSTITUTION (UNIQUE_NAME)
```


MODULE Table

Description

Represents an instance of module belonging to a form which has Case Report Form (CRF) metadata defined in the caDSR metadata repository.

Notes

- **Local Fields:** MODULE_ID, FORM_REVISION_ID, MODIFICATION_TIME, and MODIFICATION_USER_ID are meaningful only within the local AGNIS (repository or staging) database. For inter-system AGNIS communications in XML format, these fields are normally omitted.

- **Alternate Identifiers:** A module instance is identified by either (FORM_REVISION_ID, PUBLIC_ID, VERSION, REPEAT_SEQUENCE_NUMBER) or (FORM_REVISION_ID, ALTERNATE_METADATA_TYPE, ALTERNATE_METADATA_ID, REPEAT_SEQUENCE_NUMBER), depending on whether caDSR or alternate (non-caDSR) metadata has been specified.

Columns

Column	Data Type	Description
MODULE_ID	64 bit integer	numeric identifier generated by database
FORM_REVISION_ID	64 bit integer	partial identifier for this module instance; identifies form revision associated with this question (foreign key)
PUBLIC_ID	64 bit integer	partial identifier for this module instance; also partial identifier for the caDSR Module associated with this module
VERSION	numeric(4,2)	partial identifier for this module instance; also partial identifier for the caDSR Module associated with this module
ALTERNATE_METADATA_TYPE	varchar(255)	partial identifier for this module instance; also partial identifier for alternate (non-caDSR) metadata associated with this question
ALTERNATE_METADATA_ID	varchar(255)	partial identifier for this module instance; also partial identifier for alternate (non-caDSR) metadata associated with this question
REPEAT_SEQUENCE_NUMBER	32 bit integer	partial identifier for this module instance
MODIFICATION_TIME	timestamp	database time this record was modified (set by trigger?)
MODIFICATION_USER_ID	varchar(30)	database user who modified this record (set by trigger?)

Indexes

Index Type	Columns
Primary Key	MODULE_ID
Non Unique	FORM_REVISION_ID

Foreign Keys

Column	Foreign Table	Foreign Column
FORM_REVISION_ID	FORM_REVISION	FORM_REVISION_ID

Object Relational Mapping

Type	Column	Class	Attribute/Association
attr	MODULE_ID	Module	id
attr	FORM_REVISION_ID	Module	formRevisionId
attr	PUBLIC_ID	Module	publicId
attr	VERSION	Module	version
attr	ALTERNATE_METADATA_TYPE	Module	alternateMetadataType
attr	ALTERNATE_METADATA_ID	Module	alternateMetadataId
attr	REPEAT_SEQUENCE_NUMBER	Module	repeatSequenceNumber
attr	MODIFICATION_TIME	Module	modificationTime
attr	MODIFICATION_USER_ID	Module	modificationUserId
assn	FORM_REVISION_ID	Module	formRevision

(attr = mapped-attributes, assn = implements-association)

MySQL DDL

```
DROP TABLE IF EXISTS `MODULE`;
CREATE TABLE `MODULE` (
  `MODULE_ID` bigint(20) NOT NULL auto_increment,
  `FORM_REVISION_ID` bigint(20) NOT NULL,
  `PUBLIC_ID` bigint(20) default NULL,
  `VERSION` decimal(4,2) default NULL,
  `ALTERNATE_METADATA_TYPE` varchar(255) default NULL,
  `ALTERNATE_METADATA_ID` varchar(255) default NULL,
  `REPEAT_SEQUENCE_NUMBER` int(11) NOT NULL,
  `MODIFICATION_TIME` datetime default NULL,
  `MODIFICATION_USER_ID` varchar(30) default NULL,
  PRIMARY KEY  (`MODULE_ID`),
  KEY `FORM_REVISION_ID` (`FORM_REVISION_ID`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
```

Oracle DDL

```
CREATE SEQUENCE "MODULE_ID_SEQ"

CREATE TABLE "MODULE"
(
   "MODULE_ID" NUMBER(18,0),
   "FORM_REVISION_ID" NUMBER(18,0),
   "PUBLIC_ID" NUMBER(19,0),
   "VERSION" NUMBER(4,2),
   "ALTERNATE_METADATA_TYPE" VARCHAR2(255),
   "ALTERNATE_METADATA_ID" VARCHAR2(255),
   "REPEAT_SEQUENCE_NUMBER" NUMBER(8,0),
   "MODIFICATION_TIME" TIMESTAMP (6),
   "MODIFICATION_USER_ID" VARCHAR2(30),
    CONSTRAINT "MODULE_PK" PRIMARY KEY ("MODULE_ID") ENABLE
```



```
)

CREATE INDEX "MODULE_IDX1"
 ON "MODULE" ("FORM_REVISION_ID")

CREATE OR REPLACE TRIGGER "BI_MODULE"
  before insert on "MODULE"
  for each row
begin
    if :NEW.MODULE_ID IS NULL then
        select "MODULE_ID_SEQ".nextval into :NEW.MODULE_ID from dual;
    end if;
end;

ALTER TRIGGER  "BI_MODULE" ENABLE
```

(Microsoft) SQL Server DDL

```
CREATE TABLE MODULE (
  MODULE_ID bigint NOT NULL IDENTITY CONSTRAINT MODULE_ID PRIMARY KEY,
  FORM_REVISION_ID bigint NOT NULL,
  PUBLIC_ID bigint default NULL,
  VERSION decimal(4,2) default NULL,
  ALTERNATE_METADATA_TYPE varchar(255) default NULL,
  ALTERNATE_METADATA_ID varchar(255) default NULL,
  REPEAT_SEQUENCE_NUMBER int NOT NULL,
  MODIFICATION_TIME datetime default NULL,
  MODIFICATION_USER_ID varchar(30) default NULL
)

CREATE INDEX FORM_REVISION_ID
ON MODULE (FORM_REVISION_ID)
```

Sybase DDL

```
CREATE TABLE MODULE (
  MODULE_ID bigint IDENTITY CONSTRAINT MODULE_ID PRIMARY KEY,
  FORM_REVISION_ID bigint NOT NULL,
  PUBLIC_ID bigint NULL,
  VERSION decimal(4,2) NULL,
  ALTERNATE_METADATA_TYPE varchar(255) NULL,
  ALTERNATE_METADATA_ID varchar(255) NULL,
  REPEAT_SEQUENCE_NUMBER int NOT NULL,
  MODIFICATION_TIME datetime NULL,
  MODIFICATION_USER_ID varchar(30) NULL
)

CREATE INDEX FORM_REVISION_ID
ON MODULE (FORM_REVISION_ID)
```


PROCESSING_STATUS Table

Description

Tracks the status of a specific form revision with regard to an information process.

Notes

- **Local Fields:** PROCESSING_STATUS_ID, INFORMATION_PROCESS_ID, FORM_REVISION_ID, PARENT_PROCESSING_STATUS_ID, MODIFICATION_TIME, and MODIFICATION_USER_ID are meaningful only within the local AGNIS (repository or staging) database. For inter-system AGNIS communications in XML format, these fields are normally omitted.

Columns

Column	Data Type	Description
PROCESSING_STATUS_ID	64 bit integer	numeric identifier generated by database
INFORMATION_PROCESS_ID	64 bit integer	partial identifier for this proceessing status; references information process (foreign key)
FORM_REVISION_ID	64 bit integer	partial identifier for this processing status; references form revision (foreign key)
PARENT_PROCESSING_STATUS_ID	64 bit integer	identifier for preceding processing attempt
PROCESSING_RANK	64 bit integer	numeric rank of process; e.g. staging client processes these records in ascending order by rank
ATTEMPT_NUMBER	64 bit integer	numeric attempt number, e.g. for use by staging client retry logic
OUTPUT_FORM_REVISION_ID	64 bit integer	output result from this process, e.g. return value from web service submit operation
VALUE	varchar(20)	status of this process
START_TIME	timestamp	time processing started
END_TIME	timestamp	time processing ended
NOTE	varchar(255)	comment regarding processing status (for unusual circumstances)
MODIFICATION_TIME	timestamp	database time this record was modified (set by trigger?)
MODIFICATION_USER_ID	varchar(30)	database user who modified this record (set by trigger?)

Indexes

Index Type	Columns
Primary Key	PROCESSING_STATUS_ID
Non Unique	FORM_REVISION_ID
Non Unique	INFORMATION_PROCESS_ID
Non Unique	PARENT_PROCESSING_STATUS_ID

Foreign Keys

Column	Foreign Table	Foreign Column
INFORMATION_PROCESS_ID	INFORMATION_PROCESS	INFORMATION_PROCESS_ID
FORM_REVISION_ID	FORM_REVISION	FORM_REVISION_ID

Object Relational Mapping

Type	Column	Class	Attribute/Association
attr	PROCESSING_STATUS_ID	ProcessingStatus	id
attr	INFORMATION_PROCESS_ID	ProcessingStatus	informationProcessId
attr	FORM_REVISION_ID	ProcessingStatus	formRevisionId
attr	PARENT_PROCESSING_STATUS_ID	ProcessingStatus	parentProcessingStatusId
attr	PROCESSING_RANK	ProcessingStatus	processingRank
attr	ATTEMPT_NUMBER	ProcessingStatus	attemptNumber
attr	OUTPUT_FORM_REVISION_ID	ProcessingStatus	outputFormRevisionId
attr	VALUE	ProcessingStatus	value
attr	START_TIME	ProcessingStatus	startTime
attr	END_TIME	ProcessingStatus	endTime
attr	NOTE	ProcessingStatus	note
attr	MODIFICATION_TIME	ProcessingStatus	modificationTime
attr	MODIFICATION_USER_ID	ProcessingStatus	modificationUserId
assn	INFORMATION_PROCESS_ID	ProcessingStatus	informationProcess
assn	FORM_REVISION_ID	ProcessingStatus	formRevision
assn	PARENT_PROCESSING_STATUS_ID	ProcessingStatus	parentProcessingStatus

(attr = mapped-attributes, assn = implements-association)

MySQL DDL

```
CREATE TABLE `PROCESSING_STATUS` (
  `PROCESSING_STATUS_ID` bigint(20) NOT NULL auto_increment,
  `INFORMATION_PROCESS_ID` bigint(20) NOT NULL,
  `FORM_REVISION_ID` bigint(20) default NULL,
  `PARENT_PROCESSING_STATUS_ID` bigint(20) default NULL,
  `PROCESSING_RANK` bigint(20) default NULL,
  `ATTEMPT_NUMBER` bigint(20) default NULL,
  `OUTPUT_FORM_REVISION_ID` bigint(20) default NULL,
  `VALUE` varchar(20) NOT NULL,
  `START_TIME` datetime default NULL,
  `END_TIME` datetime default NULL,
  `NOTE` varchar(255) default NULL,
  `MODIFICATION_TIME` datetime default NULL,
  `MODIFICATION_USER_ID` varchar(30) default NULL,
  PRIMARY KEY  (`PROCESSING_STATUS_ID`),
  KEY `FORM_REVISION_ID` (`FORM_REVISION_ID`),
  KEY `INFORMATION_PROCESS_ID_2` (`INFORMATION_PROCESS_ID`),
  KEY `PARENT_PROCESSING_STATUS_ID` (`PARENT_PROCESSING_STATUS_ID`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
```

Oracle DDL

```
CREATE SEQUENCE "PROCESSING_STATUS_ID_SEQ"

CREATE TABLE "PROCESSING_STATUS"
(
   "PROCESSING_STATUS_ID" NUMBER(18,0),
   "INFORMATION_PROCESS_ID" NUMBER(18,0),
   "FORM_REVISION_ID" NUMBER(18,0),
   "PARENT_PROCESSING_STATUS_ID" NUMBER(18,0),
   "PROCESSING_RANK" NUMBER(18,0),
   "ATTEMPT_NUMBER" NUMBER(18,0),
   "OUTPUT_FORM_REVISION_ID" NUMBER(18,0),
   "VALUE" VARCHAR2(30),
   "START_TIME" TIMESTAMP (6),
   "END_TIME" TIMESTAMP (6),
   "NOTE" VARCHAR2(255),
   "MODIFICATION_TIME" TIMESTAMP (6),
   "MODIFICATION_USER_ID" VARCHAR2(30),
    CONSTRAINT "PROCESSING_STATUS_PK" PRIMARY KEY ("PROCESSING_STATUS_ID") ENABLE
)

CREATE INDEX "PROCESSING_STATUS_IDX1"
 ON "PROCESSING_STATUS" ("FORM_REVISION_ID")

CREATE INDEX "PROCESSING_STATUS_IDX2"
 ON "PROCESSING_STATUS" ("INFORMATION_PROCESS_ID")

CREATE INDEX "PROCESSING_STATUS_IDX3"
 ON "PROCESSING_STATUS" ("PARENT_PROCESSING_STATUS_ID")

CREATE OR REPLACE TRIGGER "BI_PROCESSING_STATUS"
  before insert on "PROCESSING_STATUS"
  for each row
```



```
begin
    if :NEW.PROCESSING_STATUS_ID IS NULL then
        select "PROCESSING_STATUS_ID_SEQ".nextval into :NEW.PROCESSING_STATUS_ID from dual;
    end if;
end;

ALTER TRIGGER "BI_PROCESSING_STATUS" ENABLE
```

(Microsoft) SQL Server DDL

```
CREATE TABLE PROCESSING_STATUS (
  PROCESSING_STATUS_ID bigint NOT NULL IDENTITY CONSTRAINT PROCESSING_STATUS_ID PRIMARY KEY,
  INFORMATION_PROCESS_ID bigint NOT NULL,
  FORM_REVISION_ID bigint default NULL,
  PARENT_PROCESSING_STATUS_ID bigint default NULL,
  PROCESSING_RANK bigint default NULL,
  ATTEMPT_NUMBER bigint default NULL,
  OUTPUT_FORM_REVISION_ID bigint default NULL,
  VALUE varchar(20) NOT NULL,
  START_TIME datetime default NULL,
  END_TIME datetime default NULL,
  NOTE varchar(255) default NULL,
  MODIFICATION_TIME datetime default NULL,
  MODIFICATION_USER_ID varchar(30) default NULL
)

CREATE INDEX FORM_REVISION_ID
ON PROCESSING_STATUS (FORM_REVISION_ID)

CREATE INDEX INFORMATION_PROCESS_ID
ON PROCESSING_STATUS (INFORMATION_PROCESS_ID)

CREATE INDEX PARENT_PROCESSING_STATUS_ID
ON PROCESSING_STATUS (PARENT_PROCESSING_STATUS_ID)
```

Sybase DDL

```
CREATE TABLE PROCESSING_STATUS (
  PROCESSING_STATUS_ID bigint IDENTITY CONSTRAINT PROCESSING_STATUS_ID PRIMARY KEY,
  INFORMATION_PROCESS_ID bigint NOT NULL,
  FORM_REVISION_ID bigint NULL,
  PARENT_PROCESSING_STATUS_ID bigint NULL,
  PROCESSING_RANK bigint NULL,
  ATTEMPT_NUMBER bigint NULL,
  OUTPUT_FORM_REVISION_ID bigint NULL,
  VALUE varchar(20) NOT NULL,
  START_TIME datetime NULL,
  END_TIME datetime NULL,
  NOTE varchar(255) NULL,
  MODIFICATION_TIME datetime NULL,
  MODIFICATION_USER_ID varchar(30) NULL
)

CREATE INDEX FORM_REVISION_ID
ON PROCESSING_STATUS (FORM_REVISION_ID)
```



```
CREATE INDEX INFORMATION_PROCESS_ID_2
ON PROCESSING_STATUS (INFORMATION_PROCESS_ID)

CREATE INDEX PARENT_PROCESSING_STATUS_ID
ON PROCESSING_STATUS (PARENT_PROCESSING_STATUS_ID)
```


QUESTION Table

Description

Represents an instance of question belonging to a form which has Case Report Form (CRF) metadata defined in the caDSR metadata repository.

Notes

- **Local Fields:** QUESTION_ID, MODULE_ID, MODIFICATION_TIME, and MODIFICATION_USER_ID are meaningful only within the local AGNIS (repository or staging) database. For inter-system AGNIS communications in XML format, these fields are normally omitted.

- **Alternate Identifiers:** A question instance is identified by either (MODULE_ID, DATA_ELEMENT_PUBLIC_ID, DATA_ELEMENT_VERSION, REPEAT_SEQUENCE_NUMBER) or (MODULE_ID, ALTERNATE_METADATA_TYPE, ALTERNATE_METADATA_ID, REPEAT_SEQUENCE_NUMBER), depending on whether caDSR or alternate (non-caDSR) metadata has been specified.

- **Delete Flag:** The *Question.delete* attribute is not strictly part of the Question data, but is intended to be used for communication purposes by the AGNIS staging client or similar.

Columns

Column	Data Type	Description
QUESTION_ID	64 bit integer	numeric identifier generated by database
MODULE_ID	64 bit integer	partial identifier for this question instance; identifies module associated with this question (foreign key)
DATA_ELEMENT_PUBLIC_ID	64 bit integer	partial identifier for this question instance; also partial identifier for the caDSR CDE associated with this question
DATA_ELEMENT_VERSION	numeric(4,2)	partial identifier for this question instance; also partial identifier for the caDSR CDE associated with this question
ALTERNATE_METADATA_TYPE	varchar(255)	partial identifier for this question instance; also partial identifier for alternate (non-caDSR) metadata associated with this question
ALTERNATE_METADATA_ID	varchar(255)	partial identifier for this question instance; also partial identifier for alternate (non-caDSR) metadata associated with this question
VALUE	varchar(255)	data value for this question, based on caDSR CDE
ALTERNATE_VALUE	varchar(255)	data value for this question, based on alternate (non-caDSR) metadata
ERROR_EXCEPTION	varchar(20)	indicates whether any validation error was overridden for this question
DELETE_FLAG	8 bit integer	pseudo-boolean delete flag for this question
VALUE_MEANING_PUBLIC_ID	64 bit integer	partial identifier for caDSR value meaning associated with this question's value
VALUE_MEANING_VERSION	numeric(4,2)	also partial identifier for caDSR value meaning associated with this question's value
MODIFICATION_TIME	timestamp	database time this record was modified (set by trigger?)
MODIFICATION_USER_ID	varchar(30)	database user who modified this record (set by trigger?)

Indexes

Index Type	Columns
Primary Key	QUESTION_ID
Non Unique	MODULE_ID

Foreign Keys

Column	Foreign Table	Foreign Column
MODULE_ID	MODULE	MODULE_ID

Object Relational Mapping

Type	Column	Class	Attribute/Association
attr	QUESTION_ID	Question	id
attr	MODULE_ID	Question	moduleId
attr	DATA_ELEMENT_PUBLIC_ID	Question	dataElementPublicId
attr	DATA_ELEMENT_VERSION	Question	dataElementVersion
attr	ALTERNATE_METADATA_TYPE	Question	alternateMetadataType
attr	ALTERNATE_METADATA_ID	Question	alternateMetadataId
attr	VALUE	Question	value
attr	ALTERNATE_VALUE	Question	alternateValue
attr	ERROR_EXCEPTION	Question	errorException
attr	DELETE_FLAG	Question	delete
attr	VALUE_MEANING_PUBLIC_ID	Question	valueMeaningPublicId
attr	VALUE_MEANING_VERSION	Question	valueMeaningVersion
attr	MODIFICATION_TIME	Question	modificationTime
attr	MODIFICATION_USER_ID	Question	modificationUserId
assn	MODULE_ID	Question	module

(attr = mapped-attributes, assn = implements-association)

MySQL DDL

```
CREATE TABLE `QUESTION` (
  `QUESTION_ID` bigint(20) NOT NULL auto_increment,
  `MODULE_ID` bigint(20) NOT NULL,
  `DATA_ELEMENT_PUBLIC_ID` bigint(20) default NULL,
  `DATA_ELEMENT_VERSION` decimal(4,2) default NULL,
  `ALTERNATE_METADATA_TYPE` varchar(255) default NULL,
  `ALTERNATE_METADATA_ID` varchar(255) default NULL,
  `VALUE` varchar(255) default NULL,
  `ALTERNATE_VALUE` varchar(255) default NULL,
  `ERROR_EXCEPTION` varchar(20) default NULL,
  `DELETE_FLAG` tinyint(1) default NULL,
  `VALUE_MEANING_PUBLIC_ID` bigint(20) default NULL,
  `VALUE_MEANING_VERSION` decimal(4,2) default NULL,
  `MODIFICATION_TIME` datetime default NULL,
  `MODIFICATION_USER_ID` varchar(30) default NULL,
  PRIMARY KEY  (`QUESTION_ID`),
  KEY `MODULE_ID` (`MODULE_ID`)
```



```
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
```

Oracle DDL

```
CREATE SEQUENCE "QUESTION_ID_SEQ"

CREATE TABLE "QUESTION"
(
   "QUESTION_ID" NUMBER(18,0) NOT NULL ENABLE,
   "MODULE_ID" NUMBER(18,0),
   "DATA_ELEMENT_PUBLIC_ID" NUMBER(19,0),
   "DATA_ELEMENT_VERSION" NUMBER(4,2),
   "VALUE" VARCHAR2(4000),
   "ALTERNATE_VALUE" VARCHAR2(4000),
   "ERROR_EXCEPTION" VARCHAR2(4000),
   "DELETE_FLAG" NUMBER(1,0),
   "VALUE_MEANING_PUBLIC_ID" NUMBER(19,0),
   "VALUE_MEANING_VERSION" NUMBER(4,2),
   "MODIFICATION_TIME" TIMESTAMP (6),
   "MODIFICATION_USER_ID" VARCHAR2(4000),
   "ALTERNATE_METADATA_TYPE" VARCHAR2(255),
   "ALTERNATE_METADATA_ID" VARCHAR2(255),
    CONSTRAINT "QUESTION_PK" PRIMARY KEY ("QUESTION_ID") ENABLE
)

CREATE INDEX "QUESTION_IDX1"
 ON "QUESTION" ("MODULE_ID")

CREATE OR REPLACE TRIGGER "BI_QUESTION"
  before insert on "QUESTION"
  for each row
begin
    if :NEW.QUESTION_ID IS NULL then
        select "QUESTION_ID_SEQ".nextval into :NEW.QUESTION_ID from dual;
    end if;
end;

ALTER TRIGGER "BI_QUESTION" ENABLE
```

(Microsoft) SQL Server DDL

```
CREATE TABLE QUESTION (
  QUESTION_ID bigint NOT NULL IDENTITY CONSTRAINT QUESTION_ID PRIMARY KEY,
  MODULE_ID bigint NOT NULL,
  DATA_ELEMENT_PUBLIC_ID bigint default NULL,
  DATA_ELEMENT_VERSION decimal(4,2) default NULL,
  ALTERNATE_METADATA_TYPE varchar(255) default NULL,
  ALTERNATE_METADATA_ID varchar(255) default NULL,
  VALUE varchar(255) default NULL,
  ALTERNATE_VALUE varchar(255) default NULL,
  ERROR_EXCEPTION varchar(20) default NULL,
  DELETE_FLAG tinyint default NULL,
  VALUE_MEANING_PUBLIC_ID bigint default NULL,
  VALUE_MEANING_VERSION decimal(4,2) default NULL,
  MODIFICATION_TIME datetime default NULL,
  MODIFICATION_USER_ID varchar(30) default NULL
```



```
)

CREATE INDEX MODULE_ID
ON QUESTION (MODULE_ID)
```

Sybase DDL

```
CREATE TABLE QUESTION (
  QUESTION_ID bigint IDENTITY CONSTRAINT QUESTION_ID PRIMARY KEY,
  MODULE_ID bigint NOT NULL,
  DATA_ELEMENT_PUBLIC_ID bigint NULL,
  DATA_ELEMENT_VERSION decimal(4,2) NULL,
  ALTERNATE_METADATA_TYPE varchar(255) NULL,
  ALTERNATE_METADATA_ID varchar(255) NULL,
  VALUE varchar(255) NULL,
  ALTERNATE_VALUE varchar(255) NULL,
  ERROR_EXCEPTION varchar(20) NULL,
  DELETE_FLAG tinyint NULL,
  VALUE_MEANING_PUBLIC_ID bigint NULL,
  VALUE_MEANING_VERSION decimal(4,2) NULL,
  MODIFICATION_TIME datetime NULL,
  MODIFICATION_USER_ID varchar(30) NULL
)

CREATE INDEX MODULE_ID
ON QUESTION (MODULE_ID)
```


RETRIEVAL Table

Description

Represents a set of form revisions retrieved by an AGNIS subscriber.

Notes

- **Local Fields:** RETRIEVAL_ID, SUBSCRIBER_INSTITUTION_ID, MODIFICATION_TIME, and MODIFICATION_USER_ID are meaningful only within the local AGNIS (repository or staging) database. For inter-system AGNIS communications in XML format, these fields are normally omitted.

Columns

Column	Data Type	Description
RETRIEVAL_ID	64 bit integer	numeric identifier generated by database
SUBSCRIBER_INSTITUTION_ID	64 bit integer	partial retrieval identifier; identifies subscriber associated with this retrieval (foreign key)
RETRIEVAL_SEQ_NUM	64 bit integer	partial retrieval identifier
TRANSMITTED_TIME	timestamp	numeric identifier generated by database
RETRIEVAL_ACKNOWLEDGED_STATUS	varchar(20)	indicates whether this retrieval has been acknowledged by the subscriber
RETRIEVAL_ACKNOWLEDGED_TIME	timestamp	time at which this retrieval was acknowledged by the subscriber
MODIFICATION_TIME	timestamp	database time this record was modified (set by trigger?)
MODIFICATION_USER_ID	varchar(30)	database user who modified this record (set by trigger?)

Indexes

Index Type	Columns
Primary Key	RETRIEVAL_ID
Unique	SUBSCRIBER_INSTITUTION_ID, RETRIEVAL_SEQ_NUM
Non Unique	SUBSCRIBER_INSTITUTION_ID

Foreign Keys

Column	Foreign Table	Foreign Column
SUBSCRIBER_INSTITUTION_ID	INSTITUTION	INSTITUTION_ID

Object Relational Mapping

Type	Column	Class	Attribute/Association
attr	RETRIEVAL_ID	Retrieval	id
attr	SUBSCRIBER_INSTITUTION_ID	Retrieval	subscriberInstitutionId
attr	RETRIEVAL_SEQ_NUM	Retrieval	sequenceNumber
attr	TRANSMITTED_TIME	Retrieval	transmittedTime
attr	RETRIEVAL_ACKNOWLEDGED_STATUS	Retrieval	acknowledgedStatus
attr	RETRIEVAL_ACKNOWLEDGED_TIME	Retrieval	acknowledgedTime
attr	MODIFICATION_TIME	Retrieval	modificationTime
attr	MODIFICATION_USER_ID	Retrieval	modificationUserId
assn	SUBSCRIBER_INSTITUTION_ID	Retrieval	subscriber

(attr = mapped-attributes, assn = implements-association)

MySQL DDL

```
CREATE TABLE `RETRIEVAL` (
  `RETRIEVAL_ID` bigint(20) NOT NULL auto_increment,
  `SUBSCRIBER_INSTITUTION_ID` bigint(20) NOT NULL,
  `RETRIEVAL_SEQ_NUM` bigint(20) NOT NULL,
  `TRANSMITTED_TIME` datetime default NULL,
  `RETRIEVAL_ACKNOWLEDGED_STATUS` varchar(30) default NULL,
  `RETRIEVAL_ACKNOWLEDGED_TIME` datetime default NULL,
  `MODIFICATION_TIME` datetime default NULL,
  `MODIFICATION_USER_ID` varchar(30) default NULL,
  PRIMARY KEY  (`RETRIEVAL_ID`),
  UNIQUE KEY `SUBSCRIBER_INSTITUTION_ID` (`SUBSCRIBER_INSTITUTION_ID`,`RETRIEVAL_SEQ_NUM`),
  KEY `SUBSCRIBER_INSTITUTION_ID_2` (`SUBSCRIBER_INSTITUTION_ID`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
```

Oracle DDL

```
CREATE SEQUENCE "RETRIEVAL_ID_SEQ"

CREATE TABLE  "RETRIEVAL"
(
   "RETRIEVAL_ID" NUMBER(18,0) NOT NULL ENABLE,
   "SUBSCRIBER_INSTITUTION_ID" NUMBER(18,0),
   "RETRIEVAL_SEQ_NUM" NUMBER(18,0),
   "TRANSMITTED_TIME" TIMESTAMP (6),
   "RETRIEVAL_ACKNOWLEDGED_STATUS" VARCHAR2(30),
```



```
   "RETRIEVAL_ACKNOWLEDGED_TIME" TIMESTAMP (6),
   "MODIFICATION_TIME" TIMESTAMP (6),
   "MODIFICATION_USER_ID" VARCHAR2(30),
    CONSTRAINT "RETRIEVAL_PK" PRIMARY KEY ("RETRIEVAL_ID") ENABLE
)

CREATE UNIQUE INDEX "RETRIEVAL_IDX1"
 ON "RETRIEVAL" ("SUBSCRIBER_INSTITUTION_ID","RETRIEVAL_SEQ_NUM")

CREATE INDEX "RETRIEVAL_IDX2"
 ON "RETRIEVAL" ("SUBSCRIBER_INSTITUTION_ID")

CREATE OR REPLACE TRIGGER "BI_RETRIEVAL"
  before insert on "RETRIEVAL"
  for each row
begin
    if :NEW.RETRIEVAL_ID IS NULL then
        select "RETRIEVAL_ID_SEQ".nextval into :NEW.RETRIEVAL_ID from dual;
    end if;
end;

ALTER TRIGGER  "BI_RETRIEVAL" ENABLE
```

(Microsoft) SQL Server DDL

```
CREATE TABLE RETRIEVAL (
  RETRIEVAL_ID bigint NOT NULL IDENTITY CONSTRAINT RETRIEVAL_ID PRIMARY KEY,
  SUBSCRIBER_INSTITUTION_ID bigint NOT NULL,
  RETRIEVAL_SEQ_NUM bigint NOT NULL,
  TRANSMITTED_TIME datetime default NULL,
  RETRIEVAL_ACKNOWLEDGED_STATUS varchar(30) default NULL,
  RETRIEVAL_ACKNOWLEDGED_TIME datetime default NULL,
  MODIFICATION_TIME datetime default NULL,
  MODIFICATION_USER_ID varchar(30) default NULL
)

CREATE UNIQUE INDEX SUBSCRIBER_INSTITUTION_ID
ON RETRIEVAL (SUBSCRIBER_INSTITUTION_ID,RETRIEVAL_SEQ_NUM)

CREATE INDEX SUBSCRIBER_INSTITUTION_ID_2
ON RETRIEVAL (SUBSCRIBER_INSTITUTION_ID)
```

Sybase DDL

```
CREATE TABLE RETRIEVAL (
  RETRIEVAL_ID bigint IDENTITY CONSTRAINT RETRIEVAL_ID PRIMARY KEY,
  SUBSCRIBER_INSTITUTION_ID bigint NOT NULL,
  RETRIEVAL_SEQ_NUM bigint NOT NULL,
  TRANSMITTED_TIME datetime NULL,
  RETRIEVAL_ACKNOWLEDGED_STATUS varchar(30) NULL,
  RETRIEVAL_ACKNOWLEDGED_TIME datetime NULL,
  MODIFICATION_TIME datetime NULL,
  MODIFICATION_USER_ID varchar(30) NULL
)

CREATE UNIQUE INDEX SUBSCRIBER_INSTITUTION_ID
```



```
ON RETRIEVAL (SUBSCRIBER_INSTITUTION_ID,RETRIEVAL_SEQ_NUM)

CREATE INDEX SUBSCRIBER_INSTITUTION_ID_2
ON RETRIEVAL (SUBSCRIBER_INSTITUTION_ID)
```


RETRIEVAL_FORM_REVISION Table

Description

Correlation table for the many-to-many relationship between the RETRIEVAL and FORM_REVISION tables

Notes

- **Correlation Table:** This is a correlation table, and its data is not directly transmitted as part of inter-system AGNIS communications in XML format.

Columns

Column	Data Type	Description
RETRIEVAL_ID	64 bit integer	partial retrieval form revision identifier; identifies retrieval (foreign key)
FORM_REVISION_ID	64 bit integer	partial retrieval form revision identifier; identifies form revision (foreign key)
MODIFICATION_TIME	timestamp	database time this record was modified (set by trigger?)
MODIFICATION_USER_ID	varchar(30)	database user who modified this record (set by trigger?)

Indexes

Index Type	Columns
Unique	RETRIEVAL_ID, FORM_REVISION_ID
Non Unique	RETRIEVAL_ID
Non Unique	FORM_REVISION_ID

Foreign Keys

Column	Foreign Table	Foreign Column
RETRIEVAL_ID	RETRIEVAL	RETRIEVAL_ID
FORM_REVISION_ID	FORM_REVISION	FORM_REVISION_ID

Object Relational Mapping

None.

MySQL DDL

```
CREATE TABLE `RETRIEVAL_FORM_REVISION` (
  `RETRIEVAL_ID` bigint(20) NOT NULL,
  `FORM_REVISION_ID` bigint(20) NOT NULL,
  `MODIFICATION_TIME` datetime default NULL,
  `MODIFICATION_USER_ID` varchar(30) default NULL,
  UNIQUE KEY `RETRIEVAL_ID` (`RETRIEVAL_ID`,`FORM_REVISION_ID`),
  KEY `FORM_REVISION_ID` (`FORM_REVISION_ID`),
  KEY `RETRIEVAL_ID_2` (`RETRIEVAL_ID`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
```

Oracle DDL

```
CREATE TABLE "RETRIEVAL_FORM_REVISION"
(
   "RETRIEVAL_ID" NUMBER(18,0) NOT NULL ENABLE,
   "FORM_REVISION_ID" NUMBER(18,0) NOT NULL ENABLE,
   "MODIFICATION_TIME" TIMESTAMP (6),
   "MODIFICATION_USER_ID" VARCHAR2(30)
)

CREATE UNIQUE INDEX "RETRIEVAL_FORM_REVISION_IDX1"
 ON "RETRIEVAL_FORM_REVISION" ("RETRIEVAL_ID","FORM_REVISION_ID")

CREATE INDEX "RETRIEVAL_FORM_REVISION_IDX2"
 ON "RETRIEVAL_FORM_REVISION" ("FORM_REVISION_ID")

CREATE INDEX "RETRIEVAL_FORM_REVISION_IDX3"
 ON "RETRIEVAL_FORM_REVISION" ("RETRIEVAL_ID")
```

(Microsoft) SQL Server DDL

```
CREATE TABLE RETRIEVAL_FORM_REVISION (
  RETRIEVAL_ID bigint NOT NULL,
  FORM_REVISION_ID bigint NOT NULL,
  MODIFICATION_TIME datetime default NULL,
  MODIFICATION_USER_ID varchar(30) default NULL
)

CREATE UNIQUE INDEX RETRIEVAL_ID
ON RETRIEVAL_FORM_REVISION (RETRIEVAL_ID,FORM_REVISION_ID)

CREATE INDEX FORM_REVISION_ID
ON RETRIEVAL_FORM_REVISION (FORM_REVISION_ID)

CREATE INDEX RETRIEVAL_ID_2
ON RETRIEVAL_FORM_REVISION (RETRIEVAL_ID)
```


Sybase DDL

```
CREATE TABLE RETRIEVAL_FORM_REVISION (
  RETRIEVAL_ID bigint NOT NULL,
  FORM_REVISION_ID bigint NOT NULL,
  MODIFICATION_TIME datetime NULL,
  MODIFICATION_USER_ID varchar(30) NULL
)

CREATE UNIQUE INDEX RETRIEVAL_ID
ON RETRIEVAL_FORM_REVISION (RETRIEVAL_ID,FORM_REVISION_ID)

CREATE INDEX FORM_REVISION_ID
ON RETRIEVAL_FORM_REVISION (FORM_REVISION_ID)

CREATE INDEX RETRIEVAL_ID_2
ON RETRIEVAL_FORM_REVISION (RETRIEVAL_ID)
```


TEST_QUESTION Table

Description

Represents a secondary question associated with a validation error.

Notes

- **Local Fields:** TEST_QUESTION_ID, FORM_ELEMENT_ERROR_ID, MODIFICATION_TIME, and MODIFICATION_USER_ID are meaningful only within the local AGNIS (repository or staging) database. For inter-system AGNIS communications in XML format, these fields are normally omitted.

- **Alternate Identifiers:** A testQuestion instance can be identified by either (FORM_ELEMENT_ERROR_ID, DATA_ELEMENT_PUBLIC_ID, DATA_ELEMENT_VERSION, REPEAT_SEQUENCE_NUMBER) or (FORM_ELEMENT_ERROR_ID, ALTERNATE_METADATA_TYPE, ALTERNATE_METADATA_ID, REPEAT_SEQUENCE_NUMBER), depending on whether caDSR or alternate (non-caDSR) metadata has been specified

Columns

Column	Data Type	Description
TEST_QUESTION_ID	64 bit integer	numeric identifier generated by database
FORM_ELEMENT_ERROR_ID	64 bit integer	partial identifier for test question; identifies error associated with this test question (foreign key)
MODULE_PUBLIC_ID	64 bit integer	partial identifier for test question; also partial caDSR identifier for CRF Module metadata
MODULE_VERSION	numeric(4,2)	partial identifier for test question; also partial caDSR identifier for CRF Module metadata
MODULE_ALTERNATE_METADATA_TYPE	varchar(255)	partial identifier for test question; also partial identifier for alternate (non-caDSR) module metadata
MODULE_ALTERNATE_METADATA_ID	varchar(255)	partial identifier for test question; also partial identifier for alternate (non-caDSR) module metadata
REPEAT_SEQUENCE_NUMBER	32 bit integer	partial identifier for test question
DATA_ELEMENT_PUBLIC_ID	64 bit integer	partial identifier for test question; also partial caDSR identifier for CRF Question metadata
DATA_ELEMENT_VERSION	numeric(4,2)	partial identifier for test question; also partial caDSR identifier for CRF Question metadata
ALTERNATE_METADATA_TYPE	varchar(255)	partial identifier for test question; also partial identifier for alternate (non-caDSR) question metadata
ALTERNATE_METADATA_ID	varchar(255)	partial identifier for test question; also partial identifier for alternate (non-caDSR) question metadata
MODIFICATION_TIME	timestamp	database time this record was modified (set by trigger?)
MODIFICATION_USER_ID	varchar(30)	database user who modified this record (set by trigger?)

Indexes

Index Type	Columns
Primary Key	TEST_QUESTION_ID
Non Unique	FORM_ELEMENT_ERROR_ID

Foreign Keys

Column	Foreign Table	Foreign Column
FORM_ELEMENT_ERROR_ID	FORM_ELEMENT_ERROR	FORM_ELEMENT_ERROR_ID

Object Relational Mapping

Type	Column	Class	Attribute
attr	TEST_QUESTION_ID	TestQuestion	id
attr	FORM_ELEMENT_ERROR_ID	TestQuestion	formElementErrorId
attr	MODULE_PUBLIC_ID	TestQuestion	modulePublicId
attr	MODULE_VERSION	TestQuestion	moduleVersion
attr	MODULE_ALTERNATE_METADATA_TYPE	TestQuestion	moduleAlternateMetadataType
attr	MODULE_ALTERNATE_METADATA_ID	TestQuestion	moduleAlternateMetadataId
attr	REPEAT_SEQUENCE_NUMBER	TestQuestion	repeatSequenceNumber
attr	DATA_ELEMENT_PUBLIC_ID	TestQuestion	dataElementPublicId
attr	DATA_ELEMENT_VERSION	TestQuestion	dataElementVersion
attr	ALTERNATE_METADATA_TYPE	TestQuestion	alternateMetadataType
attr	ALTERNATE_METADATA_ID	TestQuestion	alternateMetadataId
attr	MODIFICATION_TIME	TestQuestion	modificationTime
attr	MODIFICATION_USER	TestQuestion	modificationUser
assn	FORM_ELEMENT_ERROR_ID	TestQuestion	formElementError

(attr = mapped-attributes, assn = implements-association)

MySQL DDL

```
CREATE TABLE `TEST_QUESTION` (
  `TEST_QUESTION_ID` bigint(20) NOT NULL auto_increment,
  `FORM_ELEMENT_ERROR_ID` bigint(20) NOT NULL,
  `MODULE_PUBLIC_ID` bigint(20) default NULL,
  `MODULE_VERSION` decimal(4,2) default NULL,
  `MODULE_ALTERNATE_METADATA_TYPE` varchar(255) default NULL,
  `MODULE_ALTERNATE_METADATA_ID` varchar(255) default NULL,
  `REPEAT_SEQUENCE_NUMBER` int(11) NOT NULL,
  `DATA_ELEMENT_PUBLIC_ID` bigint(20) default NULL,
  `DATA_ELEMENT_VERSION` decimal(4,2) default NULL,
```



```
  `ALTERNATE_METADATA_TYPE` varchar(255) default NULL,
  `ALTERNATE_METADATA_ID` varchar(255) default NULL,
  `MODIFICATION_TIME` datetime default NULL,
  `MODIFICATION_USER_ID` varchar(30) default NULL,
  PRIMARY KEY  (`TEST_QUESTION_ID`),
  KEY `FORM_ELEMENT_ERROR_ID` (`FORM_ELEMENT_ERROR_ID`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
```

Oracle DDL

```
CREATE SEQUENCE "TEST_QUESTION_ID_SEQ"

CREATE TABLE "TEST_QUESTION"
(
   "TEST_QUESTION_ID" NUMBER(18,0) NOT NULL ENABLE,
   "FORM_ELEMENT_ERROR_ID" NUMBER(18,0),
   "MODULE_PUBLIC_ID" NUMBER(19,0),
   "MODULE_VERSION" NUMBER(4,2),
   "MODULE_ALTERNATE_METADATA_TYPE" VARCHAR2(255),
   "MODULE_ALTERNATE_METADATA_ID" VARCHAR2(255),
   "REPEAT_SEQUENCE_NUMBER" NUMBER(8,0),
   "DATA_ELEMENT_PUBLIC_ID" NUMBER(19,0),
   "DATA_ELEMENT_VERSION" NUMBER(4,2),
   "ALTERNATE_METADATA_TYPE" VARCHAR2(255),
   "ALTERNATE_METADATA_ID" VARCHAR2(255),
   "MODIFICATION_TIME" TIMESTAMP (6),
   "MODIFICATION_USER_ID" VARCHAR2(4000),
    CONSTRAINT "TEST_QUESTION_PK" PRIMARY KEY ("TEST_QUESTION_ID") ENABLE
)

CREATE INDEX "TEST_QUESTION_IDX1"
 ON "TEST_QUESTION" ("FORM_ELEMENT_ERROR_ID")

CREATE OR REPLACE TRIGGER "BI_TEST_QUESTION"
  before insert on "TEST_QUESTION"
  for each row
begin
    if :NEW.TEST_QUESTION_ID IS NULL then
        select "TEST_QUESTION_ID_SEQ".nextval into :NEW.TEST_QUESTION_ID from dual;
    end if;
end;

ALTER TRIGGER "BI_TEST_QUESTION" ENABLE
```

(Microsoft) SQL Server DDL

```
CREATE TABLE TEST_QUESTION (
  TEST_QUESTION_ID bigint NOT NULL IDENTITY CONSTRAINT TEST_QUESTION_ID PRIMARY KEY,
  FORM_ELEMENT_ERROR_ID bigint NOT NULL,
  MODULE_PUBLIC_ID bigint default NULL,
  MODULE_VERSION decimal(4,2) default NULL,
  MODULE_ALTERNATE_METADATA_TYPE varchar(255) default NULL,
  MODULE_ALTERNATE_METADATA_ID varchar(255) default NULL,
  REPEAT_SEQUENCE_NUMBER int NOT NULL,
  DATA_ELEMENT_PUBLIC_ID bigint default NULL,
  DATA_ELEMENT_VERSION decimal(4,2) default NULL,
```



```
  ALTERNATE_METADATA_TYPE varchar(255) default NULL,
  ALTERNATE_METADATA_ID varchar(255) default NULL,
  MODIFICATION_TIME datetime default NULL,
  MODIFICATION_USER_ID varchar(30) default NULL
)

CREATE INDEX FORM_ELEMENT_ERROR_ID
ON TEST_QUESTION (FORM_ELEMENT_ERROR_ID)
```

Sybase DDL

```
CREATE TABLE TEST_QUESTION (
  TEST_QUESTION_ID bigint IDENTITY CONSTRAINT TEST_QUESTION_ID PRIMARY KEY,
  FORM_ELEMENT_ERROR_ID bigint NOT NULL,
  MODULE_PUBLIC_ID bigint NULL,
  MODULE_VERSION decimal(4,2) NULL,
  MODULE_ALTERNATE_METADATA_TYPE varchar(255) NULL,
  MODULE_ALTERNATE_METADATA_ID varchar(255) NULL,
  REPEAT_SEQUENCE_NUMBER int NOT NULL,
  DATA_ELEMENT_PUBLIC_ID bigint NULL,
  DATA_ELEMENT_VERSION decimal(4,2) NULL,
  ALTERNATE_METADATA_TYPE varchar(255) NULL,
  ALTERNATE_METADATA_ID varchar(255) NULL,
  MODIFICATION_TIME datetime NULL,
  MODIFICATION_USER_ID varchar(30) NULL
)

CREATE INDEX FORM_ELEMENT_ERROR_ID
ON TEST_QUESTION (FORM_ELEMENT_ERROR_ID)
```


Appendix: Database Creation Examples

The AGNIS formhandler source code package, downloadable from the AGNIS.net web site, contains database creation scripts for MySQL, Oracle, (Microsoft) SQL Server, and Sybase. Below are brief examples illustrating how to execute those database creation scripts using command line tools from each of the various databases.

MySQL

```
> mysql -u root -p -h hostname -P 3306
Enter password:
mysql> create database AGNIS_FORMS;
mysql> use AGNIS_FORMS;
mysql> source AGNIS_FORMS-createdb-mysql.sql
mysql> exit
```

Oracle

With Oracle, each schema is tied to a user, so the example below illustrates creating a user first, followed by connecting as that user and running the AGNIS database setup script.

```
> sqlplus /nolog
SQL> connect SYSTEM@hostname:1521
Enter password:
Connected.
SQL> create user AGNIS_FORMS identified by "password"
  2  default tablespace USERS temporary tablespace TEMP;
User created.
SQL> grant CONNECT, RESOURCE to AGNIS_FORMS;
Grant succeeded.
SQL> disconnect
Disconnected ...
SQL> connect AGNIS_FORMS@hostname:1521
Enter password:
Connected.
SQL> start 'AGNIS_FORMS-createdb-oracle.sql'
...
SQL> exit
```

(Microsoft) SQL Server

Newer versions of Microsoft SQL Server provide the `sqlcmd` utility depicted here. Older versions provide an `isql` utility similar to that used by Sybase (see Sybase section for `isql` example).

```
> sqlcmd -U sa -S servername
Password:
1> create database AGNIS_FORMS
2> go
1> exit
> sqlcmd -U sa -S servername -d AGNIS_FORMS -i AGNIS_FORMS-createdb-mssql.sql
Password:
```


Sybase

```
> isql -U sa -S servername
Password:
1> create database AGNIS_FORMS on DEFAULT = 10
2> go
...
1> exit
> isql -U sa -S servername -D AGNIS_FORMS -i AGNIS_FORMS-createdb-sybase.sql
Password:
```

